sensors

MDPI

Article

Robust Identification System for Spanish Sign Language Based on Three-Dimensional Frame Information

Jesús Galván-Ruiz [1], Carlos M. Travieso-González [1,2,*], Alejandro Pinan-Roescher [1] and Jesús B. Alonso-Hernández [1,2]

[1] IDeTIC, Universidad de Las Palmas de G.C. (ULPGC), 35017 Las Palmas de Gran Canaria, Spain
[2] Signals and Communications Department, Universidad de Las Palmas de G.C. (ULPGC), 35017 Las Palmas de Gran Canaria, Spain
* Correspondence: carlos.travieso@ulpgc.es

Abstract: Nowadays, according to the World Health Organization (WHO), of the world's population suffers from a hearing disorder that makes oral communication with other people challenging. At the same time, in an era of technological evolution and digitization, designing tools that could help these people to communicate daily is the base of much scientific research such as that discussed herein. This article describes one of the techniques designed to transcribe Spanish Sign Language (SSL). A Leap Motion volumetric sensor has been used in this research due to its capacity to recognize hand movements in 3 dimensions. In order to carry out this research project, an impaired hearing subject has collaborated in the recording of 176 dynamic words. Finally, for the development of the research, Dynamic Time Warping (DTW) has been used to compare the samples and predict the input with an accuracy of 95.17%.

Keywords: dynamic time warping; Spanish sign language; gesture recognition; pattern recognition

check for updates

Citation: Galván-Ruiz, J.; Travieso-González, C.M.; Pinan-Roescher, A.; Alonso-Hernández, J.B. Robust Identification System for Spanish Sign Language Based on Three-Dimensional Frame Information. *Sensors* **2023**, *23*, 481. https://doi.org/10.3390/s23010481

Academic Editor: Eui Chul Lee

Received: 6 December 2022
Revised: 23 December 2022
Accepted: 30 December 2022
Published: 2 January 2023

Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Based on data retrieved from the United Nations (UN) [1] and considering that a percentage of people with impaired hearing, depending on the area where they live, use at least one of the 300 sign languages that currently exist, there is a need to break down barriers to better integrate hearing-impaired people into a speaking society. Even though most people with a hearing impairment are capable of lip reading, people without a hearing impairment find it challenging to communicate with them. This fact, together with the challenges brought on by the current COVID-19 pandemic, where the use of face masks covering nose and mouth became mandatory in many places, has made it increasingly necessary to develop a system that helps people with this disability. Likewise, the progress in new technologies such as artificial intelligence (AI), sensors, hardware, networks, etc., are improving the lives of many people in different knowledge areas of everyday life. The field of gesture recognition began 70 years ago.

1.1. Technologies

In the 1960s, tablets and pens that were capable of capturing writing by using touch-sensitive interfaces or pointing devices came into use. Data gloves, such as the Data Glover [2], also began to be used with the disadvantage that this type of system uses an invasive element. However, these active systems were connected by cables that were a handicap for their versatility. Later, thanks to wireless technologies, these elements were removed, yet the sensors made the glove thicker and, hence, complicated to use.

At the start of the 1980s, other types of systems were developed, such as passive gloves, which had different colors depending on the part of the hand they covered. With these gloves and the vision systems that began to appear at the turn of the decade, together with the best computer systems, hand movements in two dimensions began to be detected.

In the 1990s, there was a very important leap in the creation of new sensors and the improvement of computing equipment. This led to the creation of new active gloves with more precise sensors that were able to reproduce the movement of a hand in a real way. There were also significant advances in the treatment of images and new digital cameras that greatly improved the detection of hand movements in real time [3–5].

There are also studies which work with electromyography (EMG) and electrodes [6–9]. These studies are based on the neuromuscular system using electrodes that can detect electrical signals produced by muscles and nerves.

Ultrasound motion-detection systems have been used in several studies, thanks to the Doppler Effect [10–12]. Our tissues can endure different acoustic impedances by reflecting different amounts of energy. Nonetheless, this system is not very precise compared to the others mentioned due to losses by occlusion where some parts of the body are left behind others.

The boom of wireless communications began in the year 2000. The system imposed to date, Wi-Fi, has made it possible for the infrastructures and systems in this field to evolve exponentially. Hence, research into gesture recognition was initiated using these networks. However, these systems had the disadvantage of requiring specialized devices or the need of modifying commercial approaches since they were unable to accurately detect hand movements.

Radio frequency identification (RFID) uses commercial devices of Ultra High Frequency (UHF). These systems are mainly used for the automatic management of objects and to control human activities [13–15]. RFID also uses phase shifts and the Doppler Effect and is inexpensive. The problem with the RIFD is the detection ranges as they are reduced to a few centimeters.

Currently, the use of RGB cameras is organized in an array of pixels where each one has its value. Based on this capture, the different algorithms are applied for the processing of images. An instance of how they can be applied is by using stereoscopic vision whereby the depth of the view can be calculated by obtaining two different views of the same scene, following the binocular system of the human eye. Active lighting can also be used, and depending on its projection, can establish the depth of the space. This technique is very similar to stereoscopic vision; in fact, it is considered a modification of structured vision.

The use of RGB-d cameras is one of the most common techniques used to obtain depth in video images in real time. These devices are able, through the emitter of infrared light (IR), to obtain the depth of an image. The process that these systems follow is: data acquisition, image pre-processing, segmentation, feature extraction and classification. These cameras began to drop in price when Microsoft's Kinect came on the market and their use increased [16]. A few years later, Microsoft released the Azure Kinect DK with many more sensors and speech models. This camera was developed for professional use employing AI to detect and study the movements of people and objects.

1.2. Relative Works

The progress of electronics and microelectronics has given a very important advance to the field of sensors. This is the case of the Leap Motion [17], a volumetric sensor that is capable of capturing the movement of the hands in three dimensions. There are different projects that use the Leap Motion Controller (LMC) for the recognition of different signs. These can be divided into three trends of use: the use of static signs (dactyl alphabet), the use of static and dynamic signs (words), and finally, only the use of dynamic signs (words).

Regarding the relative works of static signs, Funasaka used 24 static signs that correspond to the alphabet of the American Sign Language [18] (ASL), and the decision tree technique with an accuracy of 82.71%. On the other hand, G. Marin [19] combines LMC and Kinect to recognize 10 ASL signs using a support vector machine (SVM) with an accuracy of 80.86%.

Simos, in [20], used 24 static signs from the Greek Sign Language (GSL), working with multilayer perceptron (MLP), for hands and fingers independently. His accuracy results

were 99.08% and 98.96%, respectively. Mapari [21] worked with 32 ASL signs and numbers using MLP with an accuracy of around 90%. Vaitkevičius [22], conversely, worked with Virtual Reality (VR), using Leap Motion 24 ASL static letters in order to create sentences that he later recognized using linear regression analysis with an accuracy of 86.1%.

Among the researchers who developed programs using dynamic and static signs, Mohandes [23,24], for instance, worked with 28 signs of the Arabic Sign Language (ArSL) in a static way, except for two signs that are dynamic. In both experiments, different processing techniques were used, including the K-nearest neighbor algorithm (K-NN), hidden Markov model (HMM), Naïve Bayes and MLP, obtaining an accuracy of 97.1%, 97.7%, 98.3% and 99.1%, respectively. In [25], Hisham worked with 16 ArSL static words with different results depending on the techniques applied, obtaining an accuracy of 90.35% for neural network, 95.22% for K-NN, and between 89.12% and 90.78% for SVM according to the methodology. Hisham also worked with 20 ArSL dynamic words using dynamic time warping, (DTW), with an accuracy of 96.41%. Naglot used 26 ASL letters with MLP and an accuracy of 96.15% [26]. Chong used 26 ASL letters and 10 digits [27]. The results working only with SVM letters were 80.30% and with deep neural network (DNN), 93.81%. The results adding the 10 digits were 72.79% and 88.79% for SVM and DNN, respectively. Lee worked with the 26 ASL letters, two of which are dynamic, using different techniques [28]. For long short-term memory (LSTM), the accuracy was 97.96%, for SVM, 98.35%, and finally for recurrent neural network (RNN), 98.19%. In [29], Tao also used 26 ASL letters with convolutional neural network (CNN), with an accuracy ranging between 80.1% and 99.7%. In [30], Anwar worked with 26 Indonesian Sign Language (SIBI), applying the K-NN and SVM techniques, obtaining a result of 95.15% and 93.85% respectively. Alnahhas [31] worked with 15 ArSL words using LSTM with a result of 96%. Lastly, Avola [32] worked with 18 static signs and 12 dynamic signs using LSTM with an accuracy of 96.41%.

Finally, there are experiments that only used dynamic signs. Among others, in [33], Elons presented an investigation that used 50 words of the Arabic Sign Language ArSL, using MLP, obtaining an accuracy of 88%. On the other hand, Jenkins worked with 13 dynamic words making a comparison with different techniques [34]. With neural network he obtained an accuracy of 99.9%, with random forest, 99.7%, with SVM, 99.9%, K-NN, 98.7% and finally, with Naïve Bayes, a result of 96.4%.

1.3. Our Proposal

After studying the state of the art, it can be seen that the use of dynamic signs is the most realistic method to try to identify sign language; however, the number of words used is limited, in the best of cases, to 50 words. In this research, 176 dynamic words from the Spanish Sign Language (SSL) have been used. For the recognition of these words, the DTW has been used, achieving an accuracy of 95.17%. This article presents how the word database has been generated, the methodology used for the experiment, and the results offered by the applied techniques.

An important innovation is to demonstrate that models of signs done by a user can be used by different people while maintaining the level of accuracy. Then, the models can be generated by any person, and these models can be used by any different person. The proposal demonstrates that it is totally independent of the signer.

The objective of the research is to develop a recognition system for Spanish sign language using machine learning techniques. Regarding the research presented in this section, there are some gaps that this article addresses, e.g., the number of dynamic signs, because other related references work with a small dataset vs. this proposal (until 176 dynamic signs).

A second issue that this paper solves is the use of dynamic sign recognition with many words without losing accuracy. In the SSL, practically all the signs are dynamic, so their treatment is crucial when it comes to transcribing gestures into words.

Another aspect that the article addresses is the use of machine learning with respect to the neural networks used by the projects described later. The use of this technique substantially improves the accuracy results with regard to other related works.

1.4. Structure of the Paper

For the development of this article and to achieve the results obtained, a strict re-search process has been followed. In Section 2, the materials and methods necessary for the project have been analyzed, considering the use of a volumetric sensor such as the Leap Motion as a stable and economical option for the execution of the investigation. This section also includes the database of the samples used and the parameters of the volumetric sensor that have been used. The last part of this section describes the use of dynamic time warping (DTW) and how the classifier is implemented.

Section 3 shows the procedures used and describes the different quality parameters for the subsequent analysis of results.

The results are presented in Section 4 where the percentages of the different quality parameters of the research are shown.

In Section 5, the results obtained are compared in a table with those of other published research. Finally, Section 6 presents the conclusions with an accuracy percentage of 95.17% for 176 words, with the number of words implemented being greater than from other publications.

2. Materials and Methods

2.1. Introduccion

In the first part of this issue, we explain how the Leap Motion Controller works. At the second part, we have introduced a table with the 176 signed words separated into three categories: medical words, verbs, and everyday words. In the final part, the number of samples is commented. The following section explains how the data has been processed according to the position and movement of the hands. Pattern generation explains the methods used for generation. The following section explains how the comparator and data treatment work using the DTW technique. In the last section, the system is implemented using the different Machine Learning Python libraries.

2.2. Leap Motion Controller

This commercial device can track the movement of the forearms, hands and fingers in real time. It is a compact, versatile, and economical device that contains 2 cameras with an angle of 120° and three infrared LEDs. It can work at 200 fps (frames per second) and adapts to the amount of light available at all times. This ensures that the device maintains a constant image resolution. It has a USB 3.0 connection and is compatible with Windows, Linux, and Mac. The company that develops it provides the necessary APIs to program in Python, Java, C++, C#, Objective-C and JavaScript. To access the Leap Motion service, C is used.

The Leap Motion controller locates the position of the hands and fingers and transfers the frames to the equipment by means of the USB. On transfer, it returns a type of Hand object (see Figure 1). Each type of Hand object has different subclasses, Arm and Fingers. At the same time, each finger has subclasses of each of the phalanges of the fingers. The different parameters they provide are those related to direction, position, orientation, length, width, and speed.

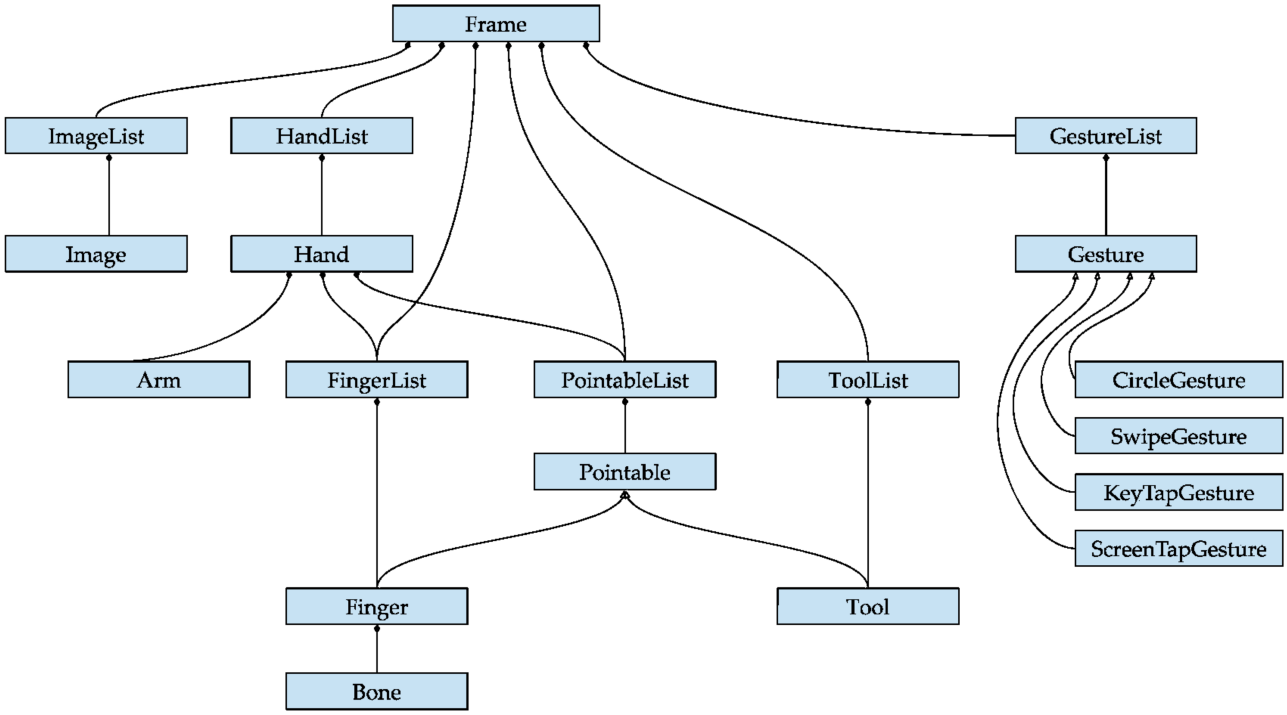


Figure 1. Frame Object of Leap Motion. Source available online: https://www.researchgate.net/figure/Frame-Object-of-Leap-Motion-Source_fig5_342433909 (accessed on 29 December 2022).

2.3. Acquisition

The implemented system has several parts. First, data is collected by using the leap motion sensor on the computer (see Figure 2).

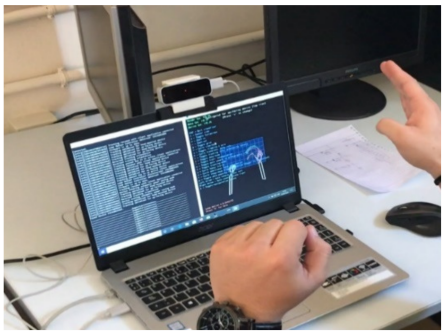


Figure 2. Recording of samples.

According to Figure 2, 176 dynamic signs have been recorded. The Spanish name of each sign (word) can be observed in Table 1. Each word has been recorded in 4 different sessions separated by at least 10 days between each session. This has been done to gain independence of the recorded data. The first three sessions were carried out by the same person whilst the fourth session was performed by 15 anonymous people, who are totally independent to the participant of first three sessions.

Table 1. Set of recording words.

Medical Words						
Allergy	Alzheimer	Ambulance	Anxiety	Asthma	Bacteria	Bladder
Blood	blood circulation	Blood Test	Breasts	Burp	Cancer	Care
Chest	Consult	Depression	Diabetes	Doctor	Fainting	Fatten
Fever	Gluteus	Headset	Health	Heart	Heart Attack	Hemorrhage
Hormone	Hospital	Ictus	Implant	Inflammation	Injection	Injury
Intestine	Jaw	Liver	Lumbago	Lungs	Mask	Medicines
Organs	Ovaries	Oxygen	Phlegm	Pressure	Prostate	Serum
Sickness	Stitches	Stomach Stress	Stress	Swelling	Urgency	Vaccine
Vagina						
Verbs						
I didn't know	I don't know	Is	To Ask	To Break	To Burn	To Cook
To Cure	To Disappear	To Do	To Eat	To Fall	To Feel	To Go
To Have	To Operate	To Reduce	To Rest	To Run	To Shower	To Size
To Take	To Take a walk	To Try	To Walk	To Work		
Everyday Words						
Accident	After	Air	All Right	Already	Also	Always
April	August	Back	Before	Car	Centre	Coat
Cold	Danger	Day	Deaf	December	Ear	Effort
Elbow	End	Example	Eyelid	Eyes	Family	Fat
February	Feet	Friday	Gloves	Good	Good Afternoon	Good Bye
Good Morning	Good Night	Hands	Head	High	Higher	Hour
How are you?	Hungry	Information	June	Last Night	Little	Lonely
Man	March	Midday	Moto	Mouth	Much	Mummy
Neck	Night	No	No Problem	Noise	Nothing	Now
October	Panic	Private	Rain	Regular	Risk	Saturday
Sensation	September	Sex	Shape	Shoulder	Something	Suffer
Sure	Thank You	They	Thin	To	Too	Trouble
Unusual	Upset	Wednesday	Without	Yesterday	Yours	

The implemented acquisition system uses Leap Motion as a hardware device and the tracking software offered by the manufacturer to extract the different parameters. In order to collect information from the hands, the Leap Motion is placed in front of the hands on a support designed for this purpose with a 3D printer. The connection of the Leap Motion with the equipment is made through a USB 3.0 BUS.

The manufacturer-supplied software is a service available on Microsoft Windows or a daemon on MAC and Linux. The current versions of this software offer an API developed in C language, LeapC, which allows access to the data of the service.

Next, a software was developed that works as a user interface so that deaf people could sign and, at the same time, record each of the samples necessary to create the dictionary of signs. In order to make the recordings as accurate as possible, a series of steps were implemented within this proposal to make recording easy for them and to avoid unwanted movement.

A text file is obtained for each word (dynamic sign). Each one of them is defined by 276 parameters or values. These values correspond to the movement of the hands and their position at any given time. At the same time, for the validation of each sign, another software capable of visualizing the movement of the hands and verifying their movement was developed.

At the end of this entire process, a total of 5780 samples were obtained. Each sample corresponds to a plain text file where the data of all the parameters of the LMC are recorded. In each file, there is a different number of frames according to the duration of each sign. In this case, the number of frames also depends on the sampling frequency. This frequency cannot be controlled to a fixed value because it will depend on hardware factors such as the type of port, microprocessor, etc., and on software factors such as the operating system. For example, the hello sign (see Figure 3) is sampled at 110 Frames per Second, fps, for each hand.

	frame_id	tracking_frame_id	timestamp	framerate	nHands	confidence	visible_time	id	type	position.x	...	arm.next_joint.x	arm.next_joint.y	arm
0	467633	467628	64943015087	110.6489	1	1.0	6019647	338	1	73.2755	...	108.3714	395.1777	
1	467634	467629	64943024141	110.6400	1	1.0	6028701	338	1	73.2934	...	108.3580	395.1546	
2	467635	467630	64943033112	110.6765	1	1.0	6037672	338	1	73.3140	...	108.3583	395.1292	
3	467636	467631	64943042293	110.5977	1	1.0	6046853	338	1	73.3615	...	108.3769	395.0760	
4	467637	467632	64943051175	110.6841	1	1.0	6055735	338	1	73.3874	...	108.3793	395.0068	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	
95	467728	467723	64943873825	110.5848	1	1.0	6878385	338	1	129.5075	...	142.4177	201.8013	
96	467729	467724	64943882849	110.5950	1	1.0	6887409	338	1	129.5740	...	142.5009	201.9342	
97	467730	467725	64943891754	110.6691	1	1.0	6896314	338	1	129.0080	...	142.5626	201.4753	
98	467731	467726	64943900852	110.6355	1	1.0	6905412	338	1	129.1060	...	142.7790	201.6937	
99	467732	467727	64943909907	110.6267	1	1.0	6914467	338	1	129.2675	...	143.1082	201.9641	

100 rows × 276 columns

Figure 3. Hello sign data.

Finally, the dataset is composed by 5780 samples, of which 3520 samples have been applied for training and the rest, 2260 samples, for checking the outputs and defining the accuracy.

Spanish sign language omits all prepositions, and a sentence consists of a verb (always in infinitive) and some nouns and adjectives. In this project, authors worked on the recognition of those signs, individually. In the future, a module for processing natural language will be added. The first step is to know whether this proposal can recognize those dynamic words (signs) better than the state of the art and with a larger set of dynamic signs.

The acquisition of data has been applied under the Declaration of Helsinki as a statement of ethical principles for the research involving human subjects on identifiable human data since the sign language is performed by humans. The data is acquired under the consent of the user by anonymous caption. Moreover, the sensor only acquires the movement of the hands by 3D series temporal, and with that information it is very difficult to identify the user. The state of the art does not show references about the human identification by 3D series temporal of the hand movement.

2.4. Data Preprocessing

The proposal has been developed using the Python programming language. This language provides a large set of libraries in the field of AI.

First, the signs corresponding to a file containing multivariable time series are loaded into the memory, that is, each sign has an associated matrix where the y-axis is the number of frames, and the x-axis are the variables. Each variable corresponds to a field in the Leap Motion. Each sign has a variable number of frames (see Figure 3), since each one has a different duration; obviously, they all have the same number of variables.

The parameters come from the Frame object which is, in turn, the root of the data model and provides access to all tracked entities. This way, a frame is created at each update interval. This frame contains attached lists of the hands and fingers corresponding to the instant in which it was created. In addition, the device allows the acquisition of the fingers of a specific hand from the corresponding Hand object. The basic features common to fingers are defined in the Finger class. On the other hand, the Arm object describes the position, direction, and orientation of the arm to which a hand is attached. Finally, the Bone object represents the position and orientation of a bone. The bones tracked include the metacarpals and the phalanges of the fingers (see Figure 4).

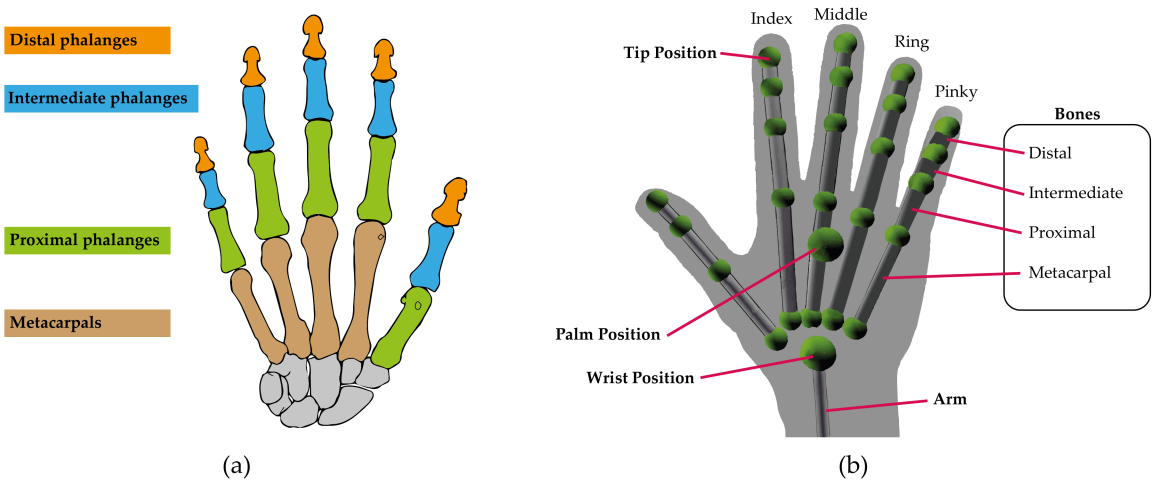


Figure 4. Tracking of the hands skeletal model (**a**) Object Bone. Source available online: https://developer-archive.leapmotion.com/documentation/javascript/devguide/Leap_Overview.html (accessed on 29 December 2022) (**b**) Object Hand. Source available online: https://ieeexplore.ieee.org/abstract/document/8538425/figures#figures (accessed on 29 December 2022).

In addition to the tracking model features, Leap Motion incorporates Image objects, which provide the raw sensor data and calibration grid for the device's cameras.

All the elements presented in Table A1 are derived from the model described above. As can be seen, regardless of elements that provide time information (timestamp, visible time), aspects such as speed, orientation, direction, normal and width of the palm are also considered (see Figure 5); identifiers of frame (id), hand and finger; distance and angles between fingers, and finally, widths, rotations, and joints of all the bones.



Figure 5. Palm Movement. Source available online: https://developer-archive.leapmotion.com/documentation/javascript/devguide/Leap_Overview.html (accessed on 29 December 2022).

In cases of occlusions with the hands, a predictive method is performed, which is quantified in terms of reliability in the confidence parameter. The coordinate axes are established according to Figure 6. Logically, their directions will be defined according to the placement of the device.

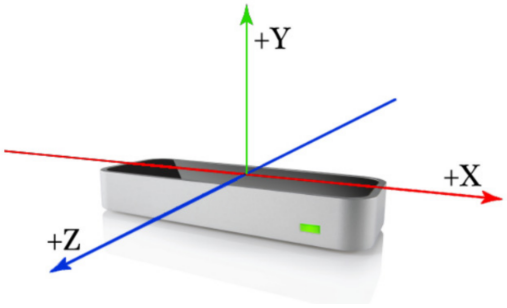


Figure 6. Leap Motion axes. Source available online: https://developer-archive.leapmotion.com/documentation/javascript/devguide/Leap_Overview.html (accessed on 29 December 2022).

2.5. Pattern Generation

2.5.1. Parameters

If all the parameters are used, the system can be slowed down in excess. Next, a study was carried out where it was verified that not all the fields are necessary. Finally, it was found that the detection was 100% reliable.

The sensor generates the information of the movement of the hands in a text type file which contains 276 parameters (see Table A1). In order to make the system faster, the number of parameters was reduced. Many of these parameters are not necessary, such as the first one that indicates the frame number. After analyzing all the parameters and carrying out different tests to select the most important parameters, 74 were selected (see Table 2). These parameters are the ones considered fundamental. The number corresponds directly to Table A1 in the Appendix A.

Table 2. Number of parameters used for the system according to Table A1.

Parameters								
4	8	9	10	11	27	28	29	32
41	42	43	52	53	54	63	64	65
74	75	76	79	88	89	90	99	100
101	110	111	112	121	122	123	126	135
136	137	146	147	148	157	158	159	168
169	170	173	182	183	184	193	194	195
204	205	206	215	216	217	220	229	230
231	240	241	242	251	252	253	262	263
264	272							

As can be seen, the parameters corresponding to the rotations in the x, y and z axes of all the fingers have been chosen. To reduce the complexity of the system, the hands were separated during the generation of the patterns.

The selection of parameters is based on keeping the rotation, translation, and size invariance. Thus, the proposal can be applied in different places, with a different positioning of the sensor and environment conditions. Therefore, the user can use the device in any situation. This will be important for the recording of the different sessions of the dataset.

2.5.2. Generation

The independence of the samples has been established by making the recordings as follows: three sessions of 10 samples each were recorded, plus an additional session with samples recorded by people different to the participant of the first three sessions. These sessions were held at different times, that is, with a temporary space of at least 10 days. Once an analysis of the time of the project was carried out, it was considered that 176 words could be recorded. To analyze the evolution of the process, the results were verified in sets of 50 words. Checks were made at 50, 100 and, finally, at 176 words. Finally, the data of the

fourth session, consisting of 50 words and 10 samples per word, have been used to validate all processes, with 15 anonymous users who did not participate in the first three sessions.

Initially, the sampling frequency is variable depending on the hardware and software conditions of the equipment. Before generating the patterns, it is necessary to equalize the size of the files, so that they have the same number of frames in each of them. Each file corresponds to a sample of each word.

The person who recorded each of the words was a specialist in Spanish sign language. To create the patterns, 20 samples from 2 different sessions have been used. One word Q discomposed by 20 observations from *Q1* to *Q20*.

For the generation of the pattern, the shortest series was searched first, causing the rest of the recordings to be compressed to the size of that series. They were then added together and divided by the number of existing samples creating the average pattern.

$$\frac{\sum_{i=1}^{i=20} Q_i(x)}{20} \tag{1}$$

In the end, the pattern keeps the form of the signed word and also adds features of each of the references, adding robustness to the system, so that it continues to be reliable for different sessions of the same user, as well as between different users.

It is very important that when generating the patterns, it is taken into account that the recordings are made correctly, verifying that they are correctly recorded and that the files have a similar size.

2.6. Comparator

2.6.1. Introduction to DTW

Dynamic time warping (DTW) is a classification technique that is used in many fields of research. It is widely used to compare time series of different lengths. A field where it is widely used is in voice recognition where each person, when speaking, can say the same thing, yet not everyone speaks at the same speed. With DTW this time warping is resolved.

There are instances where DTW is used. For example, Tuzcu [35] uses it for the automatic recognition of ECG signals; Legrand [36], for chromosome recognition; Kovacs-Vajna [37], in fingerprint recognition; whilst Rath [38] uses it in the development of the recognition of handwritten documents. Research using DTW for signature recognition can also be found in [39,40]. Within the investigations based on voice recognition are [41,42]. A field in which it is also used is in the recognition of facial [43] and body gestures [44]. In the recognition of gestures there are many investigations where some of them use the Kinect [45–48]. Finally, within the research that deals with the recognition of sign language are [49–51].

2.6.2. Development of the DTW

When a person is signing, the speed with which they move their hands changes not only with other people, but with themselves, in the same way that sometimes a person speaks faster or slower. In order to compare time series of different lengths, the DTW technique was used. If there are 2 series in time K and T, with their lengths M and N and considering that each frame captured by the sensor, as an element of the function, what remains for all the captured frames is:

$$K = k_1, k_2, \ldots \ldots, k_i, \ldots \ldots, k_n \tag{2}$$

$$T = t_1, t_2, \ldots \ldots, t_i, \ldots \ldots, t_n \tag{3}$$

Assuming that k = t, it is not necessary to calculate the distance of the 2 sequences. If $k \neq t$ the alignment is further apart and there is no other choice but to try to align both sequences. To align both sequences it is necessary to build an $M \times N$ matrix, whereby each element of a matrix (I, J) represents the point Ki and Tj, the alignment.

Now the path is defined as a deformation path of the regular path, and W is used (see Figure 7) to indicate that the KK element of W with the final result of Wk = (i,j)k, at the end the following is obtained:

$$W = w1, w2, \ldots, wk, \ldots, wk \qquad \max(m,n) \leq K < m + n - 1 \tag{4}$$

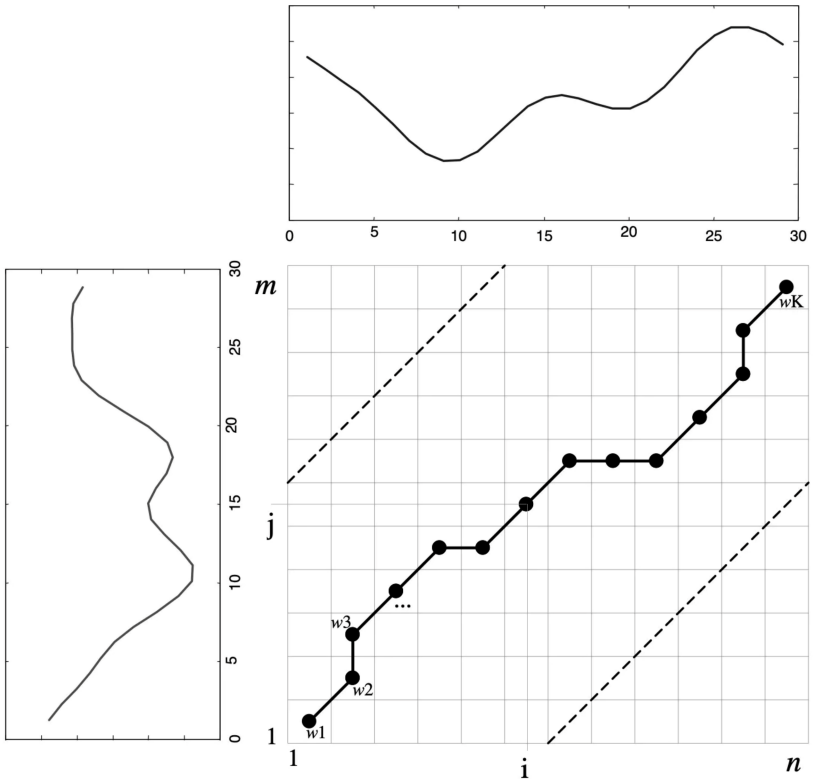


Figure 7. Example of a warping path. Source available online: https://blog.csdn.net/niyanghuahao/article/details/78612157?locationNum=9&fps=1. (accessed on 29 December 2022).

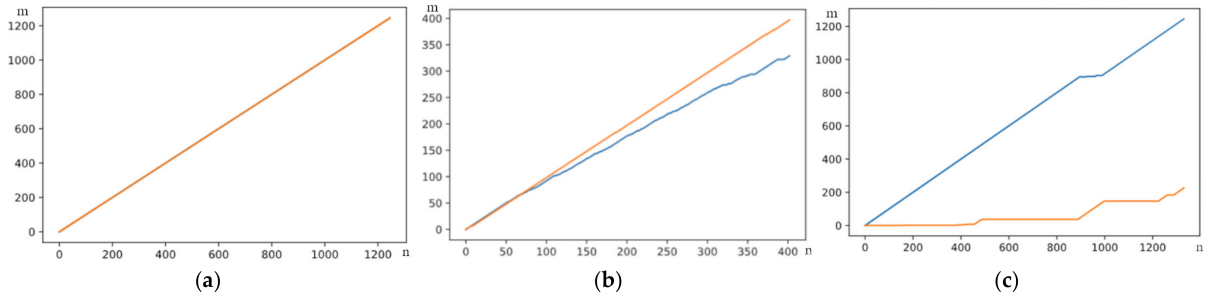
At the same time, the following conditions must be met:

1. Boundary condition; w1 = (1, 1) y wK = (m, n)
2. Continuity; Si wk-1 = (A′, B′), then the next point W for the path k = (A, B) must comply (A-A′) <= 1 y (B′) <= 1
3. Monotone; Si wk-1 = (A′, B′), then the next point W for the path k = (A, B) must comply 0 <= (A-A′) y 0 <= (B-B′).

The final objective is to extend and shorten the two time sequences and establish the shortest distance, so that in the end the distance γ(i,j) is minimal.

$$\gamma(i,j) = d(qi,cj) + \min\{\gamma(i-1,j-1), \gamma(i-1,j), \gamma(i,j-1)\} \tag{5}$$

In the graphic example of Figure 8a, a path generated by the DTW is shown, corresponding to a word. Figure 8b represents a sign of the same class, and it can be verified that both are similar. On the other hand, in Figure 8c, it can be seen how a different sign of the pattern behaves.



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

Figure 8. Comparator DTW. (**a**) Word Path of Pattern. (**b**) Comparison of a word (blue) sample with its own pattern (orange). (**c**) Comparison of a word (orange) other than the pattern (blue).

2.6.3. Comparator Scheme

The algorithm used to compare the different signs follows the structure of Figure 9. Each one of the 10 samples of each word is compared with the patterns. The output will show the result for the shortest distance from the input.

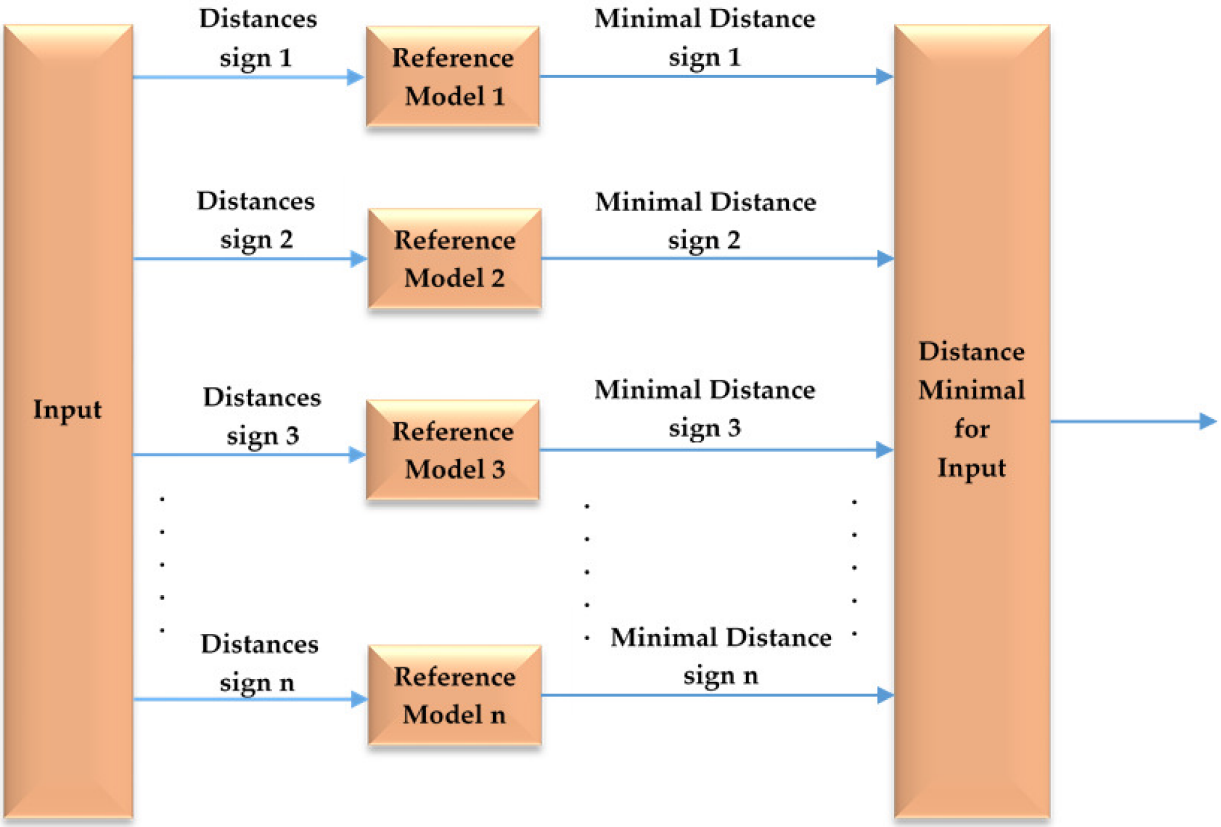


Figure 9. Comparator scheme.

The *"Reference Model"* are each of the patterns generated above, using 20 samples from 2 independent sessions. Each session has a set of 10 samples.

Since the sampling frequency is high, before entering the data in the comparator so that the system does not slow down and at the same time eliminate noise, a decimation is performed on the samples and patterns.

2.7. Implementation

For the implementation, a set of techniques have been chosen using the DTW as a comparison metric. DTW is an algorithm that allows the comparison of time series of different lengths. This algorithm is straight forward to implement, and much information

is open for use by researchers. Specifically, in Python there is a package called tslearn that already has this algorithm implemented. This package has already been tested and proven by many users. Finally, note that the implementation has been carried out using machine learning. The decision to use this type of technology was made because other investigations already had neural networks implemented and their results could be improved with this technique.

The Python packages used in the investigation are numpy, pandas, os, tslearn, and sklearn. Numpy and pandas are used for data loading and processing. The package helps manage files. This module is very important since, as mentioned in Section 2, each of the samples corresponds to a file. The tslearn package provides automatic processing tools for time series analysis. Specifically, the tslearn.metrics.dtw library has been used. To generate the confusion matrix and obtain the quality parameters of the system, accuracy, F1 Score, recall and precision, the sklearn package has been used.

3. Experimental Methodologies

3.1. Procedures

At first, different greetings were recorded: good morning, good afternoon, good evening, and hello. From the beginning, the aim was to reach a minimum of words. Due to time and progress, the final goal was 176 words with different meaning, whereby most of those words aimed at establishing a medical conversation between a deaf person and a doctor. To check the evolution, 50-word tests were carried out. The tests were carried out at 50, 100 and, finally, at 176 words. Finally, the samples (signs) from the fourth session will be applied for testing purposes only, and hence, to validate the proposal and demonstrate that the models generated by one person can be used by different people whilst the accuracy of the proposal is similar.

3.2. System Quality

In order to quantify the experiments performed with the 3-word sessions, quality measures need to be used. To obtain the results, the Python Scikit-Learn libraries were used.

In this way, 10 samples were obtained for each of the 3 sessions. Two sessions are used to generate the patterns and 1 session to compare it within the system. This ensures session invariance. Finally, at the output of the comparator, the confusion matrix and the quality parameters that provide us with the efficiency of the system are obtained (see Figure 10).

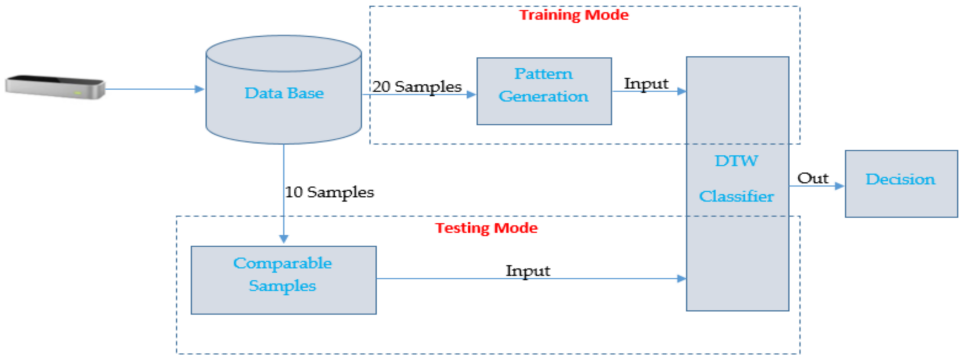


Figure 10. Implemented system.

In addition, the proposal performed a fourth session by anonymous users different to the user from the previous first three sessions. The session is composed by 50 words and 10 repetitions per word. Hence, the proposal can validate the models generated by each sign and their use for different people.

3.2.1. Confusion Matrix

From the confusion matrices, the different parameters necessary to measure the reliability and quality of the system are calculated. Figure A1 shows how the confusion matrix behaves. The diagonal shows the success of each of the words by itself. The green boxes correspond to several hits from 8 to 10, both inclusive, whilst the orange boxes show a hit rate of 5 to 7, both inclusive. Meanwhile, the boxes marked in red show a success rate of less than 4. Off-diagonal, it can be verified how there are different crosswords where the system does not recognize the pattern of the same word but does recognize the pattern of another word.

In Figure 11, Reality corresponds to the input and Prediction corresponds to the output. The different parameters are described below:

1. False Positive [FP]; A False Positive occurs when the model expects an output, and it does not occur.
2. False Negative [FN]; A False Negative occurs when the model does not expect an output that is eventually produced.
3. True Positive [TP]; A True Positive is when the model expects an output, and it does.
4. True Negative [TN]; A True Negative is when the model does not expect an output and it does not eventually produce one.

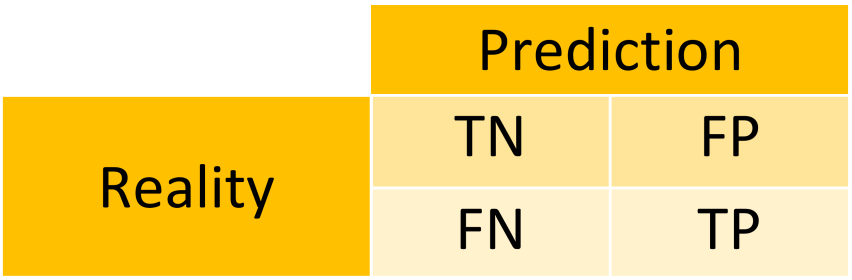


Figure 11. Graphic model of the parameters.

3.2.2. Precision

Using the accuracy metric, the quality of the model can be measured. This returns the percentage of hits that are expected.

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \times 100 \tag{6}$$

3.2.3. Recall

This measure informs about the quantity that the model is able to identify.

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \times 100 \tag{7}$$

3.2.4. F1

This feature combines the precision and recall parameters. This parameter helps compare the combined performance of both parameters. It is calculated by performing the harmonic mean of precision and recall.

$$\text{F1} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \times 100 \tag{8}$$

3.2.5. Accuracy

This parameter measures the system's percentage of success and its formula would be:

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \times 100 \quad (9)$$

4. Results

4.1. Experiment

There are two fundamental aspects to carry out the experiment: to see the success of the set of words, and to see the evolution of success as the number of words increases.

In the initial phase, the recordings were made, and the success rate and the possible word crossings were checked. For the first 50 words, an accuracy of 98.80% was obtained. The quality parameters obtained from the matrices (see Figures A1–A3) and the parameters are shown in Table 3. With 100 words, the accuracy was 97.60%, where its confusion matrix can be verified in Figure A2. The system dropped approximately 1 point with the addition of 50 words. Finally, it was tested with the total set of the database of 176 words. The accuracy of the implemented system dropped to 95.17%, with its confusion matrix in Figure A3.

Table 3. Quality parameters.

N° Words	Precision	Recall	F1
50	98.93%	98.80%	98.80%
100	97.96%	97.60%	97.59%
176	96.32%	95.17%	95.02%

4.2. Experiment for Validation

In order to show the robustness of the proposal and validate the use of this approach, a fourth session of data was built. Fifty words with 10 repetitions per word were recorded by 15 anonymous users, totally different to the user of the three first sessions. The models built for two of the three first sessions were used for this fourth session of samples. The results are shown in Table 4. The accuracy of the experiment was up to 94.80%.

Table 4. Validation of sign models.

N° Words	Precision	Recall	F1
50	95.34%	94.80%	94.77%

5. Discussion

The results show that the efficiency of the system is above 95% in all its phases. As words are added, the system's efficiency decreases, which may become an issue in recognizing some words. The approach has been validated by a session carried out by different users than that for building the models of each sign or word. The quality metric decreased 4% for accuracy, 3.59% for precision, 4% for recall and 4.03% for F1 between the user that built the models of signs and the new 15 anonymous users, who have validated those models. These results show that accuracy slightly decreases (between 3.59% and 4.03%, according to the quality metric) for different users or signers, and the models are independent of the person doing them. Therefore, this approach has inter-user invariance.

On the other hand, Table 5 compares the results obtained in this paper with the results of other publications. It should be noted that regardless of the number of words used in this paper, there are several differences with the research described in Table 5. These proposals are aimed at the three trends indicated: use of static signs, use of static and dynamic signs, and use of dynamic signs. The limitation in the number of signs is observed. This research

proposes a system based on a pattern generator and the DTW to demonstrate that the number of words can be increased while maintaining success rates.

Table 5. Set of results of papers that using LMC.

Reference	Gestures	Data Set	Classifier	Accuracy
		Static Signs		
[18]	24 ASL Letters	unclear	Decision tree	82.71%
[19] *	10 ASL	14 people × 10 sets × 10 samples	SVM	80.86%
[20]	24 GSL	6 people × 10 sets	MLP (boneTraslation)	99.08%
			MLP (palmTraslation)	98.96%
[21]	32 ASL	146 people × 1 sample/letter	MLP	90%
[22]	24 ASL Letters Use sentences	12 people × 10 samples	Linear Regression Analysis	86.1%
		Static and Dynamic Signs		
[23]	28 ArSL	10 samples/letter	KNN	97.1%
			HMM	97.7%
[24]	28 ArSL	10 samples/letter	Naïve Bayes	98.3%
			MLP	99.1%
[25] *	16 ArSL Static Words 20 Dynamic Words	2 people × 200 samples	Static	
			Neural Network	90.35%
			K-NN	95.22%
			SVM (RBF Kernel)	89.12%
			SVM (Poly Kernel)	90.78%
			Dynamic	
			DTW	96.41%
[26]	26 ASL letters	4 people × 20 samples/letter	MLP-BP	96.15%
[27]	26 ASL letters 10 digits	12 people Unclear samples	SVM (letters)	80.30%
			DNN (letters)	93.81%
			SVM (total)	72.79%
			DNN (total)	88.79%
[28]	26 ASL letters	100 people × 1 samples/letter	LSTM	97.96%
			SVM	98.35%
			RNN	98.19%
[29]	26 ASL letters	5 sets × 450 samples/letter	CNN	80.1–99.7%
[30]	26 SIBI letters	5 people × 10 samples/letter	K-NN	95.15%
			SVM	93.85%
[31]	15 ArSL words	5 people × 10 samples/letter	LSTM	96%
[32]	18 static 12 dynamic	20 people × 60 samples/word	LSTM	96.41%

Table 5. *Cont.*

Reference	Gestures	Data Set	Classifier	Accuracy
		Dynamic Signs		
[33]	50 ArSL	4 people × 1 set	MLP	88%
[34]	13 ASL words	10 samples	Neural Network	99.9%
			Random Forest	99.7%
			SVM	99.9%
			K-NN	98.7%
			Naïve Bayes	96.4%
This Paper	50 SSL words	1 people × 30 samples/word	DTW	98.80%
This Paper	100 SSL words	1 people × 30 samples/word	DTW	97.60%
This Paper	176 SSL words	1 people × 30 samples/word	DTW	95.17%
This Paper	50 SSL words	15 different people × 10 samples/word	DTW	94.80%

* Papers that used Kinect and LMC.

It should be remembered that 20 samples from two independent sessions were used to create the patterns. If someone wants to improve efficiency so that the pattern picks up hand movements better, the number of samples to generate a pattern should be increased. With 50 samples recorded in different sessions and with different people, the pattern would be more robust, and the efficiency of the system would improve significantly.

It should be noted that the proposal improves against different Deep Learning techniques and, in general, machine learning, which can obtain good results, but for a limited number of words, between 10 and 50 signs and/or words. This proposal includes 176 dynamic words, thus demonstrating that in comparison to dynamic signs, this proposal maintains a positive robustness.

In future research, the aim will be to increase the number of words with problems related to similar signs that require another type of treatment, for instance, the case of the words disappearing and breaking. Both signs have a similar behavior regarding the movement of the hands, meaning that context is sometimes needed to be able to identify them separately.

In Spanish sign language, there are words that are signed exactly the same and that only change their meaning due to the shape of the face. In those instances, it would be useful to include a camera that would interpret the shape of the face if necessary.

Finally, note that in order for a system like this to be useful in real life, it would require approximately 1000 words of common use. For this, efficiency would have to be improved, as mentioned above.

6. Conclusions

This paper shows a word recognition system for the Spanish sign language. A pat-tern creator is applied, and DTW is used to establish the correct word, achieving a success rate of 95.17% for a total of 176 dynamic signs, improving the number of words of the state of the art, which were between 10 and 50 dynamic signs. At the same time, the recognition of dynamic signs has been improved, applying automatic recognition techniques using the DTW. In addition, it improves systems using deep learning and machine learning techniques, in general, showing this proposal to be more robust than the techniques used until now.

This proposal has validated the use of sign models developed by a user and then used by different users or signers. Therefore, the proposal shows an invariance inter-user. This is an added value of the proposal because it keeps the accuracy of 94.8% with inter-user validation vs. the 98.80% for the inter-session validation with the same user with 50 words.

From a practical point of view, the model created by one user or signer can be used by any other user or signer.

Author Contributions: Conceptualization, J.G.-R. and C.M.T.-G.; methodology, J.G.-R.; software, A.P.-R.; validation, C.M.T.-G. and J.B.A.-H.; formal analysis, A.P.-R.; investigation, J.G.-R.; resources, J.B.A.-H.; data curation, A.P.-R.; writing—original draft preparation, J.G.-R.; writing—review and editing, J.G.-R., C.M.T.-G. and J.B.A.-H.; visualization, J.G.-R., C.M.T.-G.; supervision, C.M.T.-G.; project administration, J.G.-R. and C.M.T.-G.; funding acquisition, J.G.-R. and C.M.T.-G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was granted and funded by "Fundación Indra" and "Fundación Universia", under the award "Ayudas a Proyectos Inclusivos" in its 2018 Edition.

Institutional Review Board Statement: The study was conducted in accordance with the Declaration of Helsinki, but ethical review and approval were waived for this study due to it is not possible to identify persons with these data.

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. The data are not publicly available due to they is being analyzed for future publications before to openly available.

Acknowledgments: Authors show their acknowledgment to "ASOR Las Palmas" (Las Palmas de Gran Canaria, Spain), for its support, their shared knowledge and help provided by its staff.

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

Appendix A

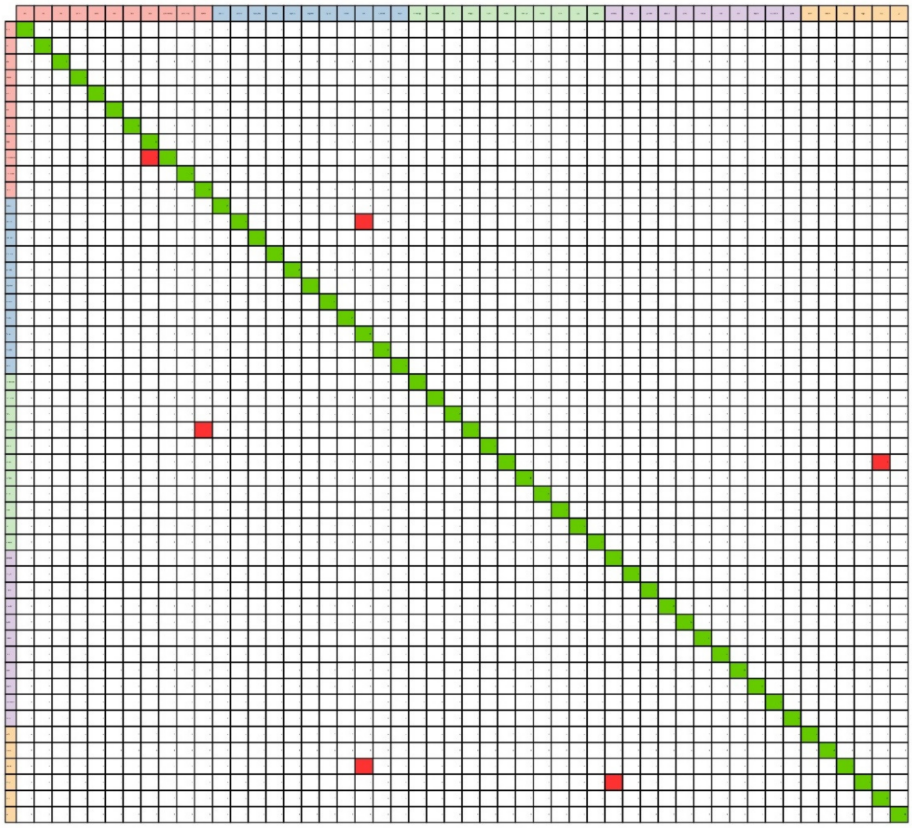


Figure A1. Confusion matrix of 50 words.

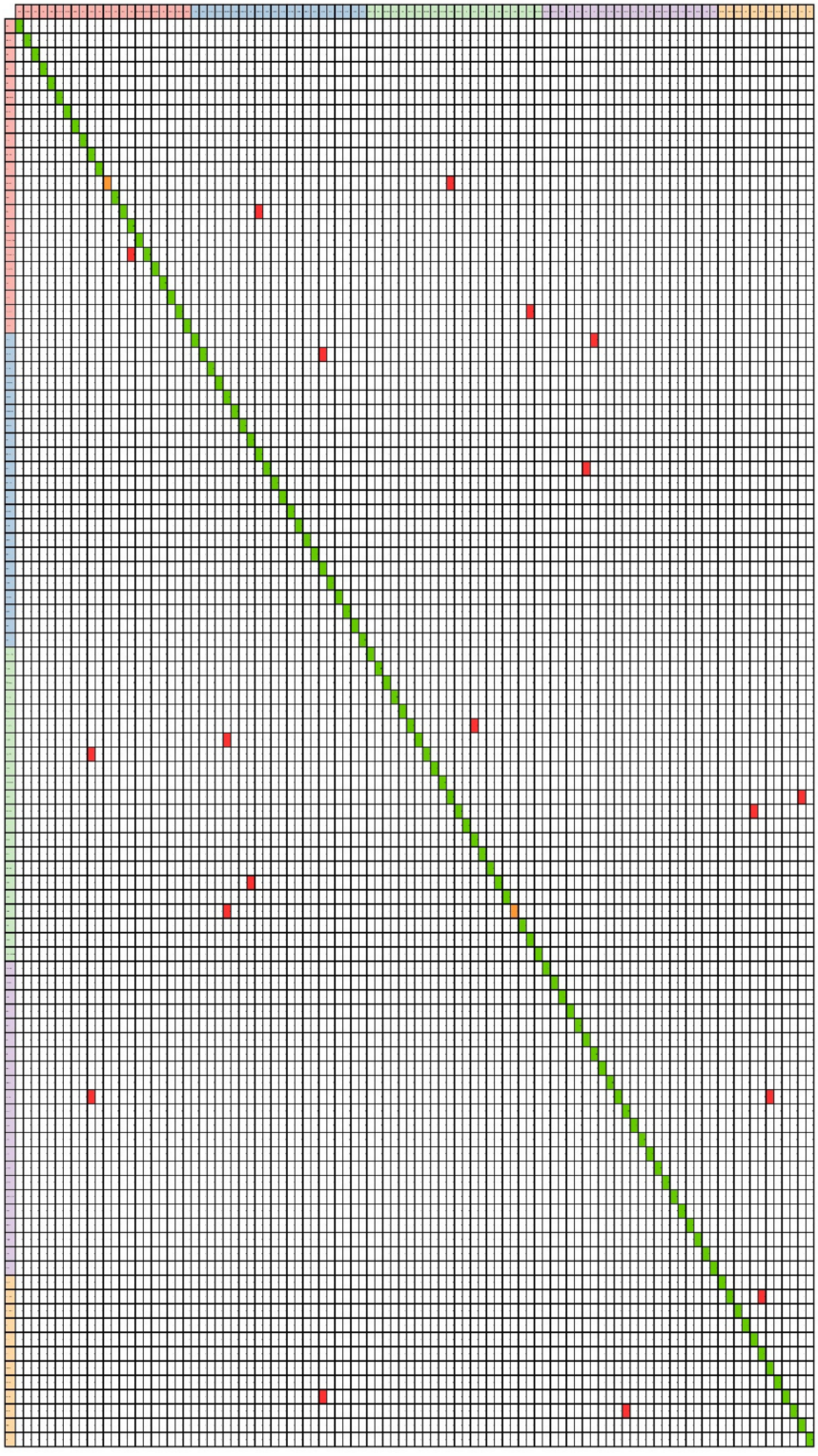


Figure A2. Confusion matrix of 100 words.

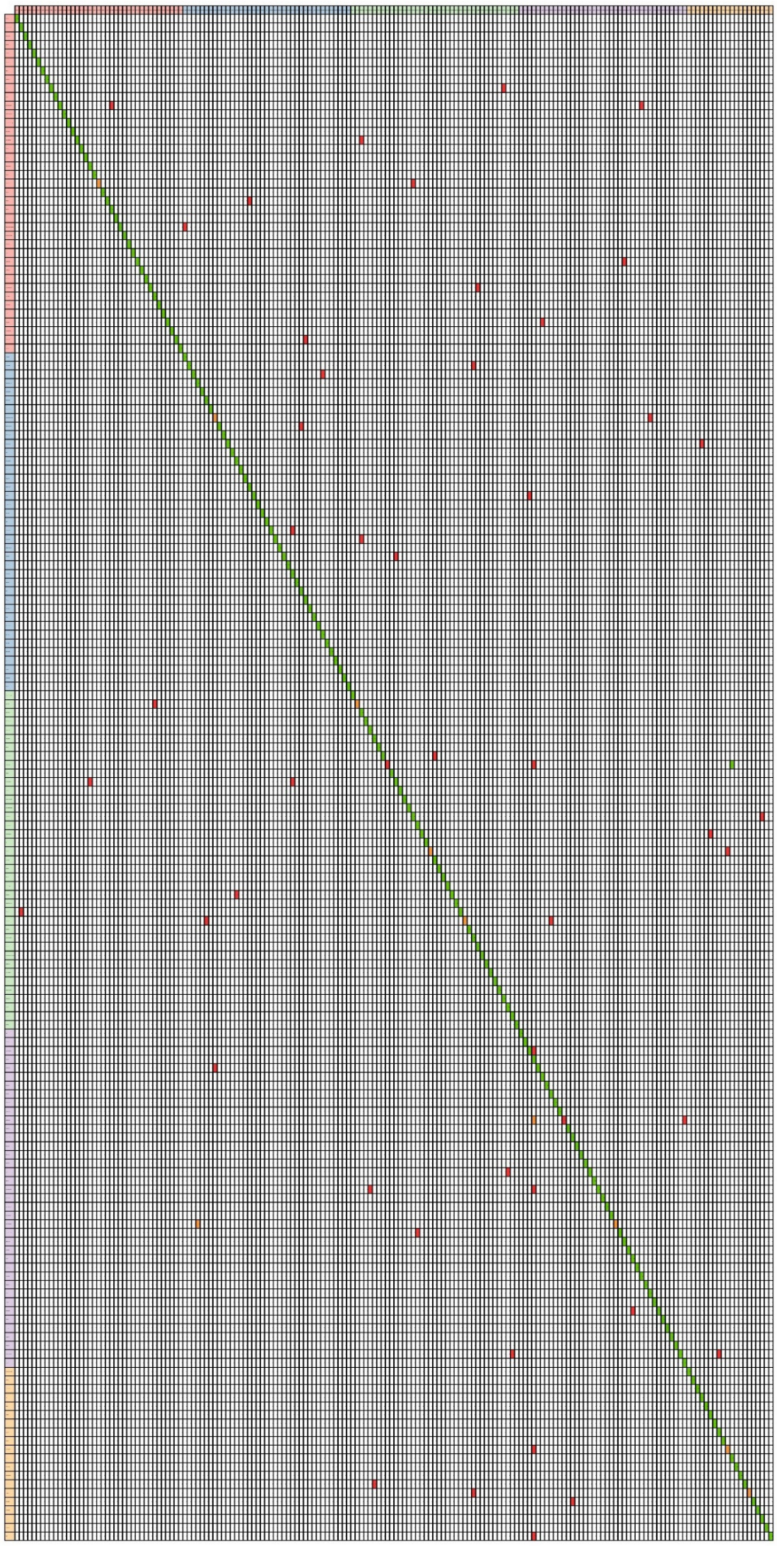


Figure A3. Confusion matrix of 176 words.

Table A1. Leap Motion parameter set.

	0	1	2	3	4	5	6	7	8	9
0	info.frame_id	tracking_frame_id	info.timestamp	framerate	nHands	confidence	visible_time	id	type	palm.position.x
1	palm.position.y	palm.position.z	pinch_distance	grab_angle	pinch_strength	grab_strength	palm.velocity.x	palm.velocity.y	palm.velocity.z	palm.normal.x
2	palm.normal.y	palm.normal.z	palm.width	palm.direction.x	palm.direction.y	palm.direction.z	palm.orientation.w	palm.orientation.x	palm.orientation.y	palm.orientation.z
3	0	digits(0).finger_id	Digits(0).is_extended	Digits(0).bones(0).width	Digits(0).bones(0).next_joint.x	Digits(0).bones(0).next_joint.y	Digits(0).bones(0).next_joint.z	Digits(0).bones(0).prev_joint.x	Digits(0).bones(0).prev_joint.y	Digits(0).bones(0).prev_joint.z
4	Digits(0).bones(0).rotation.w	Digits(0).bones(0).rotation.x	Digits(0).bones(0).rotation.y	Digits(0).bones(0).rotation.z	digits(0).bones(1).width	digits(0).bones(1).next_joint.x	Digits(0).bones(1).next_joint.y	Digits(0).bones(1).next_joint.z	digits(0).bones(1).prev_joint.x	Digits(0).bones(1).prev_joint.y
5	Digits(0).bones(1).prev_joint.z	Digits(0).bones(1).rotation.w	Digits(0).bones(1).rotation.x	Digits(0).bones(1).rotation.y	Digits(0).bones(1).rotation.z	Digits(0).bones(2).width	Digits(0).bones(2).next_joint.x	Digits(0).bones(2).next_joint.y	Digits(0).bones(2).next_joint.z	Digits(0).bones(2).prev_joint.x
6	Digits(0).bones(2).prev_joint.y	Digits(0).bones(2).prev_joint.z	Digits(0).bones(2).rotation.w	Digits(0).bones(2).rotation.x	Digits(0).bones(2).rotation.y	Digits(0).bones(2).rotation.z	Digits(0).bones(3).width	Digits(0).bones(3).next_joint.x	Digits(0).bones(3).next_joint.y	Digits(0).bones(3).next_joint.z
7	Digits(0).bones(3).prev_joint.x	Digits(0).bones(3).prev_joint.y	Digits(0).bones(3).prev_joint.z	Digits(0).bones(3).rotation.w	Digits(0).bones(3).rotation.x	Digits(0).bones(3).rotation.y	Digits(0).bones(3).rotation.z	1	Digits(1).finger_id	Digits(1).is_extended
8	Digits(1).bones(0).width	Digits(1).bones(0).next_joint.x	Digits(1).bones(0).next_joint.y	Digits(1).bones(0).next_joint.z	Digits(1).bones(0).prev_joint.x	Digits(1).bones(0).prev_joint.y	Digits(1).bones(0).prev_joint.z	Digits(1).bones(0).rotation.w	Digits(1).bones(0).rotation.x	Digits(1).bones(0).rotation.y
9	Digits(1).bones(0).rotation.z	Digits(1).bones(1).width	Digits(1).bones(1).next_joint.x	Digits(1).bones(1).next_joint.y	Digits(1).bones(1).next_joint.z	Digits(1).bones(1).prev_joint.x	Digits(1).bones(1).prev_joint.y	Digits(1).bones(1).prev_joint.z	Digits(1).bones(1).rotation.w	Digits(1).bones(1).rotation.x
10	Digits(1).bones(1).rotation.y	Digits(1).bones(1).rotation.z	Digits(1).bones(2).width	Digits(1).bones(2).next_joint.x	Digits(1).bones(2).next_joint.y	Digits(1).bones(2).next_joint.z	Digits(1).bones(2).prev_joint.x	Digits(1).bones(2).prev_joint.y	Digits(1).bones(2).prev_joint.z	Digits(1).bones(2).rotation.w
11	Digits(1).bones(2).rotation.x	Digits(1).bones(2).rotation.y	Digits(1).bones(2).rotation.z	Digits(1).bones(3).width	Digits(1).bones(3).next_joint.x	Digits(1).bones(3).next_joint.y	Digits(1).bones(3).next_joint.z	Digits(1).bones(3).prev_joint.x	Digits(1).bones(3).prev_joint.y	Digits(1).bones(3).prev_joint.z
12	Digits(1).bones(3).rotation.w	Digits(1).bones(3).rotation.x	Digits(1).bones(3).rotation.y	Digits(1).bones(3).rotation.z	2	Digits(2).finger_id	Digits(2).is_extended	Digits(2).bones(0).width	Digits(2).bones(0).next_joint.x	Digits(2).bones(0).next_joint.y
13	Digits(2).bones(0).next_joint.z	Digits(2).bones(0).prev_joint.x	Digits(2).bones(0).prev_joint.y	Digits(2).bones(0).prev_joint.z	Digits(2).bones(0).rotation.w	Digits(2).bones(0).rotation.x	Digits(2).bones(0).rotation.y	Digits(2).bones(0).rotation.z	Digits(2).bones(1).width	Digits(2).bones(1).next_joint.x
14	Digits(2).bones(1).next_joint.y	Digits(2).bones(1).next_joint.z	Digits(2).bones(1).prev_joint.x	Digits(2).bones(1).prev_joint.y	Digits(2).bones(1).prev_joint.z	Digits(2).bones(1).rotation.w	Digits(2).bones(1).rotation.x	Digits(2).bones(1).rotation.y	Digits(2).bones(1).rotation.z	Digits(2).bones(2).width
15	Digits(2).bones(2).next_joint.x	Digits(2).bones(2).next_joint.y	digits(2).bones(2).next_joint.z	digits(2).bones(2).prev_joint.x	digits(2).bones(2).prev_joint.y	digits(2).bones(2).prev_joint.z	digits(2).bones(2).rotation.w	digits(2).bones(2).rotation.x	digits(2).bones(2).rotation.y	digits(2).bones(2).rotation.z
16	digits(2).bones(3).width	digits(2).bones(3).next_joint.x	digits(2).bones(3).next_joint.y	digits(2).bones(3).next_joint.z	digits(2).bones(3).prev_joint.x	digits(2).bones(3).prev_joint.y	digits(2).bones(3).prev_joint.z	digits(2).bones(3).rotation.w	digits(2).bones(3).rotation.x	digits(2).bones(3).rotation.y
17	digits(2).bones(3).rotation.z	3	digits(3).finger_id	digits(3).is_extended	digits(3).bones(0).width	digits(3).bones(0).next_joint.x	digits(3).bones(0).next_joint.y	digits(3).bones(0).next_joint.z	digits(3).bones(0).prev_joint.x	digits(3).bones(0).prev_joint.y
18	digits(3).bones(0).prev_joint.z	digits(3).bones(0).rotation.w	digits(3).bones(0).rotation.x	digits(3).bones(0).rotation.y	digits(3).bones(0).rotation.z	digits(3).bones(1).width	digits(3).bones(1).next_joint.x	digits(3).bones(1).next_joint.y	digits(3).bones(1).next_joint.z	digits(3).bones(1).prev_joint.x
19	digits(3).bones(1).prev_joint.y	digits(3).bones(1).prev_joint.z	digits(3).bones(1).rotation.w	digits(3).bones(1).rotation.x	digits(3).bones(1).rotation.y	digits(3).bones(1).rotation.z	digits(3).bones(2).width	digits(3).bones(2).next_joint.x	digits(3).bones(2).next_joint.y	digits(3).bones(2).next_joint.z
20	digits(3).bones(2).prev_joint.x	digits(3).bones(2).prev_joint.y	digits(3).bones(2).prev_joint.z	digits(3).bones(2).rotation.w	digits(3).bones(2).rotation.x	digits(3).bones(2).rotation.y	digits(3).bones(2).rotation.z	digits(3).bones(3).width	digits(3).bones(3).next_joint.x	digits(3).bones(3).next_joint.y
21	digits(3).bones(3).next_joint.z	digits(3).bones(3).prev_joint.x	digits(3).bones(3).prev_joint.y	digits(3).bones(3).prev_joint.z	digits(3).bones(3).rotation.w	digits(3).bones(3).rotation.x	digits(3).bones(3).rotation.y	digits(3).bones(3).rotation.z	4	digits(4).finger_id
22	digits(4).is_extended	digits(4).bones(0).width	digits(4).bones(0).next_joint.x	digits(4).bones(0).next_joint.y	digits(4).bones(0).next_joint.z	digits(4).bones(0).prev_joint.x	digits(4).bones(0).prev_joint.y	digits(4).bones(0).prev_joint.z	digits(4).bones(0).rotation.w	digits(4).bones(0).rotation.x
23	digits(4).bones(0).rotation.y	digits(4).bones(0).rotation.z	digits(4).bones(1).width	digits(4).bones(1).next_joint.x	digits(4).bones(1).next_joint.y	digits(4).bones(1).next_joint.z	digits(4).bones(1).prev_joint.x	digits(4).bones(1).prev_joint.y	digits(4).bones(1).prev_joint.z	digits(4).bones(1).rotation.w
24	digits(4).bones(1).rotation.x	digits(4).bones(1).rotation.y	digits(4).bones(1).rotation.z	digits(4).bones(2).width	digits(4).bones(2).next_joint.x	digits(4).bones(2).next_joint.y	digits(4).bones(2).next_joint.z	digits(4).bones(2).prev_joint.x	digits(4).bones(2).prev_joint.y	digits(4).bones(2).prev_joint.z
25	digits(4).bones(2).rotation.w	digits(4).bones(2).rotation.x	digits(4).bones(2).rotation.y	digits(4).bones(2).rotation.z	digits(4).bones(3).width	digits(4).bones(3).next_joint.x	digits(4).bones(3).next_joint.y	digits(4).bones(3).next_joint.z	digits(4).bones(3).prev_joint.x	digits(4).bones(3).prev_joint.y
26	digits(4).bones(3).prev_joint.z	digits(4).bones(3).rotation.w	digits(4).bones(3).rotation.x	digits(4).bones(3).rotation.y	digits(4).bones(3).rotation.z	arm.width	arm.next_joint.x	arm.next_joint.y	arm.next_joint.z	arm.prev_joint.x
27	arm.prev_joint.y	arm.prev_joint.z	arm.rotation.w	arm.rotation.x	arm.rotation.y	arm.rotation.z				

References

1. United Nations. International Day of Sign Languages. Available online: https://www.un.org/en/observances/sign-languages-day (accessed on 19 December 2022).
2. Premaratne, P.; Nguyen, Q.; Premaratne, M. Human Computer Interaction Using Hand Gestures. In *Advanced Intelligent Computing Theories and Applications*; Huang, D.-S., McGinnity, M., Heutte, L., Zhang, X.-P., Eds.; Springer: Berlin, Heidelberg, 2010; pp. 381–386.
3. LaViola, J.J.J. A Survey of Hand Posture and Gesture Recognition Techniques and Technology 1999. Brown Univ. Provid. RI. 1999. Available online: https://www.semanticscholar.org/paper/A-Survey-of-Hand-Posture-and-Gesture-Recognition-LaViola/856d4bf0f1f5d4480ce3115d828f34d4b2782e1c (accessed on 12 July 2022).
4. CyberGlove Systems LLC. Available online: http://www.cyberglovesystems.com/ (accessed on 25 August 2022).
5. Hernandez-Rebollar, J.L.; Kyriakopoulos, N.; Lindeman, R.W. The AcceleGlove: A whole-hand input device for virtual reality. In Proceedings of the ACM SIGGRAPH 2002 Conference Abstracts and Applications; Association for Computing Machinery: New York, NY, USA, 2002; p. 259.
6. Barreto, A.; Scargle, S.; Adjouadi, M. Hands-off human-computer interfaces for individuals with severe motor disabilities. In Proceedings of the on Human-Computer Interaction: Communication, Cooperation, and Application Design, Hillsdale, NJ, USA, 22–26 August 1999; L. Erlbaum Associates Inc.: Munich, Alemania; Volume 2, pp. 970–974.
7. Coleman, K. Electromyography based human-computer-interface to induce movement in elderly persons with movement impairments. In Proceedings of the 2001 EC/NSF Workshop on Universal Accessibility of Ubiquitous Computing: Providing for the Elderly, Alcácer do Sal, Portugal, 22–25 May 2001; Association for Computing Machinery: New York, NY, USA, 2001; pp. 75–79.
8. Guerreiro, T.; Jorge, J. EMG as a daily wearable interface. In Proceedings of the First International Conference on Computer Graphics Theory and Applications, Setúbal, Portugal, 25–28 February 2006; pp. 216–223.
9. Ahsan, R.; Ibrahimy, M.I.; Khalifa, O.O. EMG Signal Classification for Human Computer Interaction: A Review. *Eur. J. Sci. Res.* **2009**, *33*, 480–501.
10. Booij, W.E.; Welle, K.O. Ultrasound detectors. US8792305B2, 29 July 2014. Available online: https://patents.google.com/patent/US8792305B2/en (accessed on 12 July 2022).
11. Saad, M.; Bleakley, C.J.; Nigram, V.; Kettle, P. Ultrasonic hand gesture recognition for mobile devices. *J. Multimodal. User Interfaces* **2018**, *12*, 31–39. [CrossRef]
12. Sang, Y.; Shi, L.; Liu, Y. Micro Hand Gesture Recognition System Using Ultrasonic Active Sensing. *IEEE Access* **2018**, *6*, 49339–49347. [CrossRef]
13. Asadzadeh, P.; Kulik, L.; Tanin, E. Gesture recognition using RFID technology. *Pers. Ubiquit. Comput.* **2012**, *16*, 225–234. [CrossRef]
14. Bouchard, K.; Bouzouane, A.; Bouchard, B. Gesture recognition in smart home using passive RFID technology. In Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments, Rhodes, Greece, 27–30 May 2014; Association for Computing Machinery: Nueva York, NY, USA, 2014; pp. 1–8.
15. Jayatilaka, A.; Ranasinghe, D.C. Real-time fluid intake gesture recognition based on batteryless UHF RFID technology. *Pervasive Mob. Comput.* **2017**, *34*, 146–156. [CrossRef]
16. Wen, Y.; Hu, C.; Yu, G.; Wang, C. A robust method of detecting hand gestures using depth sensors. In Proceedings of the 2012 IEEE International Workshop on Haptic Audio Visual Environments and Games (HAVE 2012), Munich, Germany, 8–9 October 2012; pp. 72–77.
17. API Overview—Leap Motion JavaScript SDK v3.2 Beta Documentation. Available online: https://developer-archive.leapmotion.com/documentation/javascript/devguide/Leap_Overview.html (accessed on 12 July 2022).
18. Funasaka, M.; Ishikawa, Y.; Takata, M.; Joe, K. Sign Language Recognition using Leap Motion; In Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA). Las Vegas, NV, USA, 25–28 July 2016; pp. 263–269.
19. Marin, G.; Dominio, F.; Zanuttigh, P. Hand gesture recognition with leap motion and kinect devices. In Proceedings of the 2014 IEEE International Conference on Image Processing (ICIP), Paris, France, 27–30 October 2014; pp. 1565–1569.
20. Simos, M.; Nikolaidis, N. Greek sign language alphabet recognition using the leap motion device. In Proceedings of the 9th Hellenic Conference on Artificial Intelligence, Thessaloniki, Greece, 18–20 May 2016; Association for Computing Machinery: Nueva York, NY, USA, 2016; pp. 1–4.
21. Mapari, R.B.; Kharat, G. American Static Signs Recognition Using Leap Motion Sensor. In Proceedings of the Second International Conference on Information and Communication Technology for Competitive Strategies, Udaipur, India, 4–5 March 2016; Association for Computing Machinery: New York, NY, USA, 2016; pp. 1–5.
22. Vaitkevičius, A.; Taroza, M.; Blažauskas, T.; Damaševičius, R.; Maskeliūnas, R.; Woźniak, M. Recognition of American Sign Language Gestures in a Virtual Reality Using Leap Motion. *Appl. Sci.* **2019**, *9*, 445. [CrossRef]
23. Mohandes, M.; Deriche, M.; Liu, J. Image-Based and Sensor-Based Approaches to Arabic Sign Language Recognition. *IEEE Trans. Hum. -Mach. Syst.* **2014**, *44*, 551–557. [CrossRef]
24. Mohandes, M.; Aliyu, S.; Deriche, M. Arabic Sign Language Recognition using the Leap Motion Controller. In Proceedings of the 2014 IEEE 23rd International Symposium on Industrial Electronics (ISIE), Istanbul, Turkey, 1–4 June 2014; pp. 960–965.

25. Hisham, B.; Hamouda, A. Arabic Sign Language Recognition using Microsoft Kinect and Leap Motion Controller. In Proceedings of the 11th International Conference on Informatics & Systems (INFOS 2018), Rochester, NY, USA, 27 October 2018.
26. Naglot, D.; Kulkarni, M. Real time sign language recognition using the leap motion controller. In Proceedings of the 2016 International Conference on Inventive Computation Technologies (ICICT), Coimbatore, India, 26–27 August 2016; Volume 3, pp. 1–5.
27. Chong, T.-W.; Lee, B.G. American Sign Language Recognition Using Leap Motion Controller with Machine Learning Approach. *Sensors* **2018**, *18*, 3554. [CrossRef] [PubMed]
28. Lee, C.K.M.; Ng, K.K.H.; Chen, C.-H.; Lau, H.C.W.; Chung, S.Y.; Tsoi, T. American sign language recognition and training method with recurrent neural network. *Expert Syst. Appl.* **2021**, *167*, 114403. [CrossRef]
29. Tao, W.; Lai, Z.-H.; Leu, M.C.; Yin, Z. American Sign Language Alphabet Recognition Using Leap Motion Controller. In Proceedings of the IIE Annual Conference, Orlando, FL, USA, 19–22 May 2018; ProQuest: Orlando, FL, USA, 2018; pp. 599–604. Available online: https://www.proquest.com/scholarly-journals/american-sign-language-alphabet-recognition-using/docview/2553578468/se-2 (accessed on 19 December 2022).
30. Anwar, A.; Basuki, A.; Sigit, R.; Rahagiyanto, A.; Zikky, M. Feature Extraction for Indonesian Sign Language (SIBI) Using Leap Motion Controller. In Proceedings of the 2017 21st International Computer Science and Engineering Conference (ICSEC), Bangkok, Thailand, 15–18 November 2017; pp. 1–5.
31. Alnahhas, A.; Alkhatib, B. Enhancing The Recognition Of Arabic Sign Language By Using Deep Learning And Leap Motion Controller. *Int. J. Sci. Technol. Res.* **2020**, *9*, 1865–1870.
32. Avola, D.; Bernardi, M.; Cinque, L.; Foresti, G.L.; Massaroni, C. Exploiting Recurrent Neural Networks and Leap Motion Controller for the Recognition of Sign Language and Semaphoric Hand Gestures. *IEEE Trans. Multimed.* **2019**, *21*, 234–245. [CrossRef]
33. Elons, A.S.; Ahmed, M.; Shedid, H.; Tolba, M.F. Arabic sign language recognition using leap motion sensor. In Proceedings of the 2014 9th International Conference on Computer Engineering & Systems (ICCES), Cairo, Egypt, 22–23 December 2014; pp. 368–373.
34. Jenkins, J.; Rashad, S. An Innovative Method for Automatic American Sign Language Interpretation using Machine Learning and Leap Motion Controller. In Proceedings of the 2021 IEEE 12th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON), New York, NY, USA, 1–4 December 2021; pp. 0633–0638.
35. Tuzcu, V.; Nas, S. Dynamic time warping as a novel tool in pattern recognition of ECG changes in heart rhythm disturbances. In Proceedings of the 2005 IEEE International Conference on Systems, Man and Cybernetics, Waikoloa, HI, USA, 10–12 October 2005; Volume 1, pp. 182–186.
36. Legrand, B.; Chang, C.S.; Ong, S.H.; Neo, S.-Y.; Palanisamy, N. Chromosome classification using dynamic time warping. *Pattern Recognit. Lett.* **2008**, *29*, 215–222. [CrossRef]
37. Kovacs-Vajna, Z.M. A fingerprint verification system based on triangular matching and dynamic time warping. *IEEE Trans. Pattern Anal. Mach. Intell.* **2000**, *22*, 1266–1276. [CrossRef]
38. Rath, T.M.; Manmatha, R. Word image matching using dynamic time warping. In Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Madison, WI, USA, 18–20 June 2003; Volume 2, p. II.
39. Okawa, M. Template Matching Using Time-Series Averaging and DTW With Dependent Warping for Online Signature Verification. *IEEE Access* **2019**, *7*, 81010–81019. [CrossRef]
40. Piyush Shanker, A.; Rajagopalan, A.N. Off-line signature verification using DTW. *Pattern Recognit. Lett.* **2007**, *28*, 1407–1414. [CrossRef]
41. Muda, L.; Begam, M.; Elamvazuthi, I. Voice Recognition Algorithms using Mel Frequency Cepstral Coefficient (MFCC) and Dynamic Time Warping (DTW) Techniques. *arXiv* **2010**, arXiv:1003.4083.
42. Amin, T.B.; Mahmood, I. Speech Recognition using Dynamic Time Warping. In Proceedings of the 2008 2nd International Conference on Advances in Space Technologies, Islamabad, Pakistan, 29–30 November 2008; pp. 74–79.
43. Adwan, S.; Arof, H. On improving Dynamic Time Warping for pattern matching. *Measurement* **2012**, *45*, 1609–1620. [CrossRef]
44. Arici, T.; Celebi, S.; Aydin, A.S.; Temiz, T.T. Robust gesture recognition using feature pre-processing and weighted dynamic time warping. *Multimed. Tools Appl.* **2014**, *72*, 3045–3062. [CrossRef]
45. Calin, A.D. Gesture Recognition on Kinect Time Series Data Using Dynamic Time Warping and Hidden Markov Models. In Proceedings of the 2016 18th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC), Timisoara, Rumanía, 24–27 September 2016; pp. 264–271.
46. Riofrío, S.; Pozo, D.; Rosero, J.; Vásquez, J. Gesture Recognition Using Dynamic Time Warping and Kinect: A Practical Approach. In Proceedings of the 2017 International Conference on Information Systems and Computer Science (INCISCOS), Quito, Ecuador, 23–25 November 2017; pp. 302–308.
47. Reyes, M.; Domínguez, G.; Escalera, S. Featureweighting in dynamic timewarping for gesture recognition in depth data. In Proceedings of the 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, Spain, 7 November 2011; pp. 1182–1188.
48. Raheja, J.L.; Minhas, M.; Prashanth, D.; Shah, T.; Chaudhary, A. Robust gesture recognition using Kinect: A comparison between DTW and HMM. *Optik* **2015**, *126*, 1098–1104. [CrossRef]

49. Ahmed, W.; Chanda, K.; Mitra, S. Vision based Hand Gesture Recognition using Dynamic Time Warping for Indian Sign Language. In Proceedings of the 2016 International Conference on Information Science (ICIS), Kochi, India, 12–13 August 2016; pp. 120–125.
50. Jambhale, S.S.; Khaparde, A. Gesture recognition using DTW & piecewise DTW. In Proceedings of the 2014 International Conference on Electronics and Communication Systems (ICECS), Coimbatore, India, 13–14 February 2014; pp. 1–5.
51. Kuzmanic, A.; Zanchi, V. Hand shape classification using DTW and LCSS as similarity measures for vision-based gesture recognition system. In Proceedings of the EUROCON 2007—The International Conference on "Computer as a Tool", Varsovia, Polonia, 9–12 September 2007; pp. 264–269.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.