Assistive Device for the Translation from Mexican Sign Language to Verbal Language

Héctor Varela-Santos[1], Anahí Morales-Jiménez[1], Diana-Margarita Córdova-Esparza[2],
Juan Terven[3], Flabio Dario Mirelez-Delgado[1], Adán Orenday-Delgado[1]

[1] Instituto Politécnico Nacional,
Unidad Profesional Interdisciplinaria de Ingeniería Zacatecas,
Mexico

[2] Universidad Autónoma de Querétaro,
Facultad de Informática, Querétaro,
Mexico

[3] AiFi Inc.,
USA

{chocovarela3, gaye971, flabio.mirelez}@gmail.com,
diana.cordova@uaq.mx, {jrterven, aorend}@hotmail.com

Abstract. In this work, we present the design and implementation of an assistive device for people with hearing disabilities, which allows words from Mexican Sign Language to be translated into verbal language. The device consists of a wearable embedded computer with a camera and a pair of gloves. The system captures the hand-gesture images, extracts features from the gloves, and runs an Artificial Neural Network as a classifier. Our system achieves an average precision of 88% and an average recall of 90% on 20 signals.

Keywords. Assistive devices, Mexican sign language, computer vision.

1 Introduction

The social inclusion of people with hearing disabilities is a current issue. On occasions, they continue to be excluded in the social environment due to their economic situation, physical characteristics, and communication limitations. In this work, we designed and implemented a prototype to help users with deaf-mute disabilities. The prototype consists of an embedded device that uses computer vision techniques and artificial neural networks to identify, characterize, classify, and recognize words from the Mexican Sign Language. The device's fundamental purpose is to allow essential communication between a hearing person who does not know Mexican Sign Language and a deaf-mute person by translating hand signs to speech in order to assist communication.

2 Related Work

There are different technologies to support the communication of people with hearing disabilities. Among them are mobile applications, such as Okisign [5], an application for educational purposes with a 581-word dictionary of Mexican Sign Language (MSL). This app includes a module to convert text to voice and vice versa, to communicate deaf and hearing. However, it is an application in development that cannot fully translate MSL. Also, it requires an internet connection to function and is ineffective for people with hearing disabilities who cannot read and write. Another application is Signamy [16], a mobile application that facilitates the learning of Sign

Language to improve communication between deaf and hearing people. This tool uses a virtual mediator that allows viewing categories and words of the Mexican Sign Language (MSL) and the American Sign Language (ASL), in addition to having a website to learn them. The work developed by [7], is a mobile application based on the Android operating system, focused mainly on facilitating the learning of the alphabet, verbs, and pronouns of the Mexican sign language.

However, the problem with the aforementioned mobile applications and websites is that even when they translate or teach MSL interactively, only a few listeners are interested in learning it. Consequently, they do not often visit websites and also don't download the apps.

Other technologies are based on electronic systems and Artificial Intelligence techniques, such as sign-translating gloves, which use sensors to identify and reproduce in text or audio the movements of the hand made by a person with hearing problems [17, 9, 14, 11].

In the proposal presented by [13], the segmentation of images that contains the Mexican sign language alphabet is performed and is used to train a neural network that allows each sign to be automatically recognized to control the tasks of a service robot. In other words, each sign is associated with a task that the robot must perform.

In 2010, Microsoft launched the Kinect sensor, a motion detection device that provides synchronized color and depth data and the user's body skeleton. Due to its low cost and accessibility, the Kinect sensor has been widely used in various works to identify hand gestures, for example, in [19], the recognition and verification of phrases of the American Sign Language were achieved for educational games focused on deaf children.

Regarding the use of neural networks for the recognition of static single-handed alphabets of MSL, in [13], a camera-based system was developed for data acquisition and the use of neural networks to control a service robot using MSL. Twenty-three alphabet signs were segmented using active contours and obtained an accuracy of 95.80%. In 2015, in the work of Galicia et al. [4] proposes a system that converts Mexican Sign Language into Spanish speech.

They used the Kinect sensor to capture 867 images that were trained using decision trees and neural networks with an accuracy of 76.19%. Another approach that uses the Kinect sensor for MSL single/double-handed word recognition is presented in [6]. They collected 700 samples of 20 Mexican words, from which skeleton data was extracted and forwarded to the training phase. The signs were then classified using dynamic time warping algorithm achieving an accuracy of 98.57% on real-time data.

According to the study developed in [18] regarding the different data acquisition techniques used by MSL systems, 33% use cameras, and 67% Kinect. In this work, we recognize a subset of MSL using a system based on a collar with a camera. The processing is performed on an embedded computer. We use Scale Invariant Feature Transform (SIFT) for feature extraction and a neural network for classification. Our system achieves an f1 score of 88%.

3 Method

This section describes the methodology used for the design and construction of the assistance device that allows the translation of twenty words from the MSL. The words are street, field, house, downtown, movie theater, city, address, building, school, party, hospital, hotel, church, garden, graveyard, park, restaurant, supermarket, theater, and university. These words were selected to achieve communication in concurrent places. The assistive device performs three main tasks (see Figure 1), which consist of:

— Capture the image. We acquire the image containing the signal made by the person with hearing problems using a camera, as indicated in the conceptual design in Figure 2a.

— Image processing. We used a Raspberry Pi to perform image processing and sign recognition.

— Sign-to-speech synthesis. Once the sign is recognized, the Raspberry Pi synthesizes the speech. It is worth mentioning that the receiver is a person without hearing problems and who does not know the MSL.

3.1 Conceptual Design

To select the most suitable design that fits the end-user, in this case, a person with hearing impairment who uses sign language to communicate, we proposed three designs where we consulted a sign language expert to determine the prototype's design characteristics based on motion sensitivity and camera stability for image acquisition, field of view, adaptation of the embedded system with the camera, ergonomics, cost, and accessibility. The proposed designs, are the following:

— A collar (see Figure 2a).

— Glasses (see Figure 3a).

— A Miner headlamp (see Figure 4a).

3.2 Design Features

This section describes the technical characteristics and usability of each design.

3.2.1 Motion Sensitivity and Camera Stability

To determine the stability of the three designs: a collar, glasses, and miner headlamp; we developed a test that measures the camera movement when gesturing a MSL signal. For this, we used a gyroscope MPU5060, and measured the angular velocity in radians/sec expressed in the graph in degrees/sec for the three rotation values: yaw, pitch, and roll.

In the case of the collar, we placed the sensor at the chest height (see Figure 2a), resulting in a suitable option because the camera is located at the height where the signs are performed. Also, regarding camera stability, when the person communicates with someone else, the body's involuntary movements may not significantly influence the camera motion and allow the system to acquire images correctly. It can be seen in Figure 2b, that the oscillations x, y, z are minimal. Yaw (rotation on the x-axis) is shown in red, pitch in green (rotation of the y-axis), and roll in yellow (where it rotates on its axis representing the z-axis).

Figure 3a shows the glasses design. The main disadvantage of this design is the camera position since many words in MSL are expressed at the abdomen's level. Figure 3b shows the camera stability expressed as the gyroscope oscillations in the three axes. It rarely stabilizes due to involuntary head movements when expressing a word in MSL.

In the miner headlamp design (see Figure 4a), the location of the camera is not feasible due to the high height making it difficult to capture the hands correctly. Figure 4b shows the camera stability expressed with the camera motion in the three axes. In this design, there is even more oscillation than the glasses design.

3.2.2 Camera's Field of View

We evaluate the camera's field of view in each of the proposed designs. The tests involved placing the camera in the area considered for each conceptual design and record hand signs. We concluded that the camera installed in the collar is the one with the best field of view.

3.2.3 Camera Connection with the Embedded System

An essential part of this research project is to achieve portability and comfort to the user. The camera must be of adequate size so that it is not invasive for the user and properly capture the sign images. When testing with potential users of the prototype, the glasses design achieved the lowest rating. Even though the camera is small, it covers part of the glasses and reduces the user's visibility. In addition, the wiring from the camera to the embedded board is not feasible for lenses and the miner headlamp.

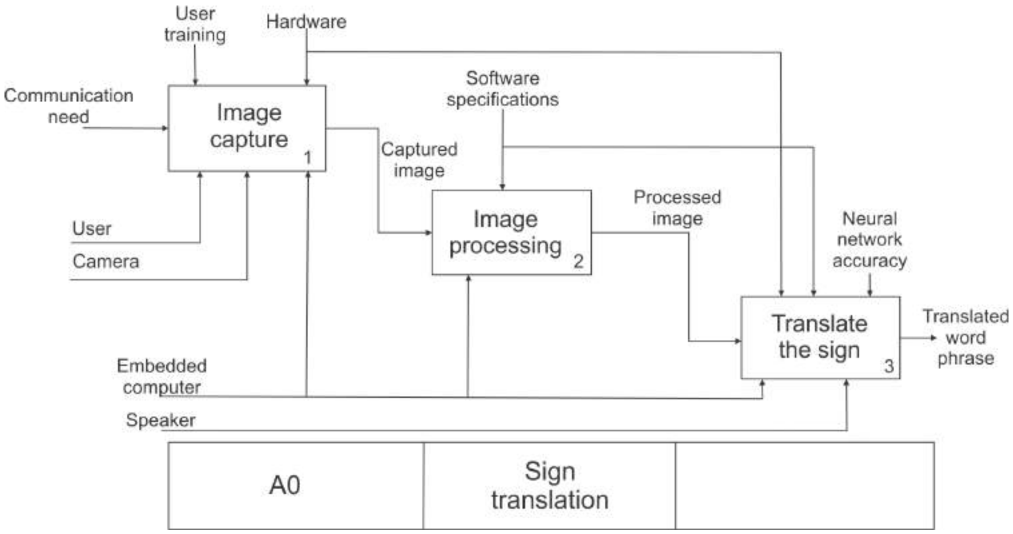
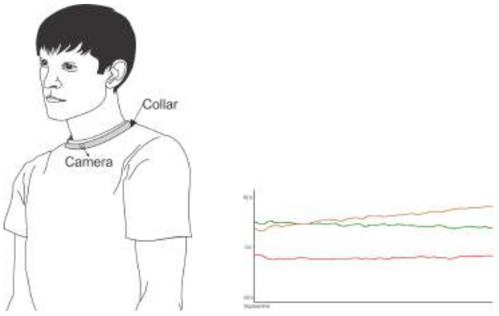


Fig. 1. IDEF-0 diagram level A0

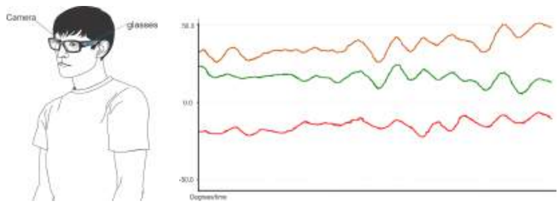


(a) Collar design **(b)** Camera stability

Fig. 2. Conceptual design. (a) Collar design with a camera. (b) The graph shows the three orientation values, the red plot represents the yaw (the rotation on the x-axis), the green plot depicts the pitch (the rotation on the y-axis), and the yellow plot is the roll (rotation on the z-axis)

3.2.4 Ergonomics, Portability and Cost

Ergonomics is one of the essential points to consider since the prototype is meant to be used by users who communicate using MSL. For this reason, we interviewed experts in sign language. Among the main conclusions of the interviews is that the device should not interfere in the area, where the hard of hearing person exercises his/hers speech.

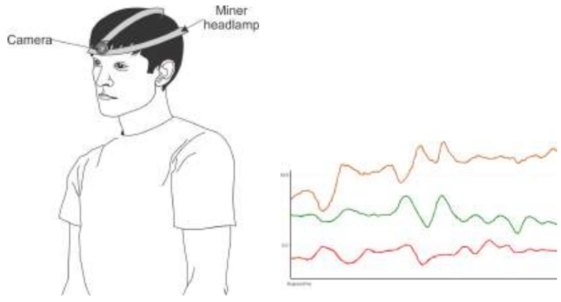


(a) Glasses design **(b)** Camera stability

Fig. 3. Conceptual design. (a) Glasses design with a camera. (b) The graph shows the three orientation values, the red plot represents the yaw (the rotation on the x-axis), the green plot depicts the pitch (the rotation on the y-axis), and the yellow plot is the roll (rotation on the z-axis)

Regarding portability, it is more feasible to wear a collar, than glasses which the user considers may interfere with her vision, or in the case of the miner headlamp, it is annoying to wear it after a certain time.

Concerning cost, we took into account the hardware mount, such as the collar, the glasses, the headlamp, and the electronic equipment such as the embedded card, the camera, battery, and LED indicators. We found the collar as the cheapest design.



(a) Miner headlamp design **(b)** Camera stability

Fig. 4. Conceptual design. (a) Miner headlamp design with a camera. (b) The graph shows the three orientation values, the red plot represents the yaw (the rotation on the x-axis), the green plot depicts the pitch (the rotation on the y-axis), and the yellow plot is the roll (rotation on the z-axis)

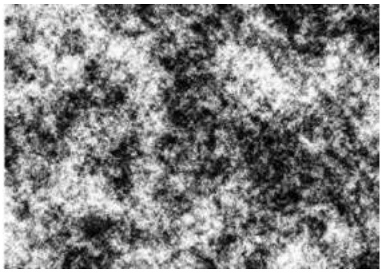


Fig. 5. Random pattern used to generate highly detectable features for pattern recognition

3.2.5 Accessibility

We performed a feasibility analysis for the acquisition of the required hardware to build the prototype.

Table 1 shows the results obtained regarding the design specifications.

In conclusion, through our analysis involving the technical and usability specifications of each design, the most suitable is the collar, which can be adjusted to the area where the hand-signs are produced without influencing the involuntary movements of the person who wears it. Also, placing the collar at the chest level allows the user to freely move their arms to produce the signs, besides being ergonomic and portable.

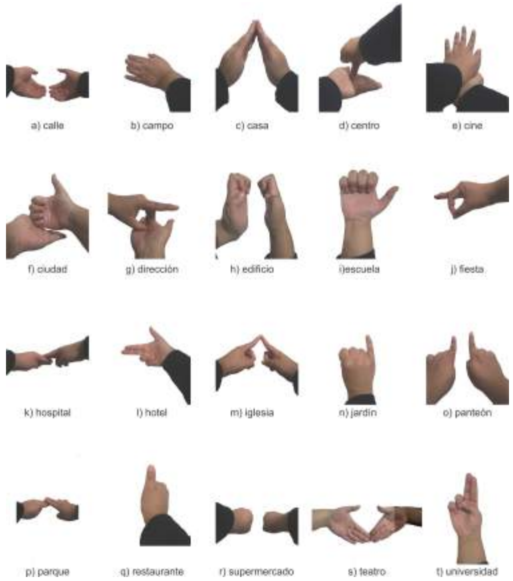


Fig. 6. Signs in Mexican Sign Language (MSL)

Table 1. Specifications of each design

Features	Collar	Glasses	Miner headlamp
Ergonomics	3	5	5
Cost	3	5	5
Accessibility	5	5	5
Motion sentivity	5	3	2
Camera adaptation	5	2	3
Camera's field of view	5	1	2
Total	26	21	22

*Where five is the best and one is the worst

3.3 Glove Design for Pattern Recognition

To recognize the hand gestures automatically, we designed a custom glove. We analyze which parts of the hand are visible from the collar camera perspective when gesturing each word using the MSL and place patterns that allow robust feature extraction for each sign. Figure 5 shows the pattern that we used to generate highly detectable features using SIFT/SURF [12, 2]. This pattern was designed by Li et al. [10] to calibrate multiple cameras. However, in this work, we use it to characterize each sign gestured in the MSL.

From the analysis made to the twenty signs (see Figure 6) that will be translated by the device, we obtained eighteen zones around the gloves, as shown in Figure 7.

Table 2. Input and output vectors of the neural network

Sign	Input	Output
CALLE	[d1,d2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]	[1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
CAMPO	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,d15,0,0,0,0]	[0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
CASA	[0,0,d3,0,0,0,0,0,0,0,d11,0,0,0,0,0,0,0,0]	[0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
CENTRO	[0,0,0,d4,d5,0,0,0,0,0,0,0,0,0,0,0,0,0,0]	[0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
CINE	[0,0,0,0,0,d6,d7,0,0,0,0,0,0,0,0,0,0,d18,0]	[0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
CIUDAD	[0,0,0,0,0,0,0,d8,0,0,0,0,0,d14,0,0,0,0,0]	[0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0]
DIRECCIÓN	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,d19]	[0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0]
EDIFICIO	[0,0,0,0,0,0,0,0,0,0,d11,0,0,0,0,0,0,0,0]	[0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0]
ESCUELA	[0,0,0,0,0,0,0,d8,0,0,0,0,d13,0,0,0,0,0,0]	[0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0]
FIESTA	[0,0,d3,0,0,0,0,0,0,0,0,0,0,0,0,d16,0,0,0]	[0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0]
HOSPITAL	[0,0,0,0,0,d6,d7,0,0,0,0,0,0,0,0,0,0,0,0]	[0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0]
HOTEL	[0,0,0,0,0,0,0,d8,d9,0,0,0,0,0,0,0,0,0,0]	[0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0]
IGLESIA	[0,0,0,0,0,0,0,0,0,d10,d11,0,0,0,0,0,0,0,0]	[0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0]
JARDÍN	[0,0,0,0,d5,0,0,0,0,0,0,0,0,0,0,0,0,0,0]	[0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0]
PANTEÓN	[0,0,0,0,0,d6,d7,0,0,0,0,d12,0,0,d15,0,0,0,0]	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0]
PARQUE	[0,0,0,0,0,d6,d7,0,0,0,0,d12,0,0,0,0,0,0,0]	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0]
RESTAURANTE	[0,0,d3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0]	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0]
SUPERMERCADO	[0,0,0,0,0,0,0,0,0,0,0,d12,0,0,0,0,0,0,0]	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0]
TEATRO	[0,0,0,0,0,0,0,d8,d9,0,0,0,0,0,0,0,d17,0,0]	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0]
UNIVERSIDAD	[0,0,0,0,0,0,0,0,d9,0,0,0,0,0,0,0,0,0,0]	[0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1]

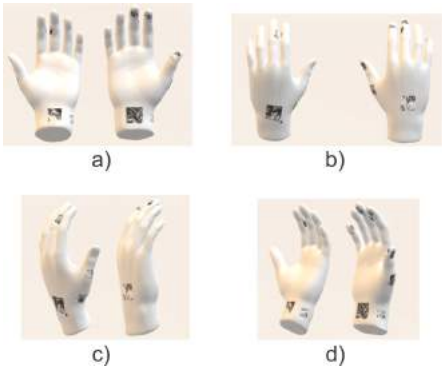


Fig. 7. Glove design according to the characteristics of the words in MSL that we want to recognize. (a) Design of inner part glove, (b) Design of outer part glove, (c) Design of outer side part glove, (d) Design of inner side part glove

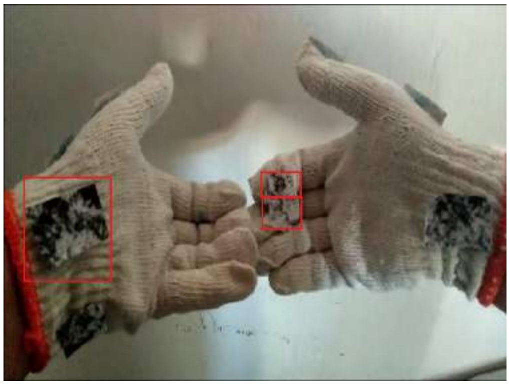


Fig. 8. Mexican sign language of the word “calle” and its adaptation in the gloves to characterize it. The red boxes show the chosen areas to place the patterns to characterize this sign

Figure 8 shows a Mexican sign language of the word “calle” and its adaptation in the glove to characterize it. In the figure, the red rectangles show the chosen zones that characterize this

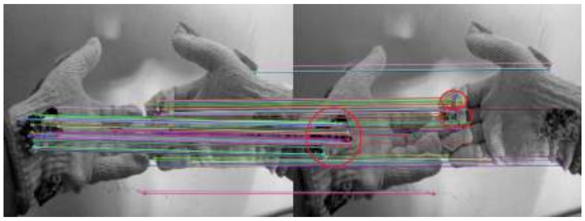


Fig. 9. Feature detection and matching between the fixed pattern for the sign “calle” (left) and the same sign performed afterwards

particular sign. The supplementary material sections shows the rest of the ten signs. We used the Scale Invariant Features Transform (SIFT) algorithm to find features in the gloves and match them with fixed sign patterns. Figure 8 shows that this sign has three defined zones; these zones are used to characterize the signal by finding enough SIFT feature matches inside these zones, as shown in Figure 9. We then compute the centroids of the points lying inside each zone. For example, for the “calle” sign, we compute three centroids, as shown in Figure 10.

We then compute normalized distances between the centroids and fixed hand locations. For example, Figure 11 shows the distances between the centroids obtained on two fingers zones and the wrist zone. The supplementary materials sections shows the rest of the twenty signs.

The Euclidean distance is computed with equation 1, where the centroid of one regions is represented as (x_i, y_i) and the centroid of the fixed region is represented as (x_j, y_j):

$$d = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}. \qquad (1)$$

For each sign, we compute a total of 19 distances generating a 19-dimensional sparse vector that is used as input for the neural network.

3.4 Neural Network Design

To classify the signs, we used a supervised fully-connected feed-forward network shown in Figure 12. We use Rectified Linear Units (ReLU) as activation functions on the hidden layers and a softmax activation at the output layer.

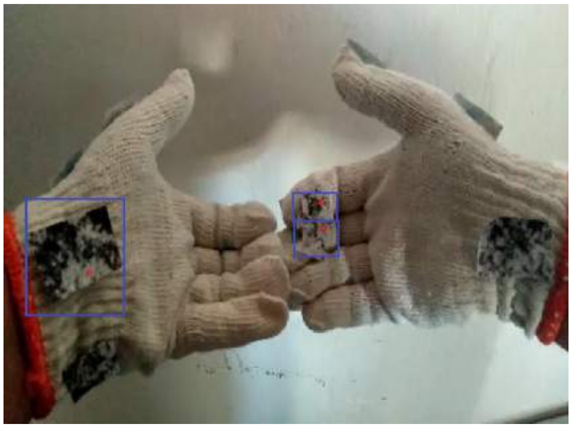


Fig. 10. Zones centroids. Each blue box represents a zone and the red marker indicates the centroid of the feature matches found inside each zone

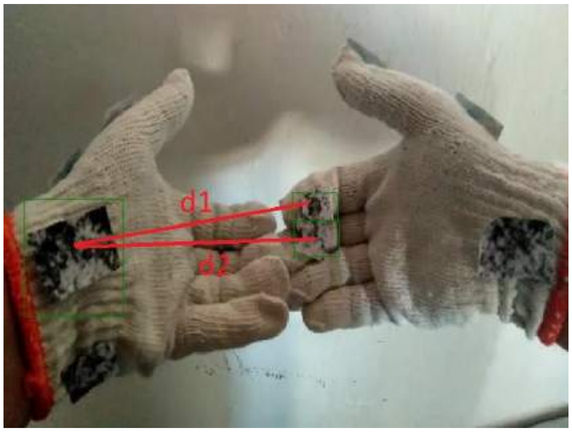


Fig. 11. Euclidean distances between the centroids of two finger zones and the left wrist zone

The input, represented by the first layer, contains 19 input values, each corresponding to the labeled distances. For example, the “calle” sign is represented as a vector with the form: $[d1, d2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$ where $d1$ and $d2$ are the distances found for that sign. Using the validation data, we empirically conclude that three hidden layers with 40 neurons give the optimal results for this task. Finally, the output is of size twenty, to be able to classify among the twenty different signs.

Table 2 shows the shape of the input and output vectors for each of the signs to classify.

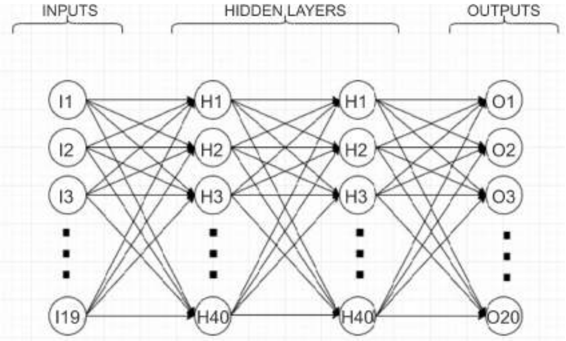


Fig. 12. Feed-forward Neural Network used to classify the signs

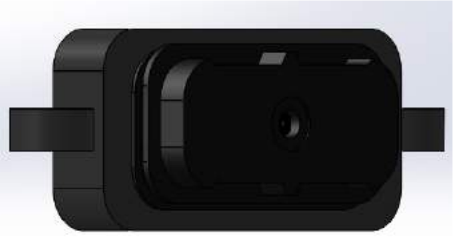


Fig. 13. Design of the board and camera enclosure. This enclosure is held by a pendant along the neck

4 Results

4.1 Prototype

Given the specifications of the project, we planned the usage of the device for the search of places, that is why the main characteristic that the design of this prototype must meet are: wearable, comfortable and flexible for the user.

4.1.1 CAD Design

Figure 13 shows the design of the board and camera enclosure. To power the embedded board, we use a portable battery, to which we also design a holder shown in Figure 14. Figure 15 shows the complete hardware device as it is intended to be worn by the users.

Table 3 shows the hardware specifications.

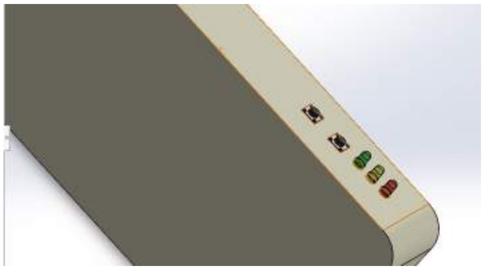


Fig. 14. Battery holder. The box contains the battery inside that supplies power to the embedded board, camera and the LEDs shown in this design. This battery holder is attached to the pants' belt

Table 3. Hardware Specifications

Hardware	Length	Width	Height
Enclosure	80 mm	38 mm	25 mm
Battery holder	150 mm	20 mm	68 mm
Battery	138 mm	30 mm	61 mm
Camera	6 mm	3.5 mm	6 mm
Enclosure holder	120 mm	17 mm	48 mm

4.1.2 Electronic Design

For the embedded board, we use a Raspberry Pi Zero W with the following specifications relevant to our system:

— 1GHz single-core CPU.

— 512 MB de RAM.

— Micro USB power.

— CSI camera connector.

According to the specifications, the Raspberry Pi Zero W consumes $150mA$ during typical use. We placed a push-button to trigger the camera and start the processing. We also put three LEDs, a red LED to indicate when the Raspberry is on, a yellow LED to indicate when the image is being processed, and a green LED shows when the sign-to-speech is played on the speaker. Each LED consumes $10mA$, the camera consumes $250mA$ at $5V$. Adding up all the currents results in $430mA$. We chose a $10000mAh$ battery with a duration given by $(10000mAh/430mA) = 23.25hr$

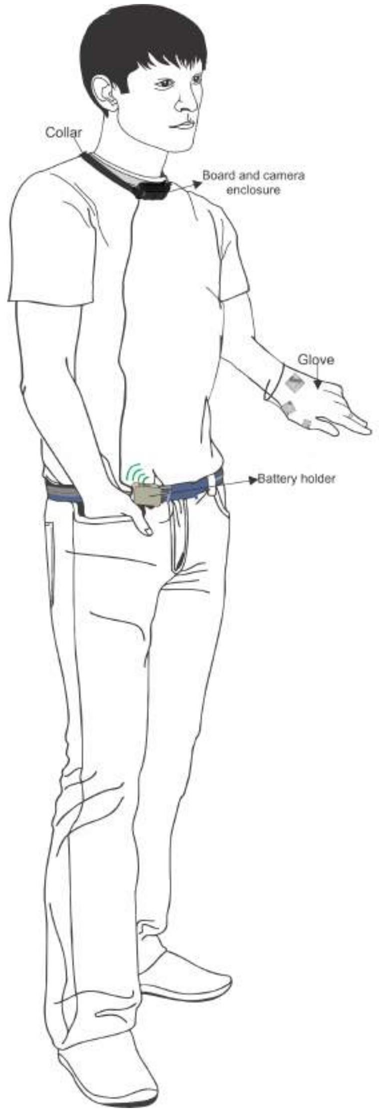


Fig. 15. Complete system. The image shows a user wearing the full system. The board enclosure and camera hanging on the neck, the battery holder attached to the belt and the gloves

4.1.3 Camera and Speech Synthesis

To select a suitable camera, we considered the field-of-view (FOV), resolution, and hardware compatibility. After testing multiple cameras, we decided to use the PiCamera V2 model with a diagonal FOV of $79°$, and a working resolution of 640×480 pixels.

For the speech part, we recorded the audio of twenty words corresponding to each of the signs and reproduce the audio files using the Pygame library [15] when a known signal is detected.

4.2 Classification Results

We collected around 3600 images of hand signals for training and 400 images for testing the neural network. We train the model using Keras [3] and Tensorflow libraries [1].

The testing data consists of around 20 samples for each of the twenty signs performed by multiple people.

To evaluate the performance of our method, we calculate the Precision, Recall, and F1-score. Table 5 shows these metrics for twenty words of the Mexican Sign Language. These metrics are based on the correctly/incorrectly classified signs which are defined with the true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) described below [8]:

— True Positive (TP) refers to the number of predictions where the classifier correctly predicts the positive class as positive.

— True Negative (TN) refers to the number of predictions where the classifier correctly predicts the negative class as negative.

— False Positive (FP) refers to the number of predictions where the classifier incorrectly predicts the negative class as positive.

— False Negative (FN) refers to the number of predictions where the classifier incorrectly predicts the positive class as negative.

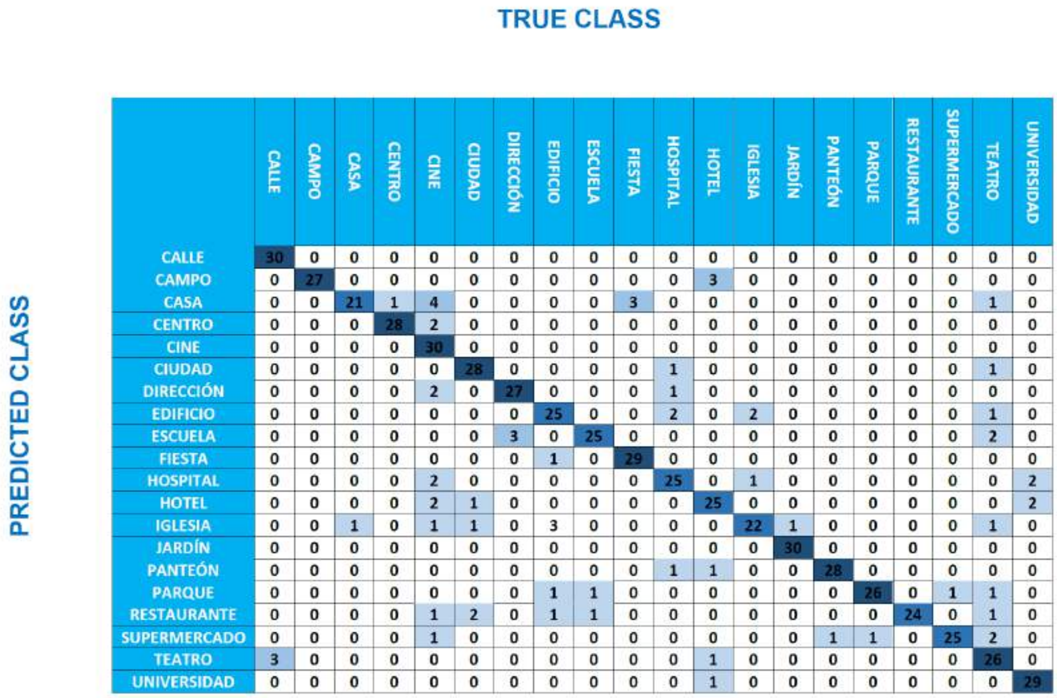
The Precision represents the proportion of positive identifications that were actually correct. For example, for the sign “hospital” (see Table 5), a Precision of 0.9 means that when it predicts a sign as “hospital”, it is correct 90% of the time. The Precision is calculated with equation 2:

$$Precision = \frac{TP}{TP + FP}. \tag{2}$$

The Recall represents the proportion of actual positives correctly identified.

Table 4. F_1 F1 score obtained with each experiment

	2 hidden layers				3 hidden layers				4 hidden layers			
	20	30	40	50	20	30	40	50	20	30	40	50
Sign	neurons	neurons	neurons	neurons	neurons	neurons	neurons	neurons	neurons	neurons	neurons	neurons
Calle	0.896	0.820	0.923	0.938	0.935	0.936	0.952	0.885	0.857	0.897	0.744	0.875
Campo	0.966	0.740	0.966	1.000	0.909	0.952	0.947	0.929	0.968	0.889	0.889	0.806
Casa	0.877	0.910	0.897	0.760	0.857	0.806	0.808	0.735	0.774	0.769	0.500	0.654
Centro	0.889	0.830	0.889	0.929	0.909	0.919	0.949	0.893	0.909	0.873	0.967	0.889
Cine	0.817	0.760	0.707	0.706	0.769	0.736	0.800	0.718	0.641	0.800	0.875	0.812
Ciudad	0.812	0.850	0.767	0.789	0.691	0.737	0.862	0.778	0.875	0.800	0.848	0.824
Dirección	0.909	0.800	0.909	0.824	0.846	0.835	0.947	0.929	0.949	0.966	0.889	0.929
Edificio	0.786	1.000	0.627	0.794	0.733	0.762	0.820	0.667	0.764	0.644	0.714	0.724
Escuela	0.912	0.860	0.808	0.873	0.847	0.860	0.893	0.847	0.873	0.909	0.873	0.909
Fiesta	0.947	0.900	0.947	0.947	0.966	0.956	0.935	0.870	0.923	0.947	0.947	0.875
Hospital	0.769	0.930	0.714	0.754	0.722	0.738	0.833	0.800	0.787	0.800	0.612	0.794
Hotel	0.792	0.810	0.745	0.808	0.808	0.808	0.820	0.842	0.792	0.794	0.833	0.857
Iglesia	0.714	0.800	0.824	0.868	0.767	0.814	0.800	0.808	0.824	0.724	0.745	0.846
Jardín	1.000	0.900	1.000	1.000	0.983	0.991	0.984	0.938	0.984	1.000	0.929	1.000
Panteón	0.929	0.700	0.921	0.967	0.951	0.959	0.949	0.951	0.947	0.915	0.966	0.935
Parque	0.871	0.900	0.862	0.885	0.897	0.891	0.912	0.881	0.877	0.912	0.918	0.903
Restaurante	0.754	0.740	0.867	0.833	0.800	0.816	0.889	0.909	0.767	0.704	0.741	0.792
Supermercado	0.873	0.890	0.893	0.868	0.889	0.878	0.893	0.893	0.862	0.862	0.868	0.881
Teatro	0.781	0.840	0.831	0.794	0.806	0.800	0.765	0.814	0.778	0.857	0.686	0.814
Universidad	0.918	0.780	0.918	0.951	0.915	0.933	0.921	0.935	0.949	0.900	0.929	0.848
Average	0.861	0.838	0.851	0.864	0.850	0.856	**0.884**	0.851	0.855	0.848	0.824	0.848



TRUE CLASS

PREDICTED CLASS

	CALLE	CAMPO	CASA	CENTRO	CINE	CIUDAD	DIRECCIÓN	EDIFICIO	ESCUELA	FIESTA	HOSPITAL	HOTEL	IGLESIA	JARDÍN	PANTEÓN	PARQUE	RESTAURANTE	SUPERMERCADO	TEATRO	UNIVERSIDAD
CALLE	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CAMPO	0	27	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0
CASA	0	0	21	1	4	0	0	0	0	3	0	0	0	0	0	0	0	0	1	0
CENTRO	0	0	0	28	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CINE	0	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
CIUDAD	0	0	0	0	0	28	0	0	0	0	1	0	0	0	0	0	0	0	1	0
DIRECCIÓN	0	0	0	0	2	0	27	0	0	0	1	0	0	0	0	0	0	0	0	0
EDIFICIO	0	0	0	0	0	0	0	25	0	0	2	0	2	0	0	0	0	0	1	0
ESCUELA	0	0	0	0	0	0	3	0	25	0	0	0	0	0	0	0	0	0	2	0
FIESTA	0	0	0	0	0	0	0	1	0	29	0	0	0	0	0	0	0	0	0	0
HOSPITAL	0	0	0	0	2	0	0	0	0	0	25	0	1	0	0	0	0	0	0	2
HOTEL	0	0	0	0	2	1	0	0	0	0	0	25	0	0	0	0	0	0	0	2
IGLESIA	0	0	1	0	1	1	0	3	0	0	0	0	22	1	0	0	0	0	1	0
JARDÍN	0	0	0	0	0	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0
PANTEÓN	0	0	0	0	0	0	0	0	0	0	1	1	0	0	28	0	0	0	0	0
PARQUE	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	26	0	1	1	0
RESTAURANTE	0	0	0	0	1	2	0	1	1	0	0	0	0	0	0	0	24	0	1	0
SUPERMERCADO	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1	1	0	25	2	0
TEATRO	3	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	26	0
UNIVERSIDAD	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	29

Fig. 16. Confusion matrix

Table 5. Classification results on the testing set

Sign	Precision	Recall	F_1
Calle	1.000	0.909	0.952
Campo	0.900	1.000	0.947
Casa	0.700	0.955	0.808
Centro	0.933	0.966	0.949
Cine	1.000	0.667	0.800
Ciudad	0.933	0.800	0.862
Dirección	0.900	1.000	0.947
Edificio	0.833	0.806	0.820
Escuela	0.833	0.962	0.893
Fiesta	0.967	0.906	0.935
Hospital	0.833	0.833	0.833
Hotel	0.833	0.806	0.820
Iglesia	0.733	0.880	0.800
Jardín	1.000	0.968	0.984
Panteón	0.933	0.966	0.949
Parque	0.867	0.963	0.912
Restaurante	0.800	1.000	0.889
Supermercado	0.833	0.962	0.893
Teatro	0.867	0.684	0.765
Universidad	0.967	0.879	0.921
Average	0.88	0.90	0.89

For the sign “hospital”, a Recall of 0.8 means that it correctly identifies 80% of all “hospital” signs. The recall is calculated with equation 3:

$$Recall = \frac{TP}{TP + FN}. \quad (3)$$

The F1 score is the harmonic mean of the Precision and Recall and is calculated with equation 4:

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \quad (4)$$

We performed the following series of experiments varying the number of hidden layers and number of neurons of the network:

— 2 hidden layers with 20 neurons.

— 2 hidden layers with 30 neurons.

— 2 hidden layers with 40 neurons.

— 2 hidden layers with 50 neurons.

— 3 hidden layers with 20 neurons.

— 3 hidden layers with 30 neurons.

— 3 hidden layers with 40 neurons.

— 3 hidden layers with 50 neurons.

— 4 hidden layers with 20 neurons.

— 4 hidden layers with 30 neurons.

— 4 hidden layers with 40 neurons.

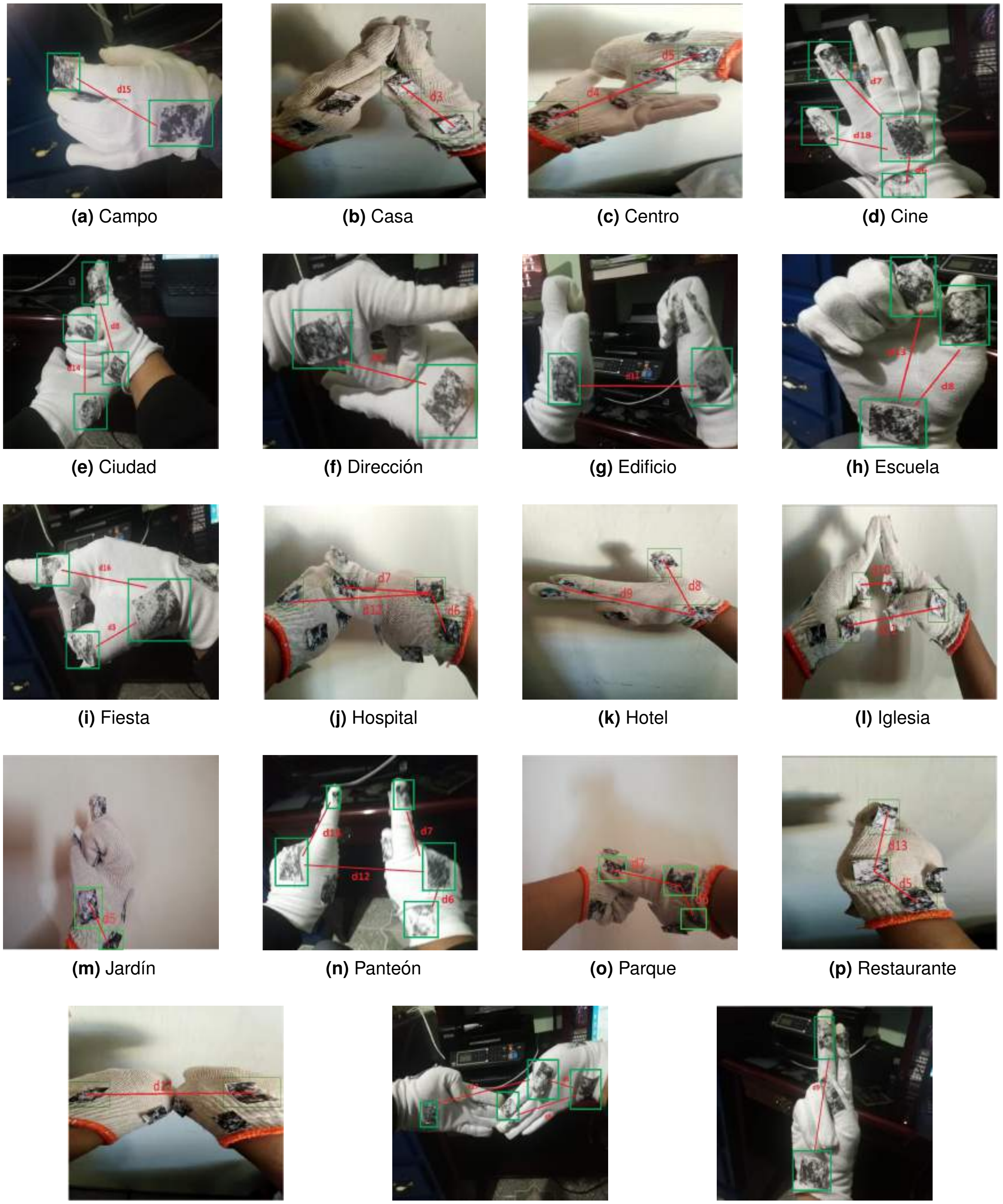
— 4 hidden layers with 50 neurons.

For each experiment, we obtained the results shown in Table 4. The best model was the one with three hidden layers and 40 neurons. This model contains 4400 parameters and can run in real time in the Raspberry Pi.

Figure 16 shows a color-coded multiclass confusion matrix. On the x-axis, we have the true labels, and on the y-axis, we have the predicted labels of our test set. A perfect classifier shows a confusion matrix where we have values only on the diagonal i.e., where we classify all the test samples for all the ten classes correctly. The values in the cells represent counts. For instance, the upper left cell has a value 30, and the rest of the row have 0s except on the “teatro” class with a value of three. This means that we can correctly classify 30 out of 33 test samples for the category “calle”, and the system mispredicted three instances of “calle” as “teatro”.

5 Supplementary Material: 19 signs

In the supplementary materials, we present the rest of the nineteen signs performed with the gloves and their distance vectors used for the classification (Figure 17).



(a) Campo **(b)** Casa **(c)** Centro **(d)** Cine

(e) Ciudad **(f)** Dirección **(g)** Edificio **(h)** Escuela

(i) Fiesta **(j)** Hospital **(k)** Hotel **(l)** Iglesia

(m) Jardín **(n)** Panteón **(o)** Parque **(p)** Restaurante

(q) Supermercado **(r)** Teatro **(s)** Universidad

Fig. 17. Distance Vectors. The images show the distance vectors for the different sign gestures. These vectors are used as inputs to the Artificial Neural Network for classification

6 Conclusion and Future Work

In this paper, we describe the development of a hand-sign-to-speech translation device that can be used for deaf and hard of hearing people to convey places with a hearing person. Our system using Computer Vision techniques and Artificial Neural Networks achieved an average precision of 88% and an average recall of 90% over twenty static signs from the MSL. We describe three different designs and select the most appropriate from the user and technology standpoint. The system works in real-time using a Raspberry Pi Zero board bundled with a PiCamera V2.

As future work, we plan to incorporate more static signs from the vast MSL to cover broader user-case applications as well as dynamic signs.

Acknowledgments

We would like to thank the Instituto Politécnico Nacional through the Unidad Profesional Interdisciplinaria de Ingeniería Campus Zacatecas (UPIIZ), for the support provided during the development of this work.

References

1. **Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., others (2016).** Tensorflow: A system for large-scale machine learning. 12th USENIX symposium on operating systems design and implementation (OSDI 16), pp. 265–283.
2. **Bay, H., Tuytelaars, T., Van Gool, L. (2006).** Surf: Speeded up robust features. European conference on computer vision, Springer, pp. 404–417.
3. **Chollet, F. (2020).** Keras: the python deep learning API. `https://keras.io/`.
4. **Galicia, R., Carranza, O., Jiménez, E., Rivera, G. (2015).** Mexican sign language recognition using movement sensor. 2015 IEEE 24th International Symposium on Industrial Electronics (ISIE), IEEE, pp. 573–578.
5. **García Bautista, G., Martínez Higareda, D. (2017).** Okisign. último acceso 25 de Agosto de 2018.
6. **García-Bautista, G., Trujillo-Romero, F., Caballero-Morales, S. O. (2017).** Mexican sign language recognition using Kinect and data time warping algorithm. 2017 International Conference on Electronics, Communications and Computers (CONIELECOMP), IEEE, pp. 1–5.
7. **Gordillo-Ramírez, A., Alonso-Cuevas, O., Ortega-Pacheco, D., Vélez-Saldaña, U. (2019).** Mobile application for the support in the learning of the alphabet, verbs and pronouns of the Mexican sign language based on augmented reality. International Congress of Telematics and Computing, Springer, pp. 183–191.
8. **Kuhn, M., Johnson, K., others (2013).** Applied predictive modeling, volume 26. Springer.
9. **Lei, L., Dashun, Q. (2015).** Design of data-glove and Chinese sign language recognition system based on ARM9. 2015 12th IEEE International Conference on Electronic Measurement & Instruments (ICEMI), volume 3, IEEE, pp. 1130–1134.
10. **Li, B., Heng, L., Koser, K., Pollefeys, M. (2013).** A multiple-camera system calibration toolbox using a feature descriptor-based calibration pattern. 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, pp. 1301–1307.
11. **Liang, R.-H., Ouhyoung, M. (1998).** A real-time continuous gesture recognition system for sign language. Proceedings third IEEE international conference on automatic face and gesture recognition, IEEE, pp. 558–567.
12. **Lowe, D. G. (1999).** Object recognition from local scale-invariant features. Proceedings of the seventh IEEE international conference on computer vision, volume 2, Ieee, pp. 1150–1157.
13. **Luis-Pérez, F. E., Trujillo-Romero, F., Martínez-Velazco, W. (2011).** Control of a service robot using the Mexican sign language. Mexican International Conference on Artificial Intelligence, Springer, pp. 419–430.
14. **Oz, C., Leu, M. C. (2005).** Linguistic properties based on American sign language recognition with artificial neural networks using a sensory glove and motion tracker. International Work-Conference on Artificial Neural Networks, Springer, pp. 1197–1205.
15. **Pygame, C. (2020).** Pygame library. `https://www.pygame.org`.
16. **Signamy (2016).** TecnologÍas asistivas para personas sordas. último acceso 18 de Marzo de 2020.

17. **Veerapalli, L., others (2015).** Sign language recognition through fusion of 5DT data glove and camera based information. 2015 IEEE International Advance Computing Conference (IACC), IEEE, pp. 639–643.

18. **Wadhawan, A., Kumar, P. (2019).** Sign language recognition systems: A decade systematic literature review. Archives of Computational Methods in Engineering, pp. 1–29.

19. **Zafrulla, Z., Brashear, H., Starner, T., Hamilton, H., Presti, P. (2011).** American sign language recognition with the Kinect. Proceedings of the 13th international conference on multimodal interfaces, pp. 279–286.

Article received on 31/07/2020; accepted on 02/04/2021.
Corresponding author is Diana-Margarita Córdova-Esparza.