Online-Journals.org

iJOE International Journal of Online and Biomedical Engineering

iJOE | eISSN: 2626-8493 | Vol. 22 No. 1 (2026) | OPEN ACCESS

https://doi.org/10.3991/ijoe.v22i01.58105

PAPER

Mobile Application for the Inclusion of People with Hearing Disabilities in Peru Using LSTM and GPT-4

Alyne Regalado-Morales, Axel Fiestas, Lenis Wong (✉)

Universidad Peruana de Ciencias Aplicadas, Lima, Peru

pcsilewo@upc.edu.pe

ABSTRACT

Communication between hearing individuals and deaf and hard-of-hearing individuals is often limited due to the general lack of sign language proficiency among the hearing population, leading to social exclusion and negatively affecting quality of life. This study proposes a mobile application that functions as an assistive technology for sign language recognition. Additionally, the application generates contextual responses using ChatGPT-4 to facilitate communication between both groups. The system was developed in five phases: 1) choice of recurrent neural network technique, 2) selection of the machine learning model, 3) implementation of the LSTM model, 4) implementation of the LLM model, and 5) construction of the mobile application. Validation involved a control experiment (A) and a test experiment (B). In A, the average response time (RT) was 95.93s, without achieving communicative clarity (CC) due to confusion during the interaction. In B, the RT was 102.28s, achieving CC evidenced by a relaxed body posture. Finally, the results of the survey after experiment B revealed acceptance of the system by the listener participants. These findings confirm the system's feasibility for facilitating communicative inclusion in real-world scenarios.

KEYWORDS

sign language recognition, mobile application, assistive technology, LSTM, ChatGPT-4, deaf and hard-of-hearing

1 INTRODUCTION

Communication is a fundamental human function essential for well-being and social participation [1]. However, when hearing is impaired, deaf and hard-of-hearing (DHH) individuals face barriers that significantly affect language development, education, employment, mental health, and interpersonal relationships. According to the *World Report on Hearing*, more than 1.5 billion people experience some degree of hearing loss, and 430 million suffer from moderate to severe impairment [2]. The number is projected to rise to 700 million by 2050 if preventive and assistive actions are not expanded globally. In the Peruvian context, the 2017 national census reported

Regalado-Morales, A., Fiestas, A., Wong, L. (2026). Mobile Application for the Inclusion of People with Hearing Disabilities in Peru Using LSTM and GPT-4. *International Journal of Online and Biomedical Engineering (iJOE)*, 22(1), pp. 114–132. https://doi.org/10.3991/ijoe.v22i01.58105

Article submitted 2025-08-21. Revision uploaded 2025-10-20. Final acceptance 2025-10-22.

that 243,486 people have hearing limitations, and 9,486 of them use Peruvian sign language (PSL) as their primary language [3]. Despite this, few technological solutions exist to support their communication needs in everyday interactions, highlighting the urgency of inclusive and accessible tools that bridge the communication gap between hearing and DHH individuals. In response to this need, various studies have developed technological solutions to support communication with DHH individuals. Examples include real-time translation systems for American sign language (ASL) using depth cameras [4], digital avatars for Arabic sign language (ArSL) [5], neural network models for the bidirectional translation of Mexican sign language (MSL) [6] and augmented reality applications for learning PSL [7] and Indonesian sign language [8]. Most existing systems focus on literal translation and ignore fluency, context, and conversational flow needed for natural communication.

This study introduces a mobile application that uses neural networks and large language models (LLMs) to facilitate communication between hearing and DHH people. The system translates PSL gestures into text and speech through a long short-term memory (LSTM) architecture and integrates the GPT-4 model to generate contextually appropriate and culturally adapted responses. A novel analytic hierarchy process (AHP) approach was also applied to guide the selection of the optimal neural network and LLM. Furthermore, unlike previous works that rely on synthetic datasets, our proposal was validated with real Peruvian participants, demonstrating its practical potential for social inclusion.

2 RELATED WORK

Given the lack of adequate resources to represent the specific sign systems of each country, multiple studies have opted to construct dedicated datasets for their respective sign languages. As a result, custom datasets have been developed for Indian, American, Saudi, Panamanian, Mexican, and Turkish sign languages [8–15]. However, these datasets are often small and heterogeneous, which restricts model generalization and may lead to overfitting when applied to new contexts.

The technological solutions identified in the literature are mainly grouped into specialized systems, web applications and mobile platforms. For example, [9] proposes a system to improve real-time sign detection, while [10] focuses on facilitating doctor–patient communication for DHH individuals by recognizing key symptoms such as “pain” or “throat.” The system in [11] addresses the recognition of isolated words in video, and [6] offers a graphical interface that provides bidirectional translation between MSL and Spanish. In the web domain, [12] focuses on the recognition of static ASL alphabet gestures, while [13] translates both static and dynamic gestures into text and voice in real time. In the mobile group, recent works have developed applications that use artificial intelligence to support inclusive communication and promote the learning of sign language [14] [15].

Various studies apply artificial intelligence techniques such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), LSTM models, and transformer architectures. Some works focus on the recognition of individual gestures using CNNs to capture spatial features. In [16], “Keras” was used along with “ImageDataGenerator” to classify static images based on shape, texture, or contour, achieving high efficiency. In [17], ArSL gestures were recognized using fine-tuned VGG16 and ResNet152. CNN-based models perform well in spatial analysis but lack temporal awareness, so they work mainly for static gestures.

Regarding dynamic gestures, [11] combines MobileNetV2 for spatial feature extraction and LSTM to capture the temporal sequence, which improves accuracy.

A similar approach is applied in [10], while [18] uses "MediaPipe Holistic" to extract key bodypoints, which are later processed by an LSTM model. These hybrid CNN–LSTM configurations enable the modeling of motion sequences, yet they tend to be computationally demanding and sensitive to the quantity and variability of training data. In [19], LSTM is used for its ability to learn long-term patterns, and [5] employs a transformer-based architecture to translate from ArSL to text, achieving 94.71% accuracy in training and 87.04% in testing. Transformer models improve contextual understanding while requiring large datasets and computing power, which limits their use in low-resource sign languages. Finally, [20] uses a 3D-CNN network that enables multi-view analysis of video sequences, processing spatial and temporal components simultaneously, demonstrating an alternative to recurrent structures but still constrained by dataset size and linguistic diversity.

3 MATERIALS AND METHODS

3.1 Selection of the RNN technique

A benchmarking process was conducted for the sign language translation task. The selected models were LSTM, gated recurrent unit (GRU), and simple RNN. These models were evaluated based on four key aspects: translation accuracy (A1), long-term retention capability (A2), training time (A3), and hardware requirements (A4). For the comparison of techniques, the AHP method was applied. This approach compares multiple criteria in pairs, ensuring transparent prioritization aligned with the study's objectives [19]. Using pairwise comparisons, AHP assigned the highest weight to translation accuracy (49.09%), followed by retention (29.13%), hardware (15.07%), and training time (6.70%). Table 1 presents the summary of the four evaluated aspects, showing the scores obtained by each technique and the corresponding weight (w) of each criterion.

Table 1. Results of the pairwise comparison matrix of technique aspects

ID	LSTM	GRU	RNN	w
A1	0.57	0.21	0.23	0.4909
A2	0.63	0.28	0.09	0.2913
A3	0.47	0.41	0.10	0.670
A4	0.63	0.25	0.10	0.1507
p	**58.68%**	**24.74%**	**16.17%**	**100%**

In addition, prioritization (p) is included, which was key to determining the most suitable technique for this study. The results indicate that the LSTM technique achieved the highest prioritization score (58.68%), making it the most appropriate option for the sign language translation task in this project.

3.2 Selection of the LLM

Five popular LLMs were evaluated based on their natural language processing capabilities and versatility in various contexts [21]. GPT-4 (M1) is a multimodal model developed by OpenAI, known for its advanced reasoning and effective handling of

complex interactions. GPT-4o Mini (M2) is a compact and efficient version of GPT-4 that retains strong language performance. GPT-3.5 Turbo (M3) is optimized for conversational tasks and features a wide context window [22]. Claude 3.5 Sonnet (M4), developed by Anthropic, is characterized by its contextual understanding and adaptability [23]. Gemini 1.5 Pro (M5), developed by Google, specializes in seamless integration with the Google ecosystem, making it ideal for mobile applications [24].

A benchmarking process was conducted based on five key criteria: contextual capability (B1), response time (B2), cultural adaptability (B3), implementation feasibility within a mobile application (B4), and cost per million tokens (B5). Aspect B1 evaluated the model's ability to interpret context and generate coherent and relevant responses. B2 measured the processing speed of each model, calculated in tokens per second. B3 assessed each model's responsiveness to the nuances of Peruvian Spanish and cultural context, achieved through targeted prompt engineering. B4 analyzed how easily the model can be implemented within a mobile application environment. Finally, B5 evaluated the cost associated with processing input and output tokens in each model.

The aspects were weighted with the AHP method, as in the RNN selection process. This approach was chosen because it enables a balanced evaluation of diverse criteria, combining performance-related factors such as response time and contextual capability with practical considerations such as implementation feasibility, cultural adaptability, and cost efficiency. Using pairwise comparisons, the weights assigned to each aspect were: contextual capability (B1) received the highest weight (51.17%), followed by response time (B2) with 26.76%, cultural adaptability (B3) with 12.12%, implementation (B4) with 5.27%, and cost per million tokens (B5) with 4.67%. The contextual capability criterion (B1) was divided into two sub-criteria: massive multitask language understanding (MMLU), which measures the cognitive and multitasking abilities of the model [25], and context window (CW), which evaluates the number of tokens the model can process in a single interaction [26]. MMLU received a higher weight (75%) due to its greater relevance in text comprehension tasks. Based on each model's performance across aspects B1–B5 and applying the AHP method weights, a global prioritization was calculated. Table 2 shows the scores obtained by each model, along with the weight values (w) and the final prioritization (p). A higher p value indicates better overall performance. Model M2 (GPT-4o Mini) achieved the highest prioritization (29.81%) and was therefore selected for implementation in the application. For response generation, the persona technique was used in prompt engineering. Through this technique, personalized profiles were established to optimize interactions between the user and the selected model, ensuring that the generated responses were contextual and coherent.

Table 2. Results of the pairwise comparison matrix of aspects for each LLM

ID	M1	M2	M3	M4	M5	*w*
B1	0.20	0.11	0.03	0.46	0.21	0.51
B2	0.03	0.55	0.22	0.09	0.11	0.27
B3	0.24	0.53	0.03	0.14	0.06	0.12
B4	0.14	0.14	0.14	0.14	0.43	0.05
B5	0.03	0.52	0.23	0.08	0.13	0.05
p	**14.93%**	**29.81%**	**9.58%**	**28.60%**	**17.09%**	

3.3 Implementation of the LSTM model

This section details the implementation of the LSTM model for translating PSL into text. The system components and the implementation process are detailed below.

Dataset construction. For this study, a dataset was built from the direct capture of PSL gestures. Data collection involved six volunteer participants (aged 18–50 years, three female and three male). Videos were recorded in controlled indoor conditions using a 1080p smartphone mobile phone camera positioned at 1.5 meters. Each participant performed eight common PSL words, repeating each sign 30 times, yielding a total of 1440 video samples, equivalent to 43,200 frames. All participants provided informed consent, and the study followed ethical guidelines ensuring anonymity and voluntary participation. Each video was divided into 30 frames.

Feature extraction. Videos were processed using "MediaPipe Holistic" to extract gestural features from the torso and hands. Each video frame was converted into a 258-element vector by concatenating "keypoints" from: (1) 33 torso "keypoints," with x, y, z coordinates and a visibility value (132 attributes); (2) 21 left hand "keypoints" with x, y, z coordinates (63 attributes); and (3) 21 right hand "keypoints" with the same structure (63 attributes), as shown in Figure 1.

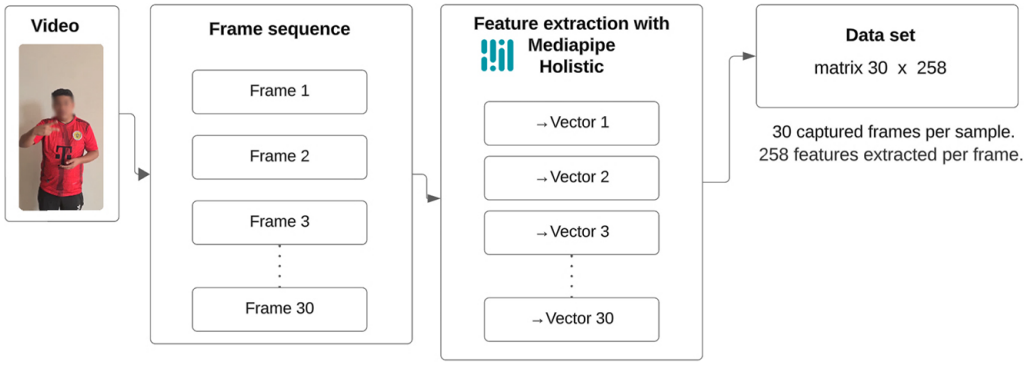

Fig. 1. Feature extraction process per sample

Model design. LSTM networks are designed to learn long-term dependencies in temporal sequences [27]. In this study, hierarchical LSTM architecture was implemented (see Figure 2) using "Keras" and "TensorFlow" for gesture recognition based on "keypoint" sequences. The model processed input sequences with a shape of (30, 258) through three LSTM layers with a progressive configuration of 128, 64, and 32 units, respectively. The first two layers preserve temporal information, and the last one generates the final vector. To prevent overfitting and stabilize the learning process, each LSTM layer used dropout (0.2) and batch normalization to improve training stability. The extracted temporal features were processed through two dense layers (64 and 32 units) with "ReLU" activation and dropout (0.3), culminating in an output layer with "softmax" activation for multiclass classification. The model was optimized using "Adam" and "categorical_crossentropy" as the loss function. Additionally, "EarlyStopping" and "ReduceLROnPlateau" callbacks monitored validation loss and adjusted the learning rate to reduce overfitting.

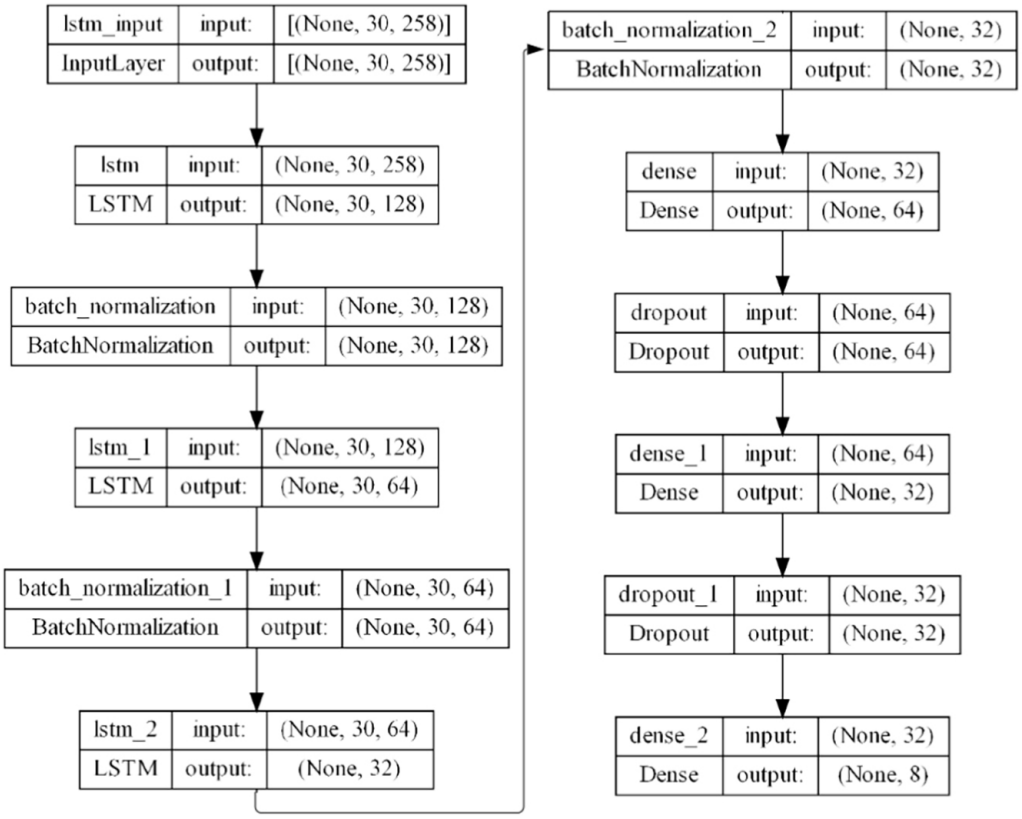

Fig. 2. Architecture of the proposed LSTM model

Model evaluation results. The dataset was split into 80% for training and 20% for validation. The average accuracy of the LSTM model reached 99.85% during the training phase and 100% during validation (see Figure 3a). From epoch 30 onward, both curves showed uniformity, indicating that the model had achieved stable learning and consistent performance in both training and validation. The loss in both phases gradually decreased over the epochs, ending with minimum values of 0.0018 and 0.003, respectively (see Figure 3b), which reflected a low prediction error and proper model fitting without signs of overfitting. Additionally, the model achieved high classification metrics, with precision, recall, and F1-score values of 99.56%, 99.55%, and 99.56% for training, and 99.43%, 99.40%, and 99.40% for validation. These results confirm the robustness and generalization ability of the LSTM model for gesture recognition tasks.

The confusion matrices in Figure 4 provide a detailed view of the model's performance in classifying eight PSL gestures. Both matrices show strong diagonal dominance, confirming high accuracy. The few off-diagonal values indicate minimal errors. In the training set (Figure 4a), nearly all gestures were correctly identified, with occasional confusions between "costar" and "uno," as well as between "kilogramo" and "uno." A similar pattern is observed in the validation set (see Figure 4b), where only one "costar" sample was misclassified as "hola." These errors are likely associated with visual similarities in hand configurations or motion trajectories between certain gestures. The results show that the LSTM model distinguishes visually and temporally similar gestures, confirming strong generalization. Nevertheless, the inclusion of additional samples could further reduce residual ambiguities and enhance the model's robustness.

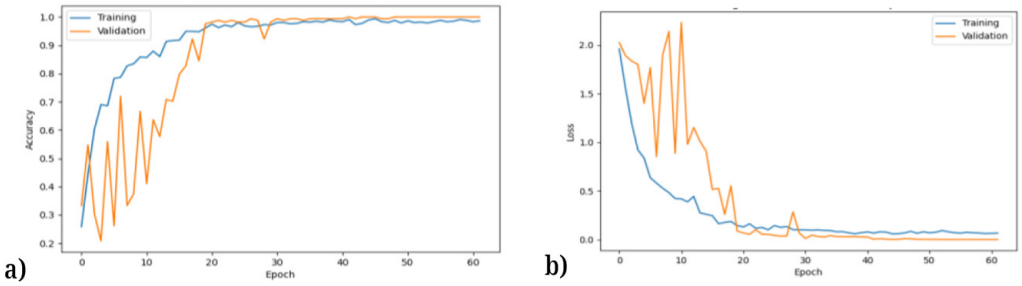

Fig. 3. Accuracy and loss during training and validation of the LSTM model

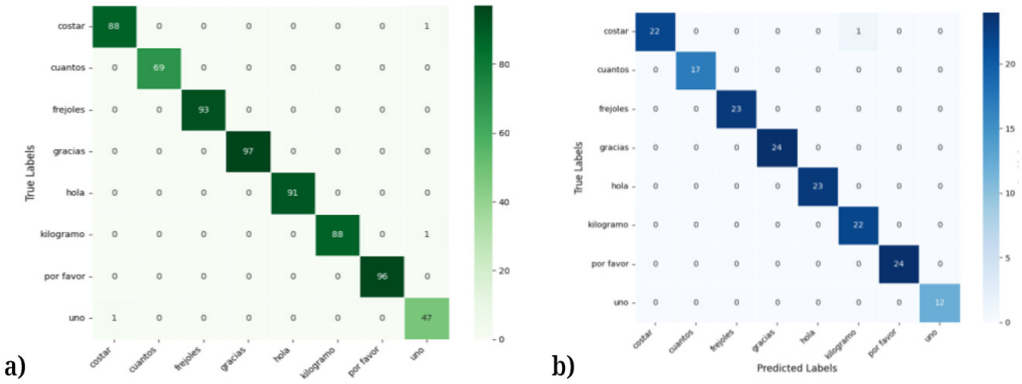

Fig. 4. Confusion matrix during training and validation of the LSTM model

3.4 Implementation of the LLM model

A prompt is an instruction or set of instructions provided to a language model to guide its response, specifying the tone, structure, and expected content [28]. Prompt engineering involves crafting inputs that guide the model's responses [29]. Figure 5 outlines the complete process of generating context-adapted responses, which was divided into four fundamental parts: 1) user input, 2) prompt construction, 3) prompt execution, and 4) prompt output.

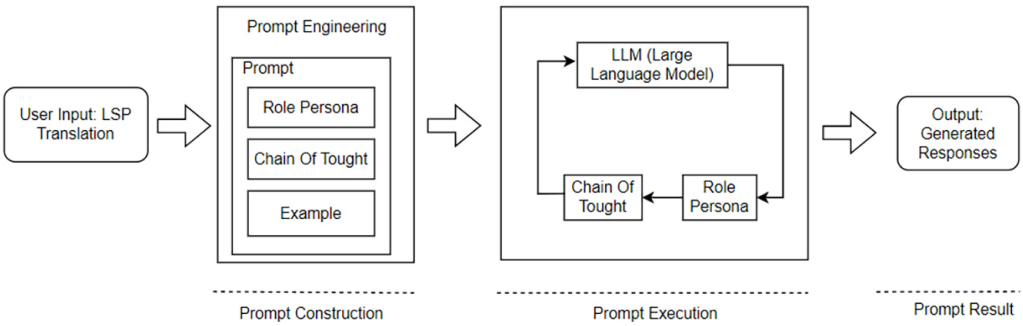

Fig. 5. Response generation process using the LLM

User input (PSL translation). The process begins with the user input, which is a message translated from PSL into text using an LSTM model. This input represents what the DHH individual wishes to communicate to the system.

Prompt construction (Prompt engineering). This involved constructing a prompt to guide the language model's response. The design included four main

elements: 1) Types of communication, which defined three levels of interaction (formal, neutral, and informal) depending on the context [30]; 2) Persona role (PR), which allowed for the simulation of cultural profiles to generate empathetic responses tailored to the user's communication style [28]; 3) Chain of Thought (CoT), a technique that guided the model through step-by-step reasoning to produce logical and structured responses [31]; and 4) Prompt execution through examples, which provided concrete references to improve the quality and coherence of the generated responses.

Prompt execution. The LLM (GPT-4o Mini) processes the input and prompt to generate a response.

Prompt output (result). A structured, logical, and culturally adapted response is generated to facilitate communication between DHH users and hearing individuals.

3.5 Mobile application development

The mobile application was developed using the Flutter framework and the Dart programming language for devices running the Android operating system. The backend was implemented in Python using the *FastAPI* framework, exposing RESTful endpoints that manage the core functionalities of the application. To facilitate communication, the OpenAI API was integrated with the GPT-4o Mini model, which is responsible for generating contextual responses based on user input. In parallel, a machine learning model was developed in Python using *MediaPipe* and LSTM neural networks, enabling the translation of gestures in PSL captured through the device's camera. Both components operate in an integrated and asynchronous manner, ensuring smooth real-time interaction.

Figure 6 shows the physical architecture of the system, which was organized into five layers: 1) the "user layer," composed of "hearing" individuals and DHH users; 2) the "device layer," represented by the smartphone that captures video and displays translated results; 3) the "connectivity layer," consisting of Wi-Fi or 4G networks that enable data transmission; 4) the "front-end layer," corresponding to the graphical interface developed with Flutter; and 5) the "back-end layer," where the LSTM and GPT-4o model APIs were hosted, responsible for translation and response generation, respectively.

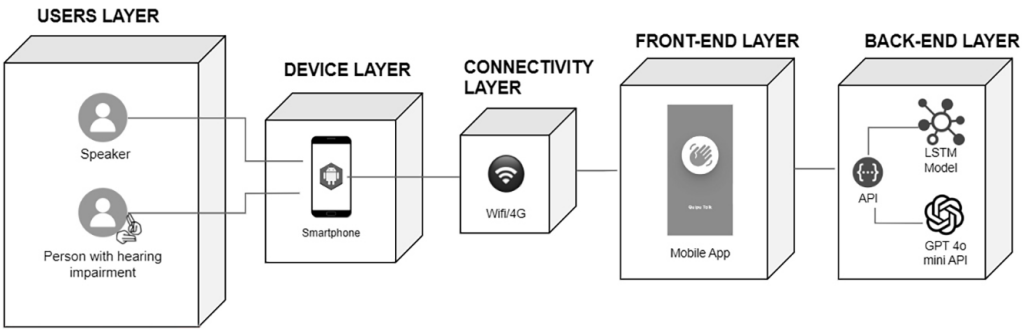

Fig. 6. Physical architecture

Additionally, the mobile application offered the following functionalities: 1) a main menu that serves as the home screen, where the user can start a conversation or access settings; 2) video recording of the DHH individual while performing sign language; 3) editing of the recorded video, allowing the user to trim or re-record the video if unsatisfied, or proceed to translation if the recording is acceptable; and 4) a conversation screen, where the translated message from the video is displayed along with suggested replies for quick selection, or the option for the "hearing" user to type or dictate a personalized response.

4 EXPERIMENTATION

The validation was conducted with 20 participants: 10 DHH who use PSL, recruited from the districts of Puente Piedra and Carabayllo (Peru), and 10 "hearing" participants with no prior knowledge of PSL. All participants were aged between 15 and 60 years. The study compared two communication modalities through consecutive experiments: traditional communication and communication assisted by the mobile application (refer to Table 3). Both experiments simulated a purchase interaction where DHH users were customers and hearing participants were sellers. In this scenario, the DHH participant requested a product, inquired about its price, and concluded the interaction by expressing gratitude. The metrics analyzed were response time (RT) and communicative clarity (CC).

Table 3. Experimental design for the comparative validation of the system

Experiment	Participants	Metric
1: Traditional Method	10 "Hearing" participants, 10 DHH	RT, CC
2: Using the Application	10 "Hearing" participants, 10 DHH	RT, CC

4.1 Experiment 1: traditional method

In the first experiment, DHH participants communicated directly with hearing participants without the aid of the application. The interaction flow is illustrated in Figure 7. (1) The sequence begins when the DHH user attempts to communicate and employs non-standardized strategies: (1.1) improvised gestures, (1.2) writing on paper or a cellphone, or (1.3) adapted signs with pointing to convey their message. (2) The "hearing" person interprets the message through deduction. (3) Based on their interpretation, the hearing user responds. If the communicative goal is achieved, (4) the interaction is deemed successful. Otherwise, (5) if no further signs are produced or mutual understanding is not reached, the interaction ends unsuccessfully. This cycle repeats each time a new message is received from the DHH user until the objective is achieved or the exchange is interrupted.

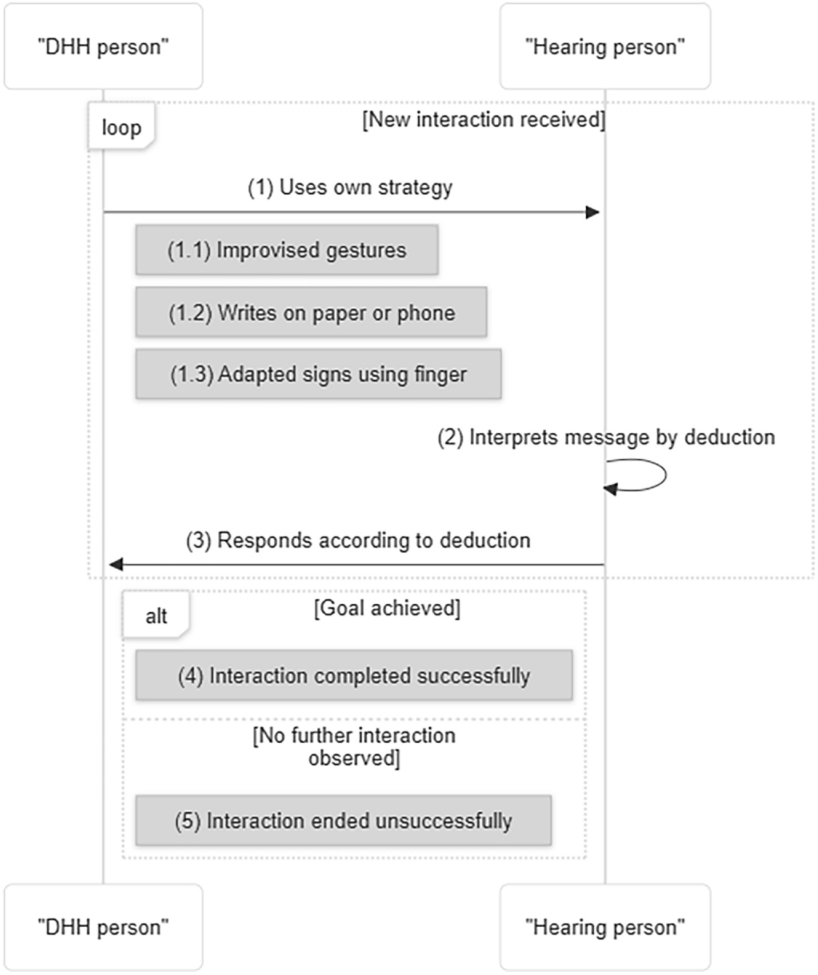

Fig. 7. Interaction flow in Experiment 1

4.2 Experiment 2: using the application

The second experiment evaluated the communication process using the developed mobile application. The interaction flow is depicted in Figure 8. (1) The DHH user performs signs in front of the device's camera, (2) the application captures the video, processes the gestures, and translates them into text, and (3) displays that text to the "hearing" person to ensure that the DHH user's message has been received. Next, (4) the application, supported by a language model (GPT-4o Mini), suggests several contextually appropriate responses to the "hearing" user; (5) the "hearing" user selects one of the suggested replies or types/dictates a personalized response, and (6) the application displays the final version of the response on screen, so that (7) the "hearing" user can show it directly to the DHH user. After this presentation, (8) the application asks the "hearing" user: "Would you like to continue?" If the user responds (9) "Yes," the process returns to step (1) and a new interaction cycle begins; if the response is (10) "No," the loop ends and the application executes (11) "End conversation," closing the session.

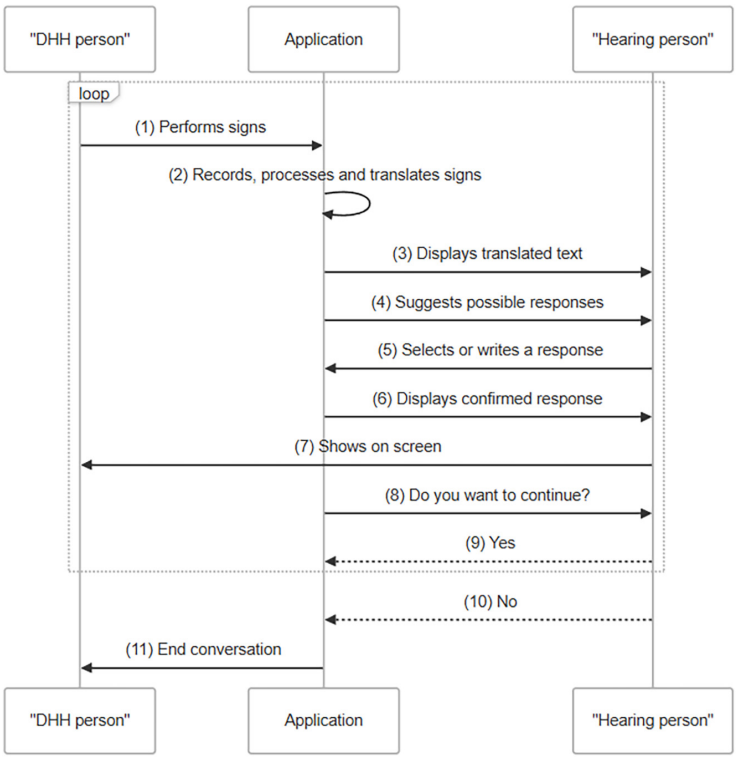

Fig. 8. Interaction flow in Experiment 2

To provide context for Experiment 2, the mobile application *Quipu Talk* was developed as an intermediary in the interaction between DHH users and "hearing" individuals. Upon starting, the user is presented with a main screen where they can choose between "PSL Translation" or "Settings" (see Figure 9a). When selecting the first option, the camera is activated to capture the DHH user's gesture (see Figure 9b), after which the application processes the video (see Figure 9c). Next, the "hearing" person views the resulting translation and is presented with several suggested responses or the option to "Customize Response" (see Figure 9d-e). The "hearing" person then writes or selects their response, which is displayed on the main screen (see Figure 9f).

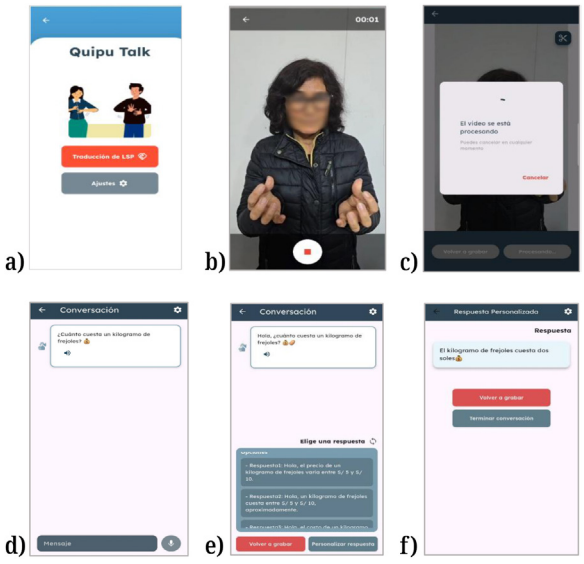

Fig. 9. User interfaces of main functionalities

4.3 Survey design

A questionnaire was developed for the "hearing" participants, consisting of closed-ended questions grouped according to the quality characteristics defined by ISO/IEC 25010 [32]. Each item was rated using a Likert scale (1 = Very poor, 2 = Poor, 3 = Neutral, 4 = Good, 5 = Very good) (refer to Table 4).

Table 4. Question design for hearing users

ID	Question
Interaction Capability	
Q01	Was the application easy to learn to use?
Q02	Were you able to operate and control the application easily (record, view translation, select a response)?
Q03	Were the application's functions and options clear and understandable without the need for external help?
Q04	Did you quickly understand that the application was suitable for facilitating communication with the deaf person?
Functional Adequacy	
Q05	Was the translated PSL message accurate and correct?
Q06	Were the suggested responses provided by the application relevant and useful for the conversation?
Q07	Did the application provide all the necessary functions for the simulated purchase interaction?
Performance Efficiency	
Q08	Did the application respond quickly (translation time, response generation)?
Quality in Use	
Q09	Overall, do you consider the application useful for facilitating communication in this situation?
Q10	Would you feel comfortable and confident using this application again in a real-life situation?

5 RESULTS AND DISCUSSION

5.1 Experiment 1: traditional method

The experiment was conducted in a controlled in-person session (see Figure 10). Each DHH participant received specific instructions to request a product. The interaction was considered successful if the customer obtained the requested item. Additionally, the duration of each interaction was recorded using a digital stopwatch, measuring the time elapsed from the beginning to the end of the exchange.

Fig. 10. Traditional method experiment

Table 5 presents the results of each interaction. While communicative success was reported in all cases, direct observation revealed that "hearing" participants frequently guessed the communicative intentions of the DHH users, indicating a lack of CC. The average interaction time was 95.93 seconds.

Table 5. Response times and success in each interaction of Experiment 1

Interaction	Hearing Participant	DHH Participant	Time (s)	Success
1	O1-EX1	DHH1-EX1	64.12	Yes
2	O2-EX1	DHH2-EX1	92.2	Yes
3	O3-EX1	DHH3-EX1	73.54	Yes
4	O4-EX1	DHH4-EX1	119.49	Yes
5	O5-EX1	DHH5-EX1	68.09	Yes
6	O6-EX1	DHH6-EX1	97	Yes
7	O7-EX1	DHH7-EX1	118	Yes
8	O8-EX1	DHH8-EX1	119	Yes
9	O9-EX1	DHH9-EX1	98.3	Yes
10	O10-EX1	DHH10-EX1	109.56	Yes

5.2 Experiment 2: using the application

This experiment evaluated the same communicative interaction using the mobile application (see Figure 11a). Prior to the session, the "hearing" participants were trained in the use of the application, and it was installed on their mobile devices. Hearing participants followed the complete interaction flow during the session. They began by launching the session (see Figure 9a), recorded the DHH participant performing signs (see Figure 9b), and processed the video to obtain a translation (see Figure 9c). Next, they reviewed the translated text (see Figure 9d), selected or

wrote a contextual response (see Figures 9d–e), and finally presented the response to the DHH participant (see Figure 9f). Figure 11b, c, and d show real screenshots of DHH participants captured through the application interface while performing various signs during the experiment.

Table 6 presents the response times for the 10 interactions conducted using the application, with an average time of 102.28 seconds. All interactions were successful in terms of communication. The responses provided by hearing participants were consistent with the original messages translated by the application.

Fig. 11. Experiment using the mobile application

Table 6. Response times and success in each interaction of Experiment 2

Interaction	Hearing Participant	DHH Participant	Time(s)	Success
1	O1-EX2	DHH1-EX2	147.25	Yes
2	O2-EX2	DHH2-EX2	97.43	Yes
3	O3-EX2	DHH3-EX2	82.67	Yes
4	O4-EX2	DHH4-EX2	130.18	Yes
5	O5-EX2	DHH5-EX2	98.52	Yes
6	O6-EX2	DHH6-EX2	69.36	Yes
7	O7-EX2	DHH7-EX2	80.91	Yes
8	O8-EX2	DHH8-EX2	100.74	Yes
9	O9-EX2	DHH9-EX2	113.82	Yes
10	O10-EX2	DHH10-EX2	101.59	Yes

The average RT using the application (102.28 seconds) was 6.6% higher than with the traditional method (95.93 seconds), due to video processing, which takes 2.56 s

per second of input. However, a significant improvement was observed in CC, as the ambiguities present in Experiment 1 were eliminated. In addition, greater comfort was noted among the DHH participants, evidenced by a more relaxed body posture during the interaction.

5.3 Comparison with similar works

Table 7 shows the comparison results between "QuipuTalk," and other research works designed for sign language translation. The attributes considered are Bidirectionality (A1), Dynamic Gesture Recognition (A2), and Technological Accessibility (A3). The first attribute indicates whether the system allows two-way communication between DHH users. The second attribute refers to whether the system processes video sequences (dynamic gestures) or only static images. The last attribute evaluates whether the application can operate on conventional mobile devices without requiring specialized hardware or complex configurations. Only "QuipuTalk" focuses on all three attributes, providing an integrated, accessible, and technologically feasible product that promotes real-time and inclusive communication adapted to the Peruvian context.

Table 7. Comparison with similar works

Works \ Attributes	A1	A2	A3
QuipuTalk	x	x	x
Terán-Quezada et al. [18]	–	x	–
González-Rodríguez et al. [6]	x	x	–
Kumari and Anand [11]	–	x	–
Marquez et al. [14]	–	–	x
Cerquín et al. [15]	–	x	x

5.4 Survey results

Figure 12 shows the results obtained by the 10 "hearing" participants regarding the "Interaction Capability" after using the application. The overall average was 4.4, indicating that "hearing" users perceived this capability as "Good." These results suggest that the application's interface was intuitive and user-friendly, enabling participants to understand its functionality and engage effectively with the system. Figure 13 shows the results for "Functional Adequacy," with an average score of 4.3, corresponding to a perception of the application as "Good." This indicates that "hearing" participants considered the core functionalities to adequately fulfill their intended purpose. Figure 14 shows an average score of 3.6 for "Performance Efficiency," corresponding to a "Good" perception by the hearing user. This reflects satisfactory performance, considering the video processing, gesture extraction, and translation using the LSTM model. With a response time of 2.56 seconds per second of video, "hearing" individuals positively value the balance between speed and communicative usefulness. Figure 15 shows an average score of 4.7 in "Quality in Use," corresponding to a "Very Good" perception by hearing participants. The result

reflects high confidence in its application in real-world contexts and highlights its potential to enhance social inclusion.

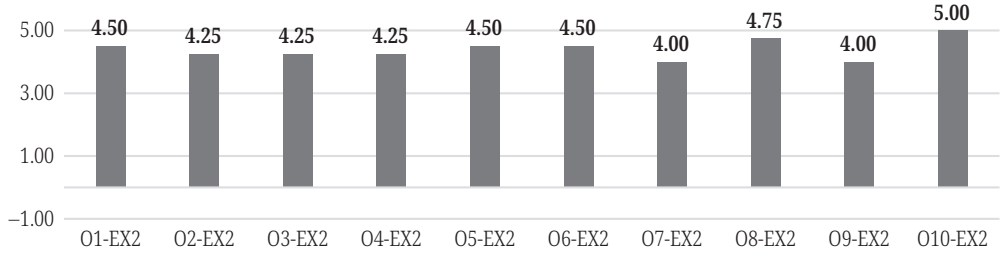

Fig. 12. Rating of "Interaction Capability"

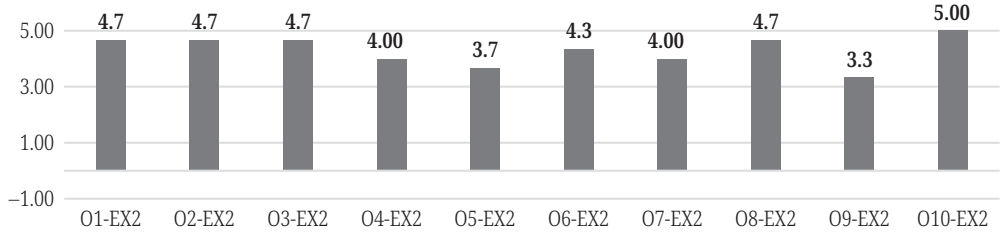

Fig. 13. Rating of "Functional Adequacy"

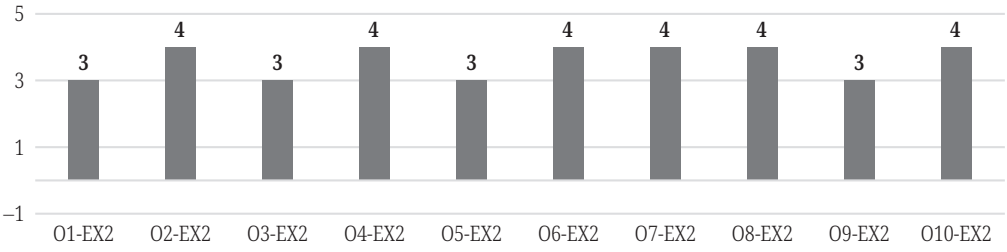

Fig. 14. Rating of "Performance Efficiency"

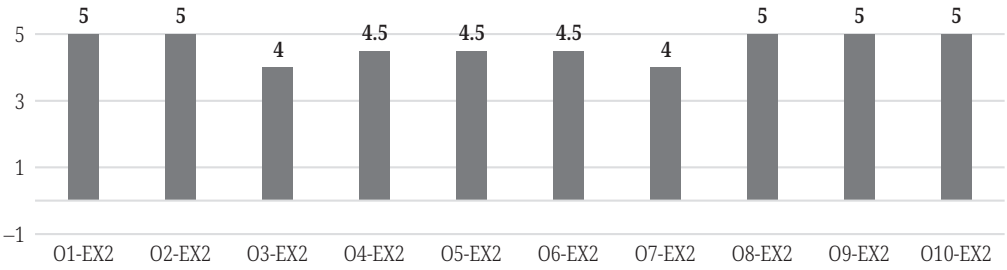

Fig. 15. Rating of "Quality in Use"

6 CONCLUSIONS AND FUTURE WORK

This study proposed the development of a mobile application to facilitate communication between "hearing" individuals and DHH users, given that most "hearing" participants do not master PSL. A neural network was used to translate PSL into text, and the GPT-4 model was integrated to generate fast and contextual responses, this integration being the main contribution by improving communicative clarity and accessibility. The proposal was developed in five phases: selection of neural

network techniques, selection of the machine learning model, implementation of the LSTM model, implementation of the LLM model, and development of the mobile application. The experimentation involved 20 participants: 10 "hearing" individuals and 10 DHH participants, who took part in a simulated purchase scenario to evaluate interaction time (RT) and communicative clarity (CC). The results of both experiments showed that CC was successfully achieved. Regarding RT, Experiment 1 recorded an average of 95.93 seconds, while Experiment 2 recorded 102.28 seconds. This 6.6% increase was due to video processing time. Despite a slight delay, communication quality improved, with participants showing relaxed posture and reduced stress. Regarding the post-Experiment 2 survey conducted with "hearing" participants, the results showed high acceptance of the system, with scores of 4.4 (Good), 4.3 (Good), 3.6 (Good), and 4.7 (Very Good) in the categories of Interaction Capability, Functional Adequacy, Performance Efficiency, Usability, Reliability, and Quality in Use, respectively. Its implications extend to the technological and social domains, demonstrating that multimodal AI-based systems can effectively reduce communication barriers and serve as a scalable foundation for future applications in other languages and contexts.

As future work, we propose expanding the application to other communicative scenarios such as educational settings, healthcare services, and banking procedures, among others. This would complement and enhance communication with DHH individuals.

7 ACKNOWLEDGMENTS

We thank the DHH and "hearing" participants who took part in the experiments conducted for this research. We also thank the Research Department of the XXX University for providing the necessary resources to carry out this study through the YYY-Expost-2025-2 incentive.

8 REFERENCES

[1] C. L. Haukedal, O. B. Wie, S. K. Schauber, B. Lyxell, E. M. Fitzpatrick, and J. von Koss Torkildsen, "Social communication and quality of life in children using hearing aids," *Int. J Pediatr. Otorhinolaryngol.*, vol. 152, p. 111000, 2022. https://doi.org/10.1016/j.ijporl.2021.111000

[2] World Health Organization, "World report on hearing," 2021. Accessed: May 16, 2025. [Online]. Available: https://www.who.int/publications/i/item/9789240020481

[3] El Instituto Nacional de Estadística e Informática (INEI), "Perfil sociodemográfico de la población con discapacidad, 2017," 2019. https://www.inei.gob.pe/media/MenuRecursivo/publicaciones_digitales/Est/Lib1675/libro.pdf

[4] H. J. Park, Y. Lee, and J. G. Ko, "Enabling real-time sign language translation on mobile platforms with on-board depth cameras," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 5, no. 2, pp. 1–30, 2021. https://doi.org/10.1145/3463498

[5] D. T. Mosa, N. A. Nasef, M. A. Lotfy, A. A. Abohany, R. M. Essa, and A. Salem, "A real-time Arabic avatar for deaf–mute community using attention mechanism," *Neural Comput. Appl.*, vol. 35, no. 29, pp. 21709–21723, 2023. https://doi.org/10.1007/s00521-023-08858-6

[6] J. R. González-Rodríguez, D. M. Córdova-Esparza, J. Terven, and J. A. Romero-González, "Towards a bidirectional Mexican sign language–Spanish translation system: A deep learning approach," *Technologies (Basel)*, vol. 12, no. 1, p. 7, 2024. https://doi.org/10.3390/technologies12010007

[7] M. Cabanillas-Carbonell, P. Cusi-Ruiz, D. Prudencio-Galvez, and J. L. H. Salazar, "Mobile application with augmented reality to improve the process of learning sign language," *International Journal of Interactive Mobile Technologies*, vol. 16, no. 11, pp. 51–64, 2022. https://doi.org/10.3991/ijim.v16i11.29717

[8] D. Novaliendry, K. Budayawan, R. Auvi, B. R. Fajri, and Y. Huda, "Design of sign language learning media based on virtual reality," *International Journal of Online and Biomedical Engineering*, vol. 19, no. 16, pp. 111–126, 2023. https://doi.org/10.3991/ijoe.v19i16.44671

[9] M. Alaftekin, I. Pacal, and K. Cicek, "Real-time sign language recognition based on YOLO algorithm," *Neural. Comput. Appl.*, vol. 36, no. 14, pp. 7609–7624, 2024. https://doi.org/10.1007/s00521-024-09503-6

[10] H. V. Das, K. Mohan, L. Paul, S. Kumaresan, and C. S. Nair, "Transforming consulting atmosphere with Indian sign language translation," *Multimed. Tools Appl.*, vol. 83, no. 5, pp. 13543–13555, 2024. https://doi.org/10.1007/s11042-023-15214-2

[11] D. Kumari and R. S. Anand, "Isolated video-based sign language recognition using a hybrid CNN-LSTM framework based on attention mechanism," *Electronics (Switzerland)*, vol. 13, no. 7, p. 1229, 2024. https://doi.org/10.3390/electronics13071229

[12] K. Kozyra, K. Trzyniec, E. Popardowski, and M. Stachurska, "Application for recognizing sign language gestures based on an artificial neural network," *Sensors*, vol. 22, no. 24, p. 9864, 2022. https://doi.org/10.3390/s22249864

[13] D. Bendarkar, P. Somase, P. Rebari, R. Paturkar, and A. Khan, "Web based recognition and translation of American sign language with CNN and RNN," *International Journal of Online and Biomedical Engineering*, vol. 17, no. 1, pp. 34–50, 2021. https://doi.org/10.3991/ijoe.v17i01.18585

[14] B. Y. Marquez, A. Alanis, A. Quezada, and J. S. Magdaleno-Palencia, "Development of a mobile application with artificial intelligence for Mexican sign language recognition," *International Journal of Interactive Mobile Technologies*, vol. 19, no. 9, pp. 122–139, 2025. https://doi.org/10.3991/ijim.v19i09.54205

[15] A. D. B. Cerquín, J. A. T. Guevara, and C. Ovalle, "Mobile application for continuous recognition and classification of sign language images through deep learning," *International Journal of Interactive Mobile Technologies*, vol. 19, no. 7, pp. 4–21, 2025. https://doi.org/10.3991/ijim.v19i07.52853

[16] R. Patil, V. Patil, A. Bahuguna, and G. Datkhile, "Indian sign language recognition using convolutional neural network," *ITM Web of Conferences*, vol. 40, p. 03004, 2021. https://doi.org/10.1051/itmconf/20214003004

[17] Y. Saleh and G. F. Issa, "Arabic sign language recognition through deep neural networks fine-tuning," *International Journal of Online and Biomedical Engineering*, vol. 16, no. 5, pp. 71–83, 2020. https://doi.org/10.3991/ijoe.v16i05.13087

[18] A. A. Teran-Quezada, V. Lopez-Cabrera, J. C. Rangel, and J. E. Sanchez-Galan, "Sign-to-text translation from Panamanian sign language to Spanish in continuous capture mode with deep neural networks," *Big Data and Cognitive Computing*, vol. 8, no. 3, p. 25, 2024. https://doi.org/10.3390/bdcc8030025

[19] S. Daga, A. Dusane, and D. Bobby, "With You-indian sign language detection and alert system," in *2024 International Conference on Emerging Smart Computing and Informatics (ESCI 2024)*, 2024. https://doi.org/10.1109/ESCI59607.2024.10497366

[20] S. Sharma and K. Kumar, "ASL-3DCNN: American sign language recognition technique using 3-D convolutional neural networks," *Multimed. Tools Appl.*, vol. 80, no. 17, pp. 26319–26331, 2021. https://doi.org/10.1007/s11042-021-10768-5

[21] T. B. Brown *et al.*, "Language models are few-shot learners," *arXiv preprint arXiv:2005.14165* 2020. https://doi.org/10.48550/arXiv.2005.14165

[22] OpenAI, "Models," 2024. Accessed: Sep. 24, 2024. [Online]. Available: https://platform.openai.com/docs/models/

[23] Anthropic, "Claude 3.5 Sonnet," 2024. Accessed: Sep. 24, 2024. [Online]. Available: https://www.anthropic.com/news/claude-3-5-sonnet

[24] Google DeepMind, "Gemini Pro," 2024. Accessed: Sep. 24, 2024. [Online]. Available: https://deepmind.google/technologies/gemini/pro/

[25] W. Xiong *et al.*, "Effective long-context scaling of foundation models," in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, 2024, pp. 4643–4663.

[26] D. Hendrycks *et al.*, "Measuring massive multitask language understanding," *arXiv preprint arXiv:2009.03300*, 2020. https://doi.org/10.48550/arXiv.2009.03300

[27] W. Wang, J. Shao, and H. Jumahong, "Fuzzy inference-based LSTM for long-term time series prediction," *Sci. Rep.*, vol. 13, no. 1, 2023. https://doi.org/10.1038/s41598-023-47812-3

[28] L. Henrickson and A. Meroño-Peñuela, "Prompting meaning: A hermeneutic approach to optimising prompt engineering with ChatGPT," *AI Soc.*, vol. 40, no. 2, pp. 903–918, 2023. https://doi.org/10.1007/s00146-023-01752-8

[29] X. Amatriain, "Prompt design and engineering: Introduction and advanced methods," *arXiv preprint arXiv:2401.14423*, 2024. Accessed: Jun. 08, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2401.14423

[30] N. Sheykh, A. E. Kandlousi, A. J. Ali, and A. Abdollahi, "Organizational citizenship behavior in concern of communication satisfaction: The role of the formal and informal communication," *International Journal of Business and Management*, vol. 5, no. 10, 2010. https://doi.org/10.5539/ijbm.v5n10p51

[31] J. Wei *et al.*, "Chain-of-thought prompting elicits reasoning in large language models chain-of-thought prompting," *arXiv preprint arXiv:2201.11903*, 2022. https://doi.org/10.48550/arXiv.2201.11903

[32] International Organization for Standardization (ISO/IEC), "ISO/IEC 25010," 2025. [Online]. Available: https://www.iso25000.com/index.php/normas-iso-25000/iso-25010

9 AUTHORS

Alyne Regalado-Morales is a Software Engineer from the Peruvian University of Applied Sciences. Her professional interests include front-end development, UX/UI design, and mobile application development. She has participated in projects where she applied modern frameworks to create intuitive and user-centered digital experiences (E-mail: u20201a976@upc.edu.pe).

Axel Fiestas is a Software Engineer from the Peruvian University of Applied Sciences. He works as a Desktop Developer specialized in C#, .NET, PSQL, and Actian Zen. His areas of interest include software architecture, database optimization, and the integration of artificial intelligence systems (E-mail: u20201b908@upc.edu.pe).

Lenis Wong is a Professor and researcher of Software Engineering and Information Systems Engineering at the Peruvian University of Applied Sciences, Peru. She holds a PhD in Systems Engineering and Computer Science. She has published several international peer-reviewed scientific articles in different multidisciplinary areas such as: ML, DL, IoT, e-Health, Software Engineering, Requirements Engineering, Cloud Computing, E-Learning, Gamification, Cyberattacks, Natural Language Processing, Networks and Blockchain Technologies (E-mail: pcsilewo@upc.edu.pe).