Online-Journals.org

iJIM International Journal of Interactive Mobile Technologies

iJIM | eISSN: 1865-7923 | Vol. 19 No. 7 (2025) | OPEN ACCESS

https://doi.org/10.3991/ijim.v19i07.52853

PAPER

# Mobile Application for Continuous Recognition and Classification of Sign Language Images through Deep Learning

Angel Diego Briones Cerquín, Johan Alonso Tumay Guevara, Christian Ovalle(✉)

Universidad Tecnológica del Perú, Lima, Perú

dovalle@utp.edu.pe

**ABSTRACT**

Throughout the world, sign languages (SL) present significant challenges for effective communication in everyday environments and technological applications. In the field of SL recognition (SLR) using artificial intelligence (AI), two approaches have been developed: isolated SLR (ISLR) and continuous SLR (CSLR). To overcome the limitations of CSLR in SL, we developed a mobile application that integrates an AI-based algorithm in Python, designed to capture and analyze sign sequences through the device's camera. The application facilitates the creation of a continuous database containing 14 dynamic signs, with 240 videos per sign, resulting in a total of 3360 videos and 50,400 frames. We used a neural network model based on the long short-term memory (LSTM) architecture to improve accuracy in sign identification and promote inclusive communication in digital environments. The model achieved 99.80% accuracy during training and 99.40% in testing, with overall accuracy, recall, and F1-score metrics above 99%. These results evidence the effectiveness of the mobile application and the LSTM model in recognizing, classifying, and translating basic SLP utterances in real time, demonstrating its ability to generalize and avoid overfitting and contributing to more inclusive and accessible communication.

**KEYWORDS**

continuous sign, sign language (SL), long short-term memory (LSTM), deep learning (DL), mobile application

## 1 INTRODUCTION

Currently, sign language recognition (SLR) represents a significant challenge in communication, especially in continuous sign recognition [1], [2], [3]. While current methods have made progress in recognizing isolated signs, they need to be improved in identifying and classifying continuous sign sequences. These limitations include insufficient accuracy and a lack of practical mobile applications

Briones Cerquín, A.D., Tumay Guevara, J.A., Ovalle, C. (2025). Mobile Application for Continuous Recognition and Classification of Sign Language Images through Deep Learning. *International Journal of Interactive Mobile Technologies (iJIM)*, 19(7), pp. 4–21. https://doi.org/10.3991/ijim.v19i07.52853

Article submitted 2024-10-14. Revision uploaded 2025-01-16. Final acceptance 2025-02-19.

that facilitate real-time communication. Globally, many countries strive to eliminate communication barriers with deaf individuals, as these barriers significantly impede the fundamental right to education and social inclusion. Promoting teaching and disseminating sign languages (SL) is essential to provide equal educational and employment opportunities, supported by various laws that officially recognize these languages [4]. At the international level, 76% of public institutions and 83% of private institutions in various countries lack the necessary conditions to provide educational services to deaf students [5], [6]. Numerous complaints exist about the communication barriers faced by individuals with this disability in education, healthcare, and other services due to the lack of qualified SL interpreters. Additionally, individuals with this condition are underrepresented in academia, with figures reflecting their limited participation [7].

Sign languages are essential for approximately 466 million deaf and hard-of-hearing individuals worldwide [8–14]. They rely on visual signs expressed through hand shapes, movements, and facial expressions [15–20]. With over 300 SLs in use globally [5], [21–24], users often face communication barriers and social isolation due to the lack of practical tools to facilitate communication [25–28]. Artificial intelligence (AI) for SLR includes two main approaches: Isolated SLR (ISLR), which analyzes static signs, and Continuous SLR (CSLR), which handles dynamic sign sequences for real-time recognition [28]. CSLR facilitates more natural and fluid communication, although it still faces challenges. Key datasets for CSLR include RWTH-PHOENIX for German SL [13], [15], [16], [21], [25], [29], SIGNUM for German SL [10], ASLG-PC12 for Australian SL [25], [29], How2Sign for American SL [9], and MEDIAPI-SKEL for French SL [17]. These datasets are critical for developing accurate and efficient SLR systems.

In the case of SL, research has mainly focused on isolated sign recognition using the ISLR approach. The lack of extensive datasets has limited the development of models for continuous sequence recognition. Several strategies have been employed to address this limitation. Custom datasets have been created, including static images of the alphabet, numbers, and essential words [27], [28], [30], [31], as well as motion sensors such as curvature sensors [30] and Kinect, which captures hand movements and enables translation via 3D avatars [7]. In 2014, the Grammar and Signs research group at PUCP developed a specific continuous SL dataset of 718 video clips. This dataset includes a limited repertoire of signs covering nouns such as "cinema," "horror," "park," "house," "couple," "night," and "heap" [5], [8]. Regarding neural network architectures, various options have been explored. ResNet50-V2 and unique CNNs have shown high accuracy in recent studies [30], [31], [32]. On the other hand, long short-term memory (LSTM) models, such as BiLSTM, ConvLSTM, LSTM, and CNN-LSTM, have proven particularly effective in capturing temporal features of signs. These models have achieved notable accuracies, with CNN-LSTM reaching up to 95.43% in tests [33]. Concerning mobile applications, a study [34] developed a mobile app to teach Arabic SL to deaf children in the Arab world using gamification methods. First, the app was developed after reviewing existing literature and applications; then, it was tested with 10 deaf children, comparing learning outcomes with traditional methods. Results showed that children using the app achieved better scores on tests. Additionally, a study [35] explored the potential of a mobile system to help deaf individuals and others communicate and learn with portable devices. The pedagogical evaluation of a prototype app using images and videos for recognition showed that the app is easy to remember and learn. However, despite AI advancements in sign recognition, current approaches still face critical issues, such as variability in sign expression and maintaining accuracy in continuous sequences.

Therefore, more continuous datasets with extensive vocabulary are needed to recognize and translate SL using CSLR. Previous efforts have focused on isolated signs and databases limited to essential words. Developing a continuous dataset that includes nouns and essential words and phrases in SL is essential to address these limitations. Recently, deep learning (DL) methods have significantly revolutionized the field of learning algorithms in various areas of AI [12], [36], [37], [38]. In particular, they are crucial in providing technological assistance to individuals with hearing loss or severe auditory disabilities today. These technologies enable them to communicate in their language and improve human-computer interaction in the digital environment [25], [31].

This study proposes an innovative solution, which is developing a mobile application for CSLR. The application utilizes an AI-based algorithm developed in Python, which employs a webcam to capture and classify dynamic sign sequences. This approach addresses the gap in CSLR and provides a practical and accessible tool to enhance communication between deaf individuals and those who do not know SL. Thus, this approach addresses the need for a solution that improves the accuracy of SLR and offers a practical tool for daily communication. The main objective is to develop and validate a model that enables continuous and accurate translation of SL in a format accessible to users through a mobile application. This advancement has the potential to bridge the gap in SLR and significantly contribute to improving communication for the deaf community worldwide.

## 2 METHODOLOGY

This study highlights the importance of improving education for students with hearing loss or severe auditory disabilities, marking a significant advancement in technological assistance. The goal is to enable effective communication in their language and optimize their interaction with digital devices in the educational environment [25], [31]. To address this challenge, a mobile application has been developed with a continuous database of sequential sign images in SL using a Python-based algorithm. A neural network model based on the LSTM architecture has been implemented using Python's DL techniques. The choice of the LSTM architecture is due to its proven capability to process and model data sequences with high precision, which is crucial for real-time recognition and translation of fundamental expressions in SL [17], [26], and [39].

Notably, the application uses the device's camera to capture and classify dynamic sign sequences, offering a practical and accessible tool to improve communication between deaf individuals and those who do not know SL. This model focuses on recognizing dynamic signs, and the expressions included in this study cover standard greetings and phrases such as "Hello," "Good morning," "Good afternoon," "Good evening," and "How are you?" "I am fine," "I am not well," "so-so," "sorry," "please," "help me," "what time is it?" "thank you," and "goodbye."

Figure 1 illustrates the five phases of this study, which include the development of the LSTM model and the implementation of the mobile application. The primary focus is establishing a system that continuously recognizes, classifies, and translates images representing basic SL expressions, thereby providing a robust and reliable solution for communication in the digital and educational environment.

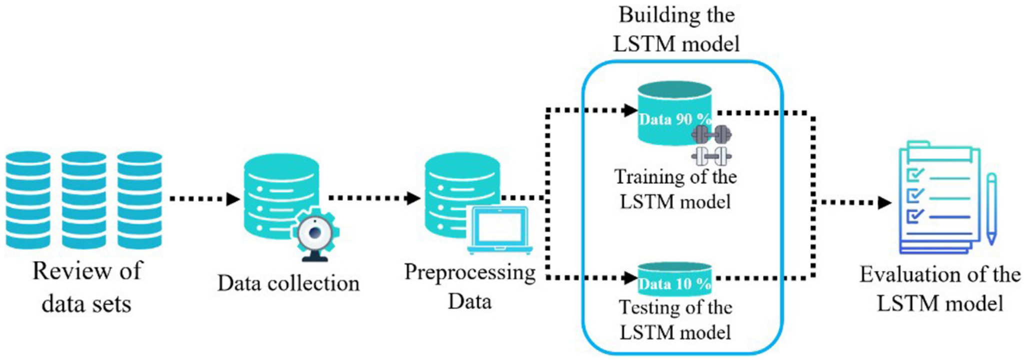


**Fig. 1.** Schematic diagram of LSTM neural network model

## 2.1 Review of data sets

Below is a summary of the datasets used in previous research. However, a paucity of datasets focuses on whole-sentence recognition and translation. So far, continuous video-based datasets have been created for SLR and translation of a limited number of words in different languages. These datasets are detailed in Table 1.

**Table 1.** A summary of the continuous data sets used in the CSLR

| Description | Type and Size | Name | SL | Reference |
|---|---|---|---|---|
| Complete sentences | 80 000 videos | How2Sign | American | [1] |
| Words (25) | 200 videos | – | Arabic | [19] |
| Word (40) | 80 00 videos | – | Arabic | [19] |
| Words (30) | 1500 videos | – | Argentinian | [19] |
| Words (12) | 12 videos | ASLGPC12 | Australian | [18], [27] |
| Words (455) | – | SIGNUM | German | [2] |
| Letters (30) and numbers (1–5) | 1400 videos | – | German | [19] |
| Complete sentences and Words (1081) | 11000 videos | RWTH-PHOENIX | German | [5], [7], [8], [14], [18], [27] |
| Words (10) | 1080 videos | – | Indian | [19] |
| Words (10) | 500 videos | – | Thai | [19] |

## 2.2 Dataset collection

This study focuses on improving communication for individuals with hearing loss in the educational sector, a critical area according to statistics from government agencies. A dataset comprising 14 dynamic gestures representing basic expressions in SL was compiled to advance this goal. This dataset is crucial for training and evaluating the sign recognition model and for the mobile application. The data collection process was meticulously executed using an automated algorithm programmed in Python within the Visual Studio Code environment. The OpenCV and MediaPipe (MP) libraries, similar to those used in [29], were instrumental in this process. Videos capturing dynamic signs were recorded using the application's camera, and these were then processed and stored as sequential images. The recordings were conducted under a variety of background and lighting conditions to ensure the dataset's comprehensive representation. The result was a balanced dataset comprising 3,360 videos, equivalent to 50,400 frames. Each of the 14 dynamic signs is represented by 240 sequences of images, with each sequence consisting of 15 frames. For training the neural network, 90% of these frames (45,360) were used, while the remaining 10% (5,040) were reserved for testing and performance evaluation. This continuous dataset, based on frame sequences, is essential for real-time recognition and translation of everyday SL expressions, as detailed in Table 2.

**Table 2.** Continuous dataset for the following basic SL expressions

| Description | Content | Type and Size |
|---|---|---|
| Words (4) | • hello | 3360 videos |
| | • excuse me | |
| | • thank you | |
| | • goodbye | |
| Phrases (10) | • good morning | |
| | • good afternoon | |
| | • good evening | |
| | • what time is it? | |
| | • how are you | |
| | • I'm well | |
| | • I don't feel well | |
| | • so-so | |
| | • please | |
| | • can you help me? | |

## 2.3 Preprocessing data

A) **Capture samples:** Data preprocessing aims to extract relevant information and reduce processing costs. The process begins by converting videos into frames [35], [36], organized into folders with descriptive names based on the dynamic SL signs in our dataset. To accomplish this, essential libraries such as OpenCV, MP, NumPy, and Pandas are installed in the Visual Studio Code editor, which allows for easier debugging, compiling, and editing of code [37], as illustrated in Figure 2. These libraries include the Holistic model (a pre-trained DL model) for tasks such as facial landmark detection, pose estimation, and gesture recognition. The draw_landmarks library is also imported for visualization purposes. Additional resources such as FACEMESH_CONTOURS, POSE_CONNECTIONS, and HAND_CONNECTIONS are used to detect and draw facial contours, poses, and hands with the Holistic model. Furthermore, NumPy and Pandas library structure and analyze the datasets, respectively. Functions and constants such as ROOT_PATH and FRAME_ACTIONS_PATH is created to refer to the location of frame samples within the "frame_actions" folder.

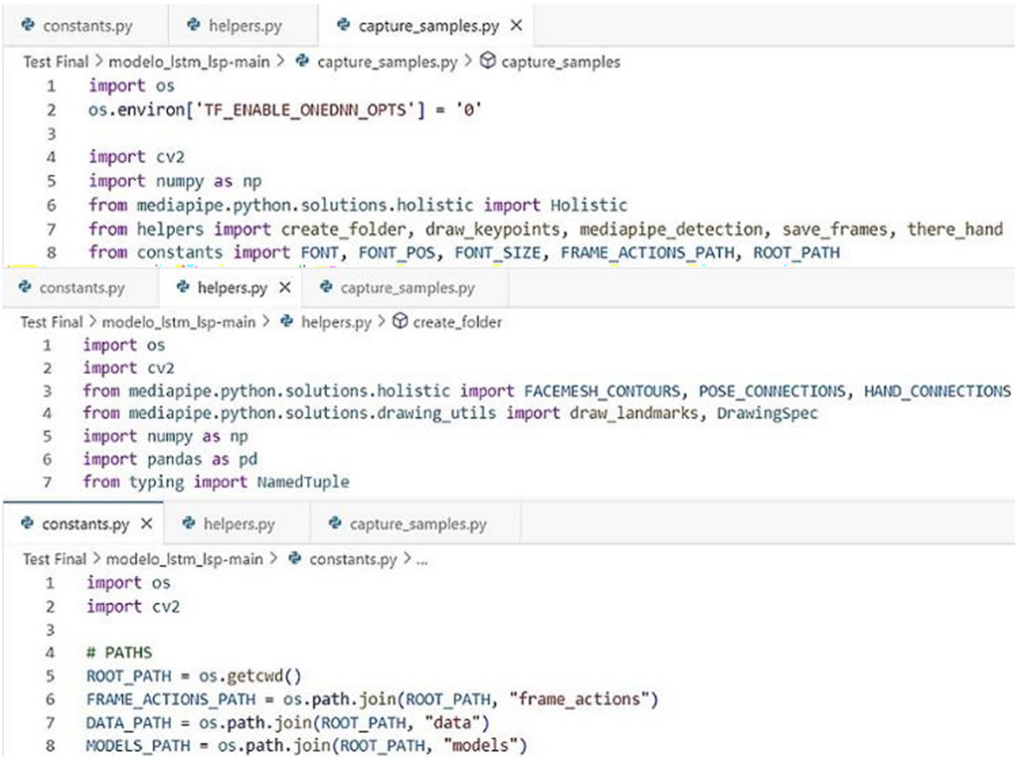


**Fig. 2.** Import libraries, constants and functions for capturing samples (frames)

Thus, the "capture_samples" code captures video samples representing the 14 basic SL expressions using dynamic signs. It starts by getting the current number of samples in a specific folder. Then, it initializes variables to keep count of samples and frames. It uses the Holistic MP model and captures video from the computer camera. In a loop, it processes each frame of the video and checks for hand detection. If hands are detected, the counters are incremented, and the frame is added to a set of frames. If no hands are detected, it checks for enough frames to form a valid sample. If so, the frame sequence is saved in a single output folder. Then, the variables are reset to start a new capture, a message is displayed, and critical points are drawn on the image. The capture is stopped by pressing the 'q' key (see Figure 3).

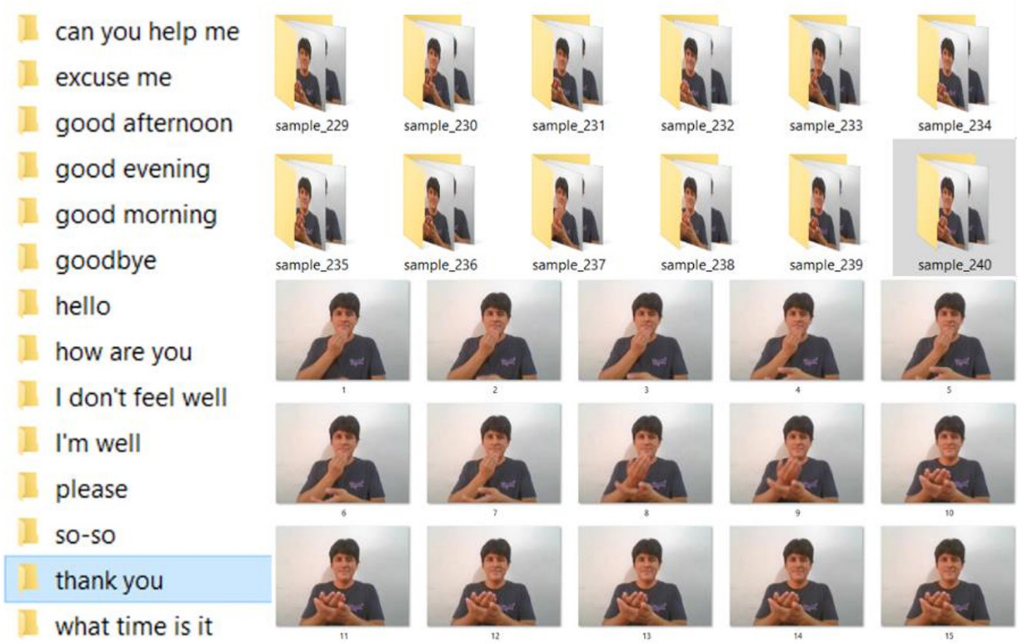


**Fig. 3.** Capture_samples code for our Data Collection

In this way, a balanced database containing the 14 dynamic signals mentioned above is created. Each sign is represented by 240 sequences of 15 frames, totaling 3600 frames per sign. This results in a dataset of 50400 frames in total, covering actions performed with both one hand and both hands and representing the basic LSP expressions. Figure 4 shows the 14 folders created for each of the 14 signs, where the 15 frames corresponding to the sign "thank you," for example, can be observed.


**Fig. 4.** Sample of SL dynamic words and phrases signs

**B) Create key points (KP):** Hand position identification and pose estimation are crucial elements for recognizing and understanding SL movements. To address these challenges, MP is employed as an effective tool that calculates poses for each frame and extracts key points in three dimensions (X, Y, and Z) for the hands, face, and body pose. A total of 21 key points is calculated for each hand (63 in total), 468 key points for the face, and 33 key points for the body pose, resulting in a total of 132 key points, which provide a comprehensive representation of the hand, face, and body pose in each frame.

The 'create_keypoints' code starts by importing necessary libraries and setting up a TensorFlow environment variable. It then defines a function called 'create_keypoints' that takes the paths for frame directories ('frames_path') and the save location for key points ('save_path'). This function uses the Holistic model from MP to process each frame, extracting critical points with 'get_keypoints(model_holistic, sample_path).' These key points are inserted into an empty Data Frame and saved as an HDF file using 'data.to_hdf()'. The code constructs the path for word directories ('words_path') using 'os.path.join(ROOT_PATH, FRAME_ACTIONS_PATH)' and iterates through these directories with 'os.listdir()'. For each word, it sets up the HDF directory and file paths ('hdf_path'), displays a message about starting key point extraction, calls the 'create_keypoints' function, and then shows a message indicating completion. This process extracts and saves key points for frame samples in the word and phrase dataset, as shown in Figure 5.

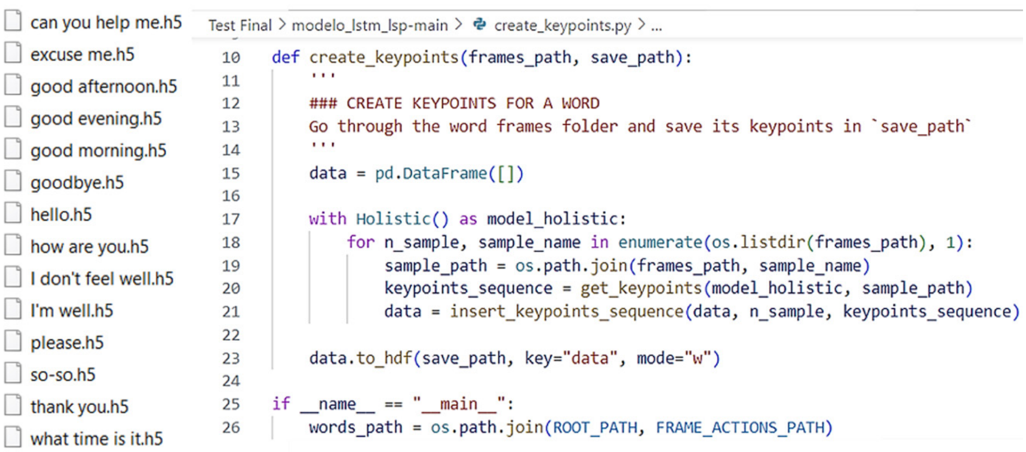


**Fig. 5.** Create_keypoints code to store the KP of our dataset

## 2.4 Building the long short-term memory model

Determining a model's optimal number of LSTM layers is challenging due to problem complexity, data availability, and computational constraints. To achieve good accuracy in SLR and translation, the parameters from [40] are used. The Keras sequential model, 'model = Sequential(),' is employed to build neural networks by stacking layers sequentially. An LSTM layer with 64 or 128 units is added to learn and extract features from the input data. The 'return_sequences = True' option outputs complete sequences at each time step. Dense layers with ReLU activation are included to capture non-linear patterns in SL signs. The LSTM layer expects an input matrix of dimensions (15, 1662), meaning it processes 15 frames with 1662 features each at each time step ('input_shape = (15,1662)').

On the other hand, to compile the model, the following command is used: optimizer = 'adam', indicating that the Adam optimizer, known for its efficiency in handling problems with large datasets and high dimensionality, will be employed to adjust the weights and parameters of the model during the training process. This choice is made to reassure you about the model's robustness in dealing with complex data. Additionally, the loss function binary_crossentropy is specified for training the neural network model, facilitating more accurate predictions. Finally, the evaluation metric accuracy is used to represent the model's precision in classifying training and evaluation data. For more details (see Figure 6).

constants.py | helpers.py | capture_samples.py | create_keypoints.py | model.py ×

Test Final > modelo_lstm_lsp-main > model.py > ...

```
NUM_EPOCH = 100

def get_model(output_lenght: int):
    model = Sequential()
    model.add(LSTM(64, return_sequences=True, activation='relu', input_shape=(MAX_LENGTH_FRAMES, LENGTH_KE
    model.add(LSTM(128, return_sequences=True, activation='relu'))
    model.add(LSTM(64, return_sequences=False, activation='relu'))
    model.add(Dense(64, activation='relu'))
    #model.add(Dense(64, activation='relu'))
    #model.add(Dense(32, activation='relu'))
    model.add(Dense(32, activation='relu'))
    model.add(Dense(output_lenght, activation='softmax'))
    #model.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])

    #We add parameters to optimize the model
    #During training have a self-assessment, which is optimized with Adam, and the metric will be accuracy
    optimization = keras.optimizers.Adam(learning_rate= 0.00001)
    model.compile(loss='binary_crossentropy', optimizer=optimization, metrics=['accuracy'])

    return model
```

**Fig. 6.** LSTM neural network model architecture

## 3 RESULTS

### 3.1 Mobile application interface

The mobile application interface developed for continuous recognition and translation of SL has been meticulously designed to offer an intuitive, accessible, and efficient user experience. The prototype design of the mobile application was developed using the Figma platform, a visual design and prototyping tool, as referenced in [41]. Figure 7a presents the home interface of the SL Real-Time Continuous Translator application, which comprises three main components: "User Guide," "Dataset," and "Record & Translate." Each element is designed to enhance user experience when navigating the application.

Figure 7b emphasizes the importance of reviewing the list of basic expressions the LSTM model can recognize and translate in real-time. This step is crucial before proceeding to record and translate dynamic SL signs.

The "Dataset" section (see Figure 7c) provides video representations of these everyday expressions, including words and phrases, through interactive videos. This functionality facilitates learning and encourages user interaction with the technology, allowing them to practice and refine their SL skills. This is especially relevant for promoting effective communication between hearing and non-hearing individuals.

Finally, the "Record and Translate" section (see Figure 7d) offers the option to record dynamic signs, display their translation in text format, and play it back in audio. This feature aims to facilitate communication between hearing individuals and those with hearing disabilities, thus promoting an inclusive and accessible environment.

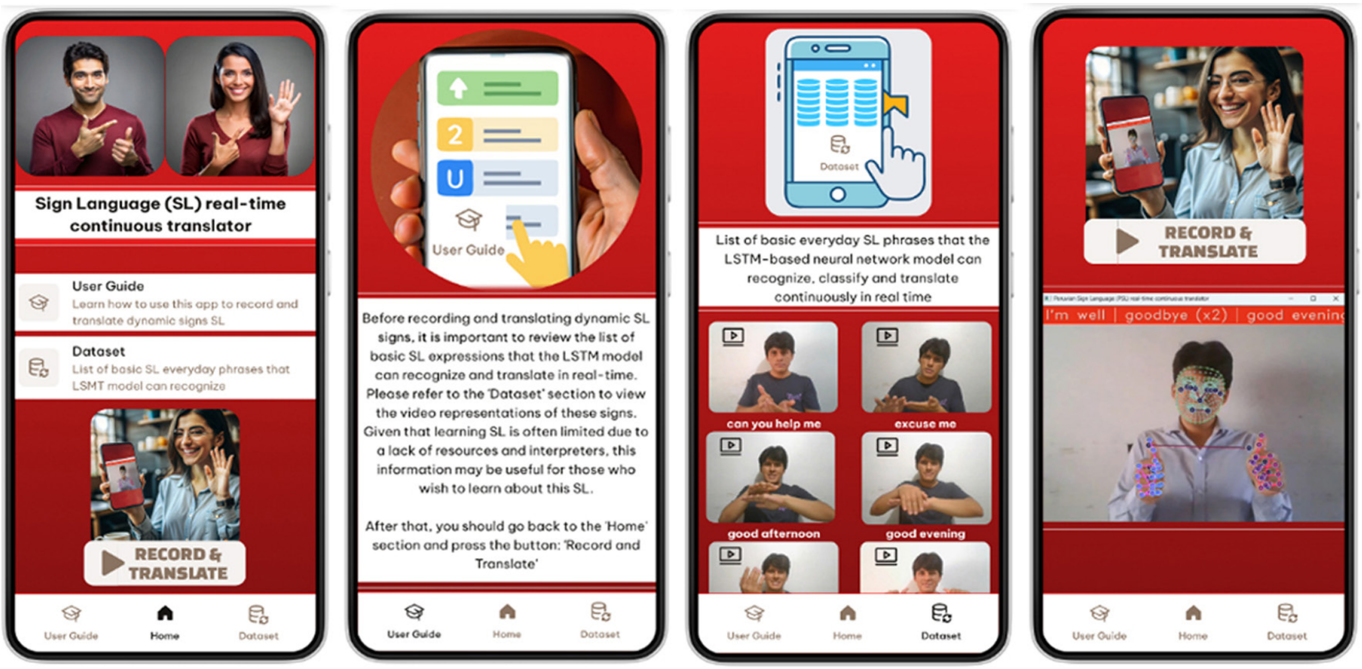


Fig. 7. Mobile application: (a) Home (b) User Guide (c) Dataset (d) Record and translate

## 3.2 Evaluations

**Evaluating model performance.** Adjustments to the previously mentioned hyperparameters, such as learning rate, number of layers, number of neurons, and epochs, were carried out through a trial-and-error process to obtain the best metrics in the test set, as seen in Table 3.

**Table 3.** Configuration of the hyperparameters of our neural network model based on LSTM architecture for evaluation of training and testing

| Epochs | Test Size | Learning Rate | Training | Testing |
|---|---|---|---|---|
| 100 | 0.1 | 0.00001 | 45360 frames (90%) | 5040 frames (10%) |

In this way, it was possible to optimize the performance of the neural network model based on the LSTM architecture, obtaining outstanding results in terms of accuracy and losses during the training and testing (validation) process. Thus, an accuracy of 99.80% was obtained for training with losses of 0.31%, and for testing (validation), an accuracy of 99.40% with losses of 0.58%, as shown in Table 4.

**Table 4.** Training and testing (validation) results

| Metric | Training | Testing (Validation) |
|---|---|---|
| Loss | 0.0017 | 0.0031 |
| Accuracy | 0.9997 | 0.9970 |

Consequently, the values obtained during training and testing the neural network are analyzed graphically, as shown in Figure 8. These analyses reveal that the LSTM model can effectively generalize to a test set containing previously unseen data, thus avoiding overfitting. As a result, the model achieves accurate real-time recognition when applied to the test set, demonstrating its ability to adapt to new data and make highly accurate predictions.

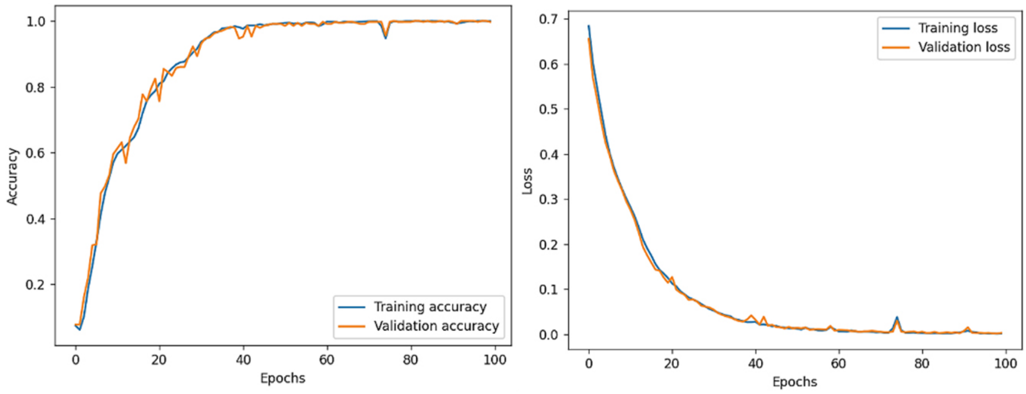


**Fig. 8.** Training and validation (test set) losses and accuracy

Our multiclass data set used a confusion matrix to perform a more detailed analysis of the results obtained. Figure 9 shows this matrix, where each row represents the actual class, while each column represents the class predicted by the model.

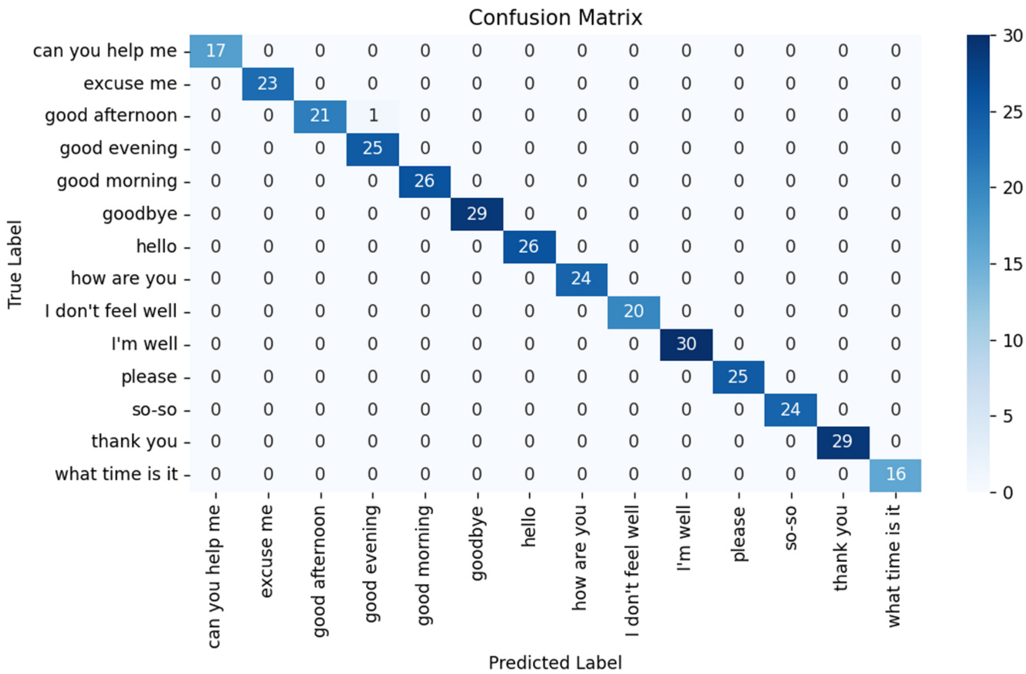


**Fig. 9.** LSTM model validation using the confusion matrix

The confusion matrix is crucial for a comprehensive and quantitative performance evaluation of a classification model. It facilitates the calculation of metrics such as precision, recall, and F1-score for each class. These metrics offer valuable insights into the model's capacity to accurately classify instances of each class and, importantly, to prevent confusion between different classes, thereby ensuring the model's accuracy. As shown in Table 5, according to the classification report, our LSTM model has achieved excellent precision, recall, and F1-score for the 11 samples: "goodbye," "good night," "good afternoon," "good morning," "how are you," "I am fine," "I am not well," "hello," "so-so," and "please." These values of 1.00 indicate perfect classification performance for these samples, meaning the model has been able to predict the cases of these 11 samples in the dataset with complete accuracy. Overall, the model demonstrates solid performance in data classification, as evidenced by the high precision, recall, and F1-score values for each class and the model's overall performance. This enables the real-time recognition, classification, and translation of dynamic sign sequences representing 14 basic SL expressions.

**Table 5.** Matrix confusion–classification report

| | Precision | Recall | F1-Score | Support |
|---|---|---|---|---|
| can you help me | 1.00 | 1.00 | 1.00 | 17 |
| excuse me | 1.00 | 1.00 | 1.00 | 23 |
| good afternoon | 1.00 | 0.95 | 0.98 | 22 |
| good evening | 0.96 | 1.00 | 0.98 | 25 |
| good morning | 1.00 | 1.00 | 1.00 | 26 |
| Goodbye | 1.00 | 1.00 | 1.00 | 29 |
| Hello | 1.00 | 1.00 | 1.00 | 26 |
| how are you | 1.00 | 1.00 | 1.00 | 24 |
| I don't feel well | 1.00 | 1.00 | 1.00 | 20 |
| I'm well | 1.00 | 1.00 | 1.00 | 30 |
| Please | 1.00 | 1.00 | 1.00 | 25 |
| so-so | 1.00 | 1.00 | 1.00 | 24 |
| thank you | 1.00 | 1.00 | 1.00 | 29 |
| what time is it | 1.00 | 1.00 | 1.00 | 16 |
| | | | | |
| Accuracy | | | 1.00 | 336 |
| macro avg | 1.00 | 1.00 | 1.00 | 336 |
| weighted avg | 1.00 | 1.00 | 1.00 | 336 |

**Expert evaluation.** The acceptance evaluation of the mobile application for the continuous recognition and translation of SL was carried out using a questionnaire covering 11 essential aspects of the system. The experts rated each aspect on a scale of 1 to 5, where 1 indicates a poor evaluation and 5 indicates excellent quality. Aspects evaluated included the accuracy of sign detection, the efficiency of real-time processing, and the usability of the user interface.

As detailed in Table 6, each of the 11 aspects received an individual rating from the experts. It should be noted that the evaluation of the results was carried out using SPSS Statistics software. The overall average obtained was an impressive 4.18, with a standard deviation of 0.49. This average significantly exceeds the acceptance threshold of 4, indicating that the mobile application is considered acceptable by the experts in general. The relatively low standard deviation suggests that the experts' evaluations were consistent and convergent. Overall, these results indicate that the mobile application not only meets the minimum required standards but has also been well received in terms of its technical and functional aspects.

**Table 6.** Expert evaluation

| Items | Evaluation of Technical Aspects | Experts | | | | Average | Standard Deviation |
|---|---|---|---|---|---|---|---|
| | | 1 | 2 | 3 | 4 | | |
| 1 | User interface design | 4 | 4 | 4 | 5 | 4.25 | .43 |
| 2 | Functionality | 3 | 5 | 3 | 5 | 4.00 | .00 |
| 3 | Performance | 4 | 4 | 4 | 5 | 4.25 | .43 |
| 4 | Accuracy of recognition | 5 | 5 | 5 | 3 | 4.50 | .87 |
| 5 | Real-time processing | 5 | 5 | 5 | 3 | 4.50 | .87 |
| 6 | Error handling | 5 | 4 | 4 | 4 | 4.25 | .43 |
| 7 | Ease of navigation | 5 | 4 | 5 | 5 | 4.75 | .43 |
| 8 | Adaptability to different environments | 5 | 4 | 5 | 5 | 4.75 | .43 |
| 9 | User feedback and interaction | 2 | 3 | 4 | 4 | 3.25 | .83 |
| 10 | Compatibility | 2 | 3 | 4 | 4 | 3.25 | .83 |
| 11 | Overall usability | 5 | 5 | 4 | 3 | 4.25 | .83 |
| | TOTAL | | | | | 4.18 | .49 |

## 4 DISCUSSION

This study addresses a critical need to recognize SL by developing a mobile application that uses advanced DL techniques to recognize and classify images continuously. Historical research has focused on the recognition of isolated signs and the use of static databases limited to aspects such as the alphabet and numbers [27], [28], [30], [31], which has restricted the capability of existing systems to handle dynamic sequences and broader contexts. To overcome these limitations, this study has developed a continuous database covering a more comprehensive range of SL expressions, including nouns and everyday phrases, consisting of 3360 videos and 50,400 images, providing a more prosperous and more diverse representation compared to previous databases [25], [26], [29].

The LSTM-based neural network model, developed in Python, has been crucial for the mobile application, allowing real-time processing of dynamic sign sequences and facilitating accurate translation of SL expressions. This model has demonstrated outstanding performance with an accuracy of 99.80% during training and 99.40% during validation, surpassing the accuracies reported in previous studies [17], [26], [39], [40], where the best results reached 95.05% and 95.43%. Furthermore, the precision and loss graphs indicate that the LSTM model has avoided overfitting, demonstrating its ability to effectively generalize across the 14 dynamic SL signs in the database. This contrasts with overfitting issues faced in other studies [40], highlighting the robustness and effectiveness of the proposed approach, which has achieved a precision, recall, and F1-score of 99%, compared to the 70% obtained in [26].

## 5 CONCLUSION

The successful development of a Python-based AI algorithm has led to the creation of an innovative mobile application for the continuous recognition and classification of SL. This advancement addresses the gap in SL databases, which traditionally focus on isolated signs and limited vocabularies. Utilizing the mobile device's webcam, the algorithm accurately captures and classifies frames of dynamic signs, facilitating real-time communication.

The created database, which includes 240 image sequences for each of the 14 basic SL expressions, consists of 3360 videos and 50,400 frames. This continuous and balanced database represents a significant step towards a more comprehensive and practical recognition of SL, surpassing the limitations of previous databases that focused solely on nouns. The neural network model based on the LSTM architecture, implemented using a DL algorithm in Python, has proven to be highly effective in this context. The model achieved an accuracy of 99.80% during training and 99.40% during testing, with losses of 0.31% and 0.58%, respectively. These metrics indicate that the model can recognize and classify basic SL expressions in real-time, avoid overfitting, and generalize effectively.

A limitation of the proposed approach is the dependence on the video capture quality, as the LSTM model may be affected in environments with low resolution or poor illumination. To mitigate this limitation, the dataset was diversified by including images taken in different lighting conditions and with varied backgrounds. In addition, normalization and contrast adjustment techniques were applied during image preprocessing, which ensured that the model could better adapt to different recording conditions and maintain high accuracy in real situations, thus increasing its robustness and practical applicability.

In future research, exploring the integration of advanced computer vision and machine learning techniques, such as convolutional neural networks (CNN) for feature extraction combined with LSTM for temporal analysis, would be valuable. This could further improve the accuracy and robustness of the system under conditions of high variability. In addition, the expansion of the database to include a more extensive and more varied vocabulary, including more complex contexts and phrases, would be crucial to improve the system's ability to handle a broader range of utterances and communicative situations, as would the integration of additional motion sensors to enrich sign interpretation and improve recognition accuracy in diverse environments.

In conclusion, the integration of this model into the mobile application is a significant step towards promoting inclusive communication in digital environments. It enables a smooth and accessible human-computer interaction, particularly for individuals with hearing impairments. This advancement significantly contributes to the evolution of SLR and translation in the realm of DL, marking a crucial milestone in the field.

## 6 AUTHORS' CONTRIBUTIONS

The authors' contributions to this study is significant: Angel Diego Briones Cerquín led the development of the LSTM-based neural network model and the mobile application, ensuring the effective implementation of the algorithm and the integration with the user interface. Johan Alonso Tumay Guevara was responsible for data collection and preprocessing, creating a balanced and diversified database that improved the model's ability to handle different capture conditions. Christian Ovalle contributed to optimizing the model's performance and the design of the DL architecture, as well as validating the system's accuracy and robustness in real time. It should also be noted that all of these contributions have contributed to the development of this study, which has allowed the development of an innovative tool that facilitates inclusive communication in digital environments and lays the groundwork for future improvements in SL recognition.

## 7 REFERENCES


[1] M. Cabanillas-Carbonell, P. Cusi-Ruiz, D. Prudencio-Galvez, and J. L. Herrera Salazar, "Mobile application with augmented reality to improve the process of learning sign language," *Int. J. Interact. Mob. Technol. (iJIM)*, vol. 16, no. 11, pp. 51–64, 2022. https://doi.org/10.3991/ijim.v16i11.29717

[2] R. Dermawi, H. Tolle, and I. Aknuranda, "Design and usability evaluation of communication board for deaf people with user-centered design approach," *Int. J. Interact. Mob. Technol. (iJIM)*, vol. 12, no. 2, pp. 197–206, 2018. https://doi.org/10.3991/ijim.v12i2.8100

[3] R. M. Kagalkar and S. V. Gumaste, "Mobile application based translation of sign language to text description in Kannada language," *Int. J. Interact. Mob. Technol. (iJIM)*, vol. 12, no. 2, pp. 92–112, 2018. https://doi.org/10.3991/ijim.v12i2.8071

[4] Defensoría del Pueblo, "Defensoría del Pueblo: el Estado debe respetar la lengua materna de las personas sordas," *Defensoría del Pueblo*, Feb. 21, 2022. https://www.defensoria.gob.pe/defensoria-del-pueblo-el-estado-debe-respetar-la-lenguamaterna-de-las-personas-sordas/ [Accessed: Jun. 23, 2023].

[5] E. R. Huiza Pereyra and C. A. Olivares Poggi, "Talking with signs a simple method to detect nouns and numbers in a non-annotated signs language corpus," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2021, pp. 1212–1220. https://doi.org/10.1109/CVPRW53098.2021.00132

[6] Defensoría del Pueblo, "Defensoría del Pueblo: debe facilitarse el aprendizaje de la lengua de señas peruana y promover la identidad lingüística y cultural de las personas sordas," *Defensoría del Pueblo*, Sep. 24, 2020. https://www.defensoria.gob.pe/defensoria-del-pueblo-debe-facilitarse-el-aprendizaje-de-la-lengua-de-senas-peruana-y-promover-la-identidad-linguistica-y-cultural-de-las-personas-sordas/ [Accessed: Jun. 23, 2023].

[7] G. P. G. Ramos and M. E. F. Choquehuanca, "Translator system for peruvian sign language texts through 3D virtual assistant," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 11, no. 1, 2020. https://doi.org/10.14569/IJACSA.2020.0110168

[8] G. F. Barrientos-Villalta, P. Quiroz, and W. Ugarte, "Peruvian sign language recognition using recurrent neural networks," in *Advanced Research in Technologies, Information, Innovation and Sustainability, Communications in Computer and Information Science*, T. Guarda, F. Portela, and M. F. Augusto, Eds., Springer, Cham, vol. 1675, 2022, pp. 459–473. https://doi.org/10.1007/978-3-031-20319-0_34

[9] R. Rastgoo, K. Kiani, S. Escalera, V. Athitsos, and M. Sabokrou, "A survey on recent advances in sign language production," *Expert Syst. Appl.*, vol. 243, p. 122846, 2024. https://doi.org/10.1016/j.eswa.2023.122846

[10] O. Koller, J. Forster, and H. Ney, "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers," *Computer Vision and Image Understanding*, vol. 141, pp. 108–125, 2015. https://doi.org/10.1016/j.cviu.2015.09.013

[11] S. Siddique, S. Islam, E. E. Neon, T. Sabbir, I. T. Naheen, and R. Khan, "Deep learning-based bangla sign language detection with an edge device," *Intell. Syst. with Appl.*, vol. 18, p. 200224, 2023. https://doi.org/10.1016/j.iswa.2023.200224

[12] Y. Saleh and G. F. Issa, "Arabic sign language recognition through deep neural networks fine-tuning," *Int. J. Online Biomed. Eng. (iJOE)*, vol. 16, no. 5, pp. 71–83, 2020. https://doi.org/10.39ic91/ijoe.v16i05.13087

[13] M. De Coster, D. Shterionov, M. Van Herreweghe, and J. Dambre, "Machine translation from signed to spoken languages: State of the art and challenges," *Univers. Access. Inf. Soc.*, vol. 23, pp. 1305–1331, 2024. https://doi.org/10.1007/s10209-023-00992-1

[14] A. Núñez-Marcos, O. Perez-de-Viñaspre, and G. Labaka, "A survey on Sign Language machine translation," *Expert Syst. Appl.*, vol. 213, p. 118993, 2023. https://doi.org/10.1016/j.eswa.2022.118993

[15] H. Ansar *et al.*, "Hand gesture recognition for characters understanding using convex hull landmarks and geometric features," *IEEE Access*, vol. 11, pp. 82065–82078, 2023. https://doi.org/10.1109/ACCESS.2023.3300712

[16] R. Jain, R. K. Karsh, and A. A. Barbhuiya, "Literature review of vision-based dynamic gesture recognition using deep learning techniques," *Concurr. Comput. Pract. Exp.*, vol. 34, no. 22, p. e7159, 2022. https://doi.org/10.1002/cpe.7159

[17] H. Bull, M. Gouiffès, and A. Braffort, "Automatic segmentation of sign language into subtitle-units," in *Computer Vision–ECCV 2020 Workshops, ECCV 2020,* in Lecture Notes in Computer Science, A. Bartoli, and A. Fusiello, Eds., Springer, Cham, vol. 12536, 2020, pp. 186–198. https://doi.org/10.1007/978-3-030-66096-3_14

[18] A. G. Mahmoud, A. M. Hasan, and N. M. Hassan, "Convolutional neural networks framework for human hand gesture recognition," *Bulletin of Electrical Engineering and Informatics*, vol. 10, no. 4, pp. 2223–2230, 2021. https://doi.org/10.11591/eei.v10i4.2926

[19] B. Y. AlKhuraym, M. M. Ben Ismail, and O. Bchir, "Arabic sign language recognition using lightweight CNN-based architecture," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 13, no. 4, pp. 319–328, 2022. https://doi.org/10.14569/IJACSA.2022.0130438

[20] S. Dwijayanti, Hermawati, S. I. Taqiyyah, H. Hikmarika, and B. Y. Suprapto, "Indonesia sign language recognition using convolutional neural network," *International Journal of Advanced Computer Science and Applications (IJACSA)*, vol. 12, no. 10, pp. 415–422, 2021. https://doi.org/10.14569/IJACSA.2021.0121046

[21] H. Zahid *et al.*, "A computer vision-based system for recognition and classification of Urdu sign language dataset," *PeerJ Comput. Sci.*, vol. 8, p. e1174, 2022. https://doi.org/10.7717/peerj-cs.1174

[22] World Health Organization, "World report on hearing," 2021. https://www.who.int/publications/i/item/9789240020481

[23] H. Luqman, E. S. M. El-Alfy, and G. M. BinMakhashen, "Joint space representation and recognition of sign language fingerspelling using Gabor filter and convolutional neural network," *Multimed. Tools. Appl.*, vol. 80, pp. 10213–10234, 2021. https://doi.org/10.1007/s11042-020-09994-0

[24] A. A. I. Sidig, H. Luqman, and S. A. Mahmoud, "Transform-based Arabic sign language recognition," *Procedia Comput. Sci.*, vol. 117, pp. 2–9, 2017. https://doi.org/10.1016/j.procs.2017.10.087

[25] L. K. S. Tolentino, R. O. Serfa Juan, A. C. Thio-Ac, M. A. B. Pamahoy, J. R. R. Forteza, and X. J. O. Garcia, "Static sign language recognition using deep learning," *Int. J. Mach. Learn. Comput. (IJMLC)*, vol. 9, no. 6, pp. 821–827, 2019. https://doi.org/10.18178/ijmlc.2019.9.6.879

[26] B. A. Al-Mohimeed, H. O. Al-Harbi, G. S. Al-Dubayan, and A. A. Al-Shargabi, "Dynamic sign language recognition based on real-time videos," *Int. J. online Biomed. Eng. (iJOE)*, vol. 18, no. 1, pp. 4–14, 2022. https://doi.org/10.3991/ijoe.v18i01.27581

[27] C. Lazo, Z. Sanchez, and C. del Carpio, "A static hand gesture recognition for peruvian sign language using digital image processing and deep learning," in *Proceedings of the 4th Brazilian Technology Symposium (BTSym'18), Smart Innovation, Systems and Technologies*, Y. Iano, R. Arthur, O. Saotome, V. Vieira Estrela, and H. Loschi, Eds., Springer, Cham, vol. 140, 2019, pp. 281–290. https://doi.org/10.1007/978-3-030-16053-1_27

[28] S. B. Abdullahi, K. Chamnongthai, V. Bolon-Canedo, and B. Cancela, "Spatial–temporal feature-based End-to-end Fourier network for 3D sign language recognition," *Expert Syst. Appl.*, vol. 248, p. 123258, 2024. https://doi.org/10.1016/j.eswa.2024.123258

[29] Y. Cao *et al.*, "Explore more guidance: A task-aware instruction network for sign language translation enhanced with data augmentation," in *Findings of the Association for Computational Linguistics: NAACL 2022*, 2022, pp. 2679–2690.

[30] A. Haque, R. A. Pulok, M. M. Rahman, S. Akter, N. Khan, and S. Haque, "Recognition of Bangladeshi Sign Language (BdSL) words using Deep Convolutional Neural Networks (DCNNs)," *Emerg. Sci. J.*, vol. 7, no. 6, pp. 2183–2201, 2023. https://doi.org/10.28991/ESJ-2023-07-06-019

[31] R. K. Pathan, M. Biswas, S. Yasmin, M. U. Khandaker, M. Salman, and A. A. F. Youssef, "Sign language recognition using the fusion of image and hand landmarks through multi-headed convolutional neural network," *Sci. Rep.*, vol. 13, pp. 1–11, 2023. https://doi.org/10.1038/s41598-023-43852-x

[32] P. J. Ezigbo, O. C. Nosiri, E. S. Mbonu, V. Ofor, and J. Obichere, "Hand gesture recognition and control for human-robot interaction using deep learning," *Int. J. Electr. Electron. Eng. Telecommun.*, vol. 12, no. 6, pp. 433–441, 2023. https://doi.org/10.18178/ijeetc.12.6.433-441

[33] S. Narayan, A. P. Mazumdar, and S. K. Vipparthi, "SBI-DHGR: Skeleton-based intelligent dynamic hand gestures recognition," *Expert Syst. Appl.*, vol. 232, p. 120735, 2023. https://doi.org/10.1016/j.eswa.2023.120735

[34] H. Mohammad, H. Tamimi, and F. Abu-Amara, "An educational arabic sign language mobile application for children with hearing impairment," *Int. J. Interact. Mob. Technol. (iJIM)*, vol. 16, no. 20, pp. 114–129, 2022. https://doi.org/10.3991/ijim.v16i20.32427

[35] S. Abou El-Seoud, I. Taj-Eddin, A. Nosseir, H. El-Sofany, and N. Abu Rumman, "A proposed pedagogical mobile application for learning sign language," *Int. J. Interact. Mob. Technol. (iJIM)*, vol. 7, no. 1, pp. 46–55, 2013. https://doi.org/10.3991/ijim.v7i1.2387

[36] Z. Katılmış and C. Karakuzu, "Double handed dynamic turkish sign language recognition using leap motion with meta learning approach," *Expert Syst. Appl.*, vol. 228, p. 120453, 2023. https://doi.org/10.1016/j.eswa.2023.120453

[37] A. Y. Saleh and L. H. Chern, "Autism spectrum disorder classification using deep learning," *Int. J. online Biomed. Eng. (iJOE)*, vol. 17, no. 8, pp. 103–114, 2021. https://doi.org/10.3991/ijoe.v17i08.24603

[38] M. Abduljabbar Ali, A. Jaafar Hussain, and A. T. Sadiq, "Deep learning algorithms for human fighting action recognition," *Int. J. Online Biomed. Eng. (iJOE)*, vol. 18, no. 2, pp. 71–87, 2022. https://doi.org/10.3991/ijoe.v18i02.28019

[39] J. Wu, P. Ren, B. Song, R. Zhang, C. Zhao, and X. Zhang, "Data glove-based gesture recognition using CNN-BiLSTM model with attention mechanism," *PLoS One*, vol. 18, no. 11, p. e0294174, 2023. https://doi.org/10.1371/journal.pone.0294174

[40] A. M. Buttar, U. Ahmad, A. H. Gumaei, A. Assiri, M. A. Akbar, and B. F. Alkhamees, "Deep learning in sign language recognition: A hybrid approach for the recognition of static and dynamic signs," *Mathematics*, vol. 11, no. 17, pp. 1–20, 2023. https://doi.org/10.3390/math11173729

[41] R. Jáuregui-Velarde, P. Molina-Velarde, C. Yactayo-Arias, and L. Andrade-Arenas, "Evaluation of a prototype mobile application based on expert system for the diagnosis of diseases transmitted by the aedes aegypti mosquito," *Int. J. online Biomed. Eng. (iJOE)*, vol. 19, no. 13, pp. 72–91, 2023. https://doi.org/10.3991/ijoe.v19i13.41431

## 8 AUTHORS

**Angel Diego Briones Cerquín** is a graduate from the Universidad Tecnológica del Perú, Faculty of Electronics Engineering, belonging to the tenth grade, with Peruvian nationality in the department of Lima, residing in Jr. Neptuno 564 – Cercado de Lima (E-mail: U18205516@utp.edu.pe).

**Johan Alonso Tumay Guevara** is a graduate from the Universidad Tecnológica del Perú, Faculty of Electronics Engineering, belonging to the tenth grade, with Peruvian nationality in the department of Lima, residing in Ca. Los Canchis 140 – Chorrillos (E-mail: U17101665@utp.edu.pe).

**Christian Ovalle** is a Systems Engineer graduated from the Universidad Peruana Union belonging to the top third, master's in management and management of information technology, with a Diploma in Process Management and Specialty in operations and Logistics from ESAN University (E-mail: c19349@utp.edu.pe).