technologies

Article

Towards a Bidirectional Mexican Sign Language–Spanish Translation System: A Deep Learning Approach

Jaime-Rodrigo González-Rodríguez [1], Diana-Margarita Córdova-Esparza [1,*], Juan Terven [2] and Julio-Alejandro Romero-González [1]

[1] Facultad de Informática, Universidad Autónoma de Querétaro, Campus Juriquilla, Av. de las Ciencias S/N, Juriquilla C.P. 76230, Querétaro, Mexico; jgonzalez.rdz28@gmail.com (J.-R.G.-R.); julio.romero@uaq.mx (J.-A.R.-G.)
[2] Instituto Politécnico Nacional, CICATA—Unidad Querétaro, Cerro Blanco No. 141, Col, Colinas del Cimatario C.P. 76090, Querétaro, Mexico; jrtervens@ipn.mx
* Correspondence: diana.cordova@uaq.mx

Abstract: People with hearing disabilities often face communication barriers when interacting with hearing individuals. To address this issue, this paper proposes a bidirectional Sign Language Translation System that aims to bridge the communication gap. Deep learning models such as recurrent neural networks (RNN), bidirectional RNN (BRNN), LSTM, GRU, and Transformers are compared to find the most accurate model for sign language recognition and translation. Keypoint detection using MediaPipe is employed to track and understand sign language gestures. The system features a user-friendly graphical interface with modes for translating between Mexican Sign Language (MSL) and Spanish in both directions. Users can input signs or text and obtain corresponding translations. Performance evaluation demonstrates high accuracy, with the BRNN model achieving 98.8% accuracy. The research emphasizes the importance of hand features in sign language recognition. Future developments could focus on enhancing accessibility and expanding the system to support other sign languages. This Sign Language Translation System offers a promising solution to improve communication accessibility and foster inclusivity for individuals with hearing disabilities.

Keywords: Mexican sign language; translation; machine learning; recurrent networks; assistive technologies

Citation: González-Rodríguez, J.-R.; Córdova-Esparza, D.-M.; Terven, J.; Romero-González, J.-A. Towards a Bidirectional Mexican Sign Language–Spanish Translation System: A Deep Learning Approach. *Technologies* **2024**, *12*, 7. https://doi.org/10.3390/technologies12010007

Academic Editor: Luc de Witte

Received: 27 November 2023
Revised: 28 December 2023
Accepted: 3 January 2024
Published: 5 January 2024

1. Introduction

Deaf communities globally encounter significant challenges in accessing vital services like education, healthcare, and employment due to language barriers, rather than auditory limitations [1]. Their primary language is often a signed language, such as American, French, German, or Greek Sign Language, each a unique and complete language, distinct from spoken languages and each other. These languages, with over two hundred identified varieties, possess the same depth and expressive power as spoken languages [2,3]. However, for the Deaf, any spoken language is secondary, leading to low literacy rates; for instance, in the U.S., deaf high school graduates have an average reading level of third to fourth grade [4]. This language gap not only hinders everyday interactions with the hearing, non-signing population but also affects access to critical services. While certified sign language interpreters are the best solution for essential services, their scarcity and cost render them impractical for everyday, brief interactions. Thus, the development of effective automatic translation systems for spoken and signed languages could significantly improve communication and inclusivity for the Deaf community.

To overcome this barrier, technological solutions have been developed. Translator gloves [5–7], mobile applications, and automatic translators are the leading technologies that have been used for unidirectional or bidirectional communication.

Our research aims to develop a bi-directional sign language translator that can translate from Spanish to Mexican Sign Language (MSL) and vice versa, bridging the gap between these two languages. The system involves two operation modes: From Mexican Sign Language to Spanish (MSL-SPA) and from Spanish to Mexican Sign Language (SPA-MSL). In the MSL-SPA mode, the system captures live video and processes it to recognize the sign and translate it to Spanish, as shown in the upper path in Figure 1. Conversely, in the SPA-MSL mode, the user types the phrase and the system displays a sign language animation, as shown in the lower path in Figure 1.

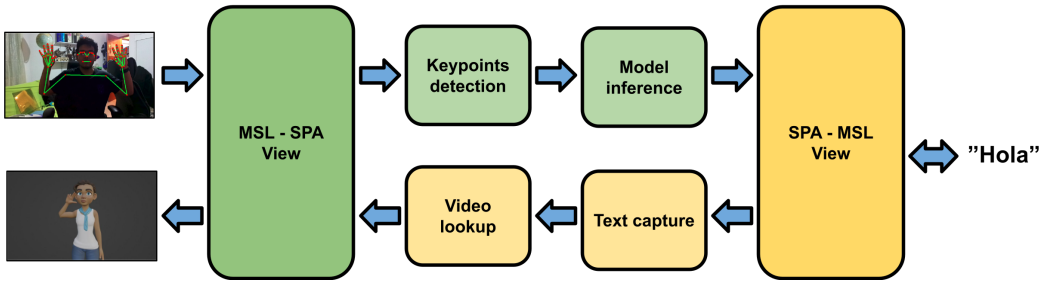


Figure 1. General pipeline of the bidirectional sign language translation system. In the MSL-SPA mode, the system recognizes the sign from live video and displays the text in Spanish on the other end. Conversely, in the SPA-MSL mode, the user types the phrase and the system displays a sign language animation on the other end.

Our system is based on deep learning techniques, which have shown great success in various computer vision and natural language processing tasks. Specifically, we used MediaPipe for keypoint detection, which is an advanced, real-time framework that utilizes machine learning to detect and track keypoints on objects, faces, hands, or poses in images and videos. We also used recurrent networks such as RNN, BRNN, LSTM, and GRU, as well as an encoder-only transformer for the translation process, which we treated as a time-series classification.

One of the main challenges in developing a bi-directional sign language translator is the variability and complexity of sign language gestures, as well as the need to capture the nuances and context of the conversation. Another challenge is the lack of large and diverse sign language datasets, which are crucial for training accurate models. To address these challenges, we collected a new dataset consisting of gestures from MSL, which we used to train and evaluate our system.

The proposed bi-directional sign language translator has the potential to significantly improve the communication and integration of the deaf community into society by allowing them to communicate more effectively with hearing people. Moreover, it can facilitate the learning of sign language for hearing people and promote a more inclusive and diverse society.

To provide an overview of the paper, we have organized it in the following manner: In Section 2, we summarize the relevant literature, while Section 3 outlines the methodology we employed in our project. Section 4 will showcase the results we obtained, and finally, in Section 5, we present our concluding thoughts.

2. Related Work

The landscape of sign language translation and recognition research is rich and varied, marked by a series of interconnected advancements that build upon each other. This section weaves through these developments, highlighting how each contribution sets the stage for the next.

Starting with Bungeroth & Ney [8], we see the foundations being laid with a German Sign Language (DGS) translation system. This innovative approach, integrating audio feedback and animated representation, utilizes IBM Model 1-4 and Hidden Markov Models

(HMM) for training. The challenge they faced due to limited training samples echoes the necessity for a robust corpus, as further exemplified by the notation method of [9].

Building on the concept of practical translation, San-Segundo et al. [10] introduced a real-time method for Spanish-to-sign language translation. Their dual approach, blending rule-based and statistical methods, demonstrated adaptability and precision, particularly in contexts with limited vocabulary.

Pichardo-Lagunas et al. [11] continued this trajectory, focusing on Mexican Sign Language (MSL). They brought a meticulous, analytical lens to Spanish text, using Freeling to classify words for accurate translation. This method, though currently limited to one-way translation, reflects the evolving complexity of sign language translation systems.

Segueing to pose detection and classification, Qiao et al. [12] utilized the OpenPose model, demonstrating a significant leap in motion analysis without the dependency on specialized hardware. This development represents a shift towards more accessible and cost-effective solutions in the field.

Barrera-Melchor et al. [13] then added a new dimension by applying these technologies to educational content translation into MSL. Their cloud-based system, which translates speech to text and then to MSL using a 3D avatar, exemplifies the integration of cloud computing in sign language translation.

In a similar vein, focusing on a specific application area, Sosa-Jimenez et al. [14] developed a research prototype tailored for primary care health services in Mexican Sign Language. Their use of Microsoft Kinect sensors [15] and HMMs highlights the trend of specialized systems addressing distinct contexts like healthcare.

Parallel to these developments, Martínez-Gutiérrez et al. [16] and Martinez-Seis et al. [17] focused on MSL alphabet recognition through advanced computational methods, each achieving notable accuracy in their respective areas.

Carmona et al. [18] introduced a system for recognizing the static alphabet in Mexican Sign Language using Leap Motion and MS Kinect 1 sensors. Their unique application of 3D affine moment invariants for sign recognition demonstrated a significant improvement in accuracy, showcasing the potential of 3D modeling in sign language recognition.

Naranjo et al. [19] attempt to expand the field, developing a graphical tool to aid in learning Costa Rican Sign Language (LESCO). Their methodology, utilizing phonological parameters and a similarity formula, provides a bridge for learners to grasp the nuances of sign languages, emphasizing the role of educational tools in sign language dissemination.

Complementing these efforts, Trujillo et al. [20] presented a translation system from Mexican Sign Language to spoken language, employing 3D hand movement trajectories. Their approach to refining movement patterns and using advanced algorithms like KNN highlights the continuous push for higher precision and efficiency in translation systems.

In a similar spirit of refinement, Jimenez et al. [21], and Cervantes et al. [22] each contributed distinct methodologies for sign language recognition, whether through 3D affine invariants or sophisticated video analysis. These studies underscore the diverse technological avenues being explored to enhance sign language translation and recognition accuracy.

With the advent of the Transformer as a powerful deep learning model for translation, it has been used to improve the accuracy of sign language translation by effectively extracting joint visual-text features and capturing contextual information [23]. One approach is to design an efficient transformer-based deep network architecture that exploits multi-level spatial and temporal contextual information, such as the proposed heterogeneous attention-based transformer (HAT) model [24]. Another approach is to address the local temporal relations and non-local and global context modeling in sign videos, using techniques like the multi-stride position encoding scheme and the adaptive temporal interaction module [25]. Additionally, transfer learning with pretrained language models, such as BERT, can be used to initialize sign language translation models and improve performance [26]. Furthermore, incorporating content-aware and position-aware convolution layers, as well as injecting relative position information to the attention mechanism, can enhance sign language understanding and improve translation quality [27].

Using avatars to translate sign language presents both challenges and benefits. One of the main challenges is the complexity of sign languages, which requires a deep understanding of their grammatical mechanisms and inflecting mechanisms [28]. Additionally, the lack of direct participation from the deaf community and the underestimation of sign language complexity have resulted in structural issues with signing avatar technologies [29]. However, the benefits of using avatars for sign language translation include increased accessibility for the deaf community and the potential for automation and efficiency in translating spoken or written language to sign language [30–32]. Avatars can also be used as educational tools and have the potential to improve the naturalness and believability of sign language motion either from text to animation [33], from animation to text [23,34] or both ways, as proposed in this work.

In recap, this collective body of work forms a tapestry of innovation, each research piece contributing to a greater understanding and capability in the field of sign language translation and recognition. Our research aims to add to this rich tapestry by developing a bi-directional translator between Spanish and Mexican Sign Language (MSL). Leveraging advanced techniques like MediaPipe and deep learning models, our goal is to bridge the communication gap for the deaf community. The Section 3 that follows will detail our unique approach, situating it within this dynamic and evolving research landscape.

3. Methods

This section describes the methodology pursued to develop the bidirectional translation system. The development stages were the following:

1. Hardware selection
2. Feature Selection
3. Data collection
4. Model definition
5. Graphical user interface

3.1. Hardware Selection

To select the computing board, we compared the Raspberry Pi 4 Model B [35], the Up Square [36], and the Nvidia Jetson Nano [37], running a benchmark to evaluate the inference speed of each of them.

For this, we run MediaPipe's Holistic model on each card. This model includes detecting points on the body, hands, and face, making it computationally expensive. The RaspberryPi 4 ran at four frames per second, the UpSquared ran at six frames per second and the Jetson Nano, being the most powerful due to its GPU, ran at 13 frames per second.

3.2. Feature Selection

The inference and translation of the model depend on the input of keypoints they receive, so it is necessary to define those features, or in this case, keypoints, that are statistically significant and contribute to the model's inference process, always seeking the balance between the number of features to process and computational cost.

To optimize the Jetson Nano's resources for keypoint coordinate detection, we reduced the number of features. This reduction freed up computational capacity for other tasks in our translation system. We conducted performance tests using MediaPipe's pose detection, hand detection, and holistic pipelines. The holistic pipeline was the most resource-intensive, leading us to combine pose and hand detection pipelines for greater efficiency. This combination created a lighter version than the holistic model by eliminating the dense facial keypoint mesh computation. Figure 2, shows the full face mesh containing 468 keypoints and the eleven keypoints we end using shown in blue. We chose this approach because the facial mesh keypoints added little value to our model's inference, particularly since the signs we needed to identify mainly involved arm movements and finger positions.

For the body, we reduced the body keypoints to five: four from the original BlazePose model and one midpoint between the shoulder keypoints for chest detection. This selection

was due to the movements in the signs being above the waist, making leg keypoints irrelevant for our model. For the hands, we kept all 21 keypoints because hand and finger positions are crucial for distinguishing between signs.

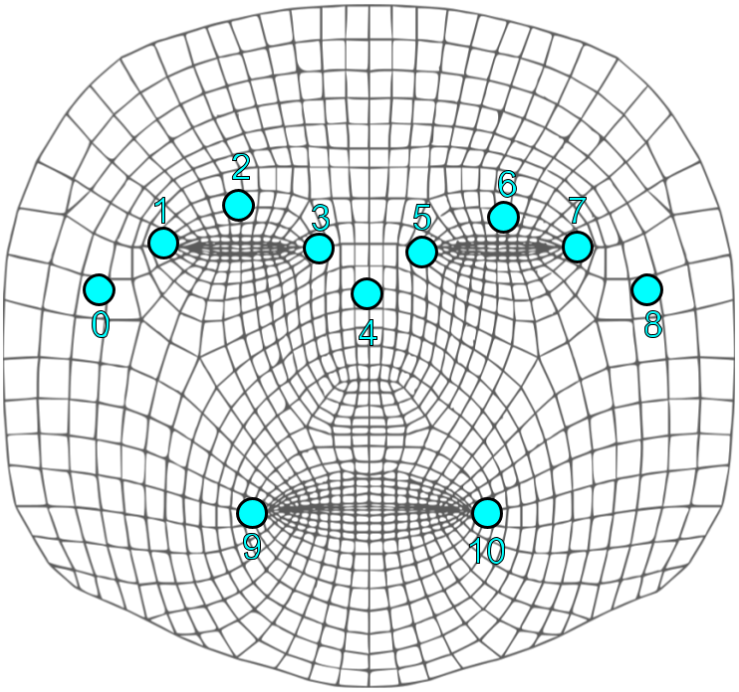


Figure 2. Mediapipe Face Mesh includes 468 3D face landmarks. To reduce computation, we used the blue keypoints obtained from the pose model instead.

Figure 3 displays the final topology of our translation system, comprising 58 keypoints. We calculated the X, Y, and Z coordinates for each, resulting in 174 features for processing.

By reducing the keypoints, we optimized the model's input layer, thereby decreasing its computational demands. This optimization made both the training and inference processes more efficient and reduced the data volume needed for training, validation, and testing splits of the model.

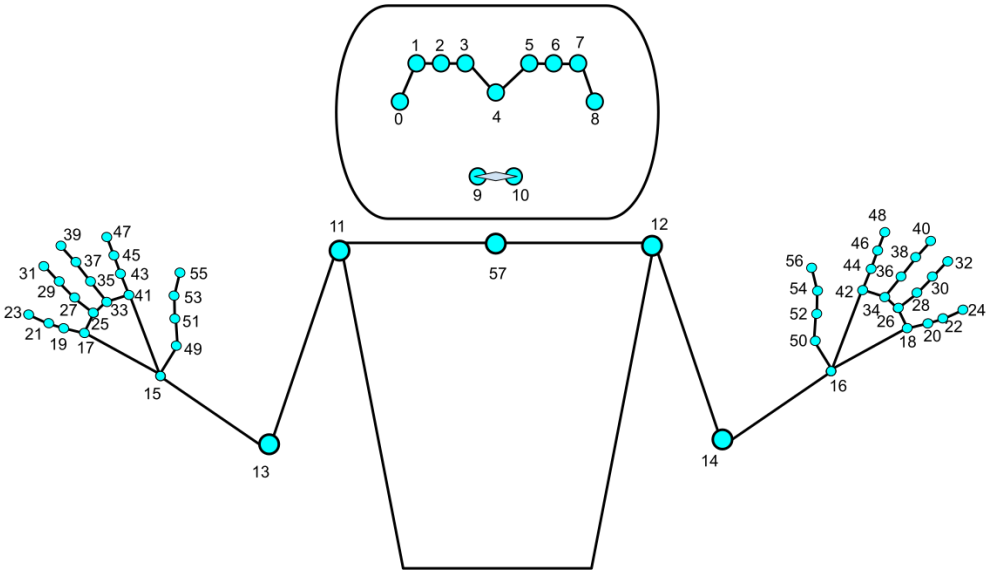


Figure 3. Final 58 Keypoints used for the sign-recognition system.

3.3. Data Collection

To make the system manageable, we chose a subset of ten signs, precisely phrases applicable in a school setting. The selected phrases are: "Hello", "Are there any questions?", "Help me", "Good morning", "Good afternoon", "Good night", "What is the homework?", "Is this correct?", "The class is over", and "Can you repeat it?". Approximately 1000 samples of each sign were collected from six individuals, comprising an equal gender split of three women and three men, with their ages ranging from 22 to 55.

For sample collection of these phrases, we developed a Python script that uses MediaPipe's keypoint detector to gather samples of each sign. We collected each sign from a distance of 2m containing 15 frames with detections. We found that in practice, all signs fit within this time period.

For each keypoint, we calculated the X, Y, and Z coordinates.

To compute the Z coordinate, we used the depth provided by the OAK-D camera [38]. The depth camera is composed of a stereo pair of OMNIVISION's OV9282 1MP grayscale image sensor [39]. The depth accuracy varies depending on the distance from the object being measured being more accurate at closer ranges. From 0.7 m to 4 m, the camera maintains an absolute depth error below 1.5 cm [40], which is sufficient for our application.

We used perspective projection to determine the distance relative to the camera for each keypoint of interest, as shown in Equation (1).

$$distance = (FocalLenth \times BaselineDistance)/XYpoint \quad (1)$$

To increase the variability of the samples, signs were collected from six different individuals, aiming to reduce sample bias. For each of the ten signs, we gathered approximately 900 samples on average, resulting in a total of around 9300 samples.

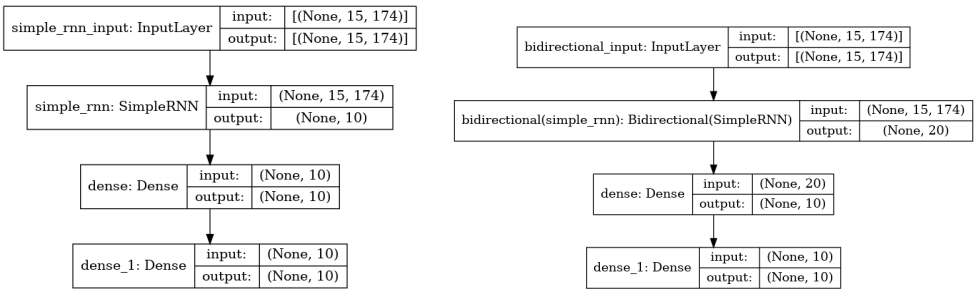
3.4. Model Definition

Given the specific challenges of our project, we chose to implement a Recurrent Neural Network (RNN) model within our translation system. RNNs are particularly effective for tasks like natural language processing, video analysis, and machine translation, mainly because of their ability to maintain a form of memory. This memory helps in understanding sequences, as it can track changes over time.

To find the most suitable RNN model for classifying signs in Mexican Sign Language (MSL), we evaluated various RNN architectures. The models we considered included the following:

- Standard RNN [41]: Ideal for handling sequences and time-series data.
- Long Short-Term Memory (LSTM) [42]: Similar to GRU but with a different gating mechanism, often used for more complex sequence data.
- Bidirectional RNN (BRNN) [43] and Bidirectional LSTM [44]: Enhances the standard recurrent networks by processing data in both forward and backward directions, offering a more comprehensive understanding of the sequence context.
- Gated Recurrent Unit (GRU) [45]: A more efficient version of the standard RNN, known for better performance on certain datasets.
- Transformer [46]: A newer model that has gained popularity in various sequence modeling tasks, known for handling long-range dependencies well.
- Model Ensemble: An ensemble averaging of all the previous models.

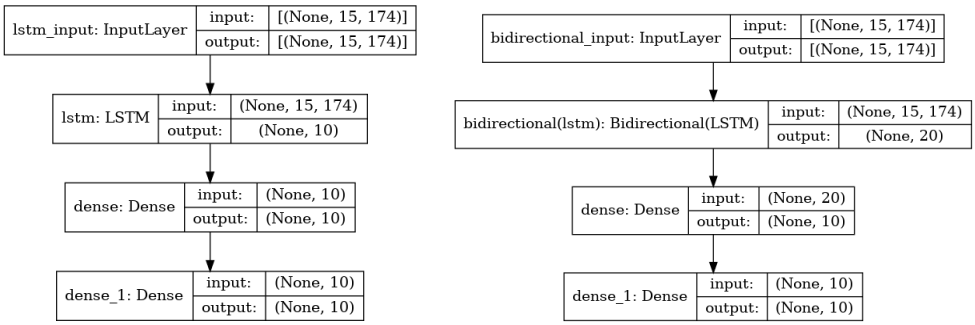
Each of these models was trained and evaluated for its effectiveness in classifying MSL signs. The designed architectures for the RNN and BRNN used in our tests are depicted in Figure 4.



(**a**) RNN architecture. (**b**) BRNN architecture.

Figure 4. RNN and BRNN architectures tested for sign language recognition.

The LSTM and Bidirectional LSTM architectures are shown in Figure 5. The GRU architecture is shown in Figure 6.



(**a**) LSTM architecture. (**b**) BLSTM architecture.

Figure 5. LSTM and BLSTM architectures tested for sign language recognition.

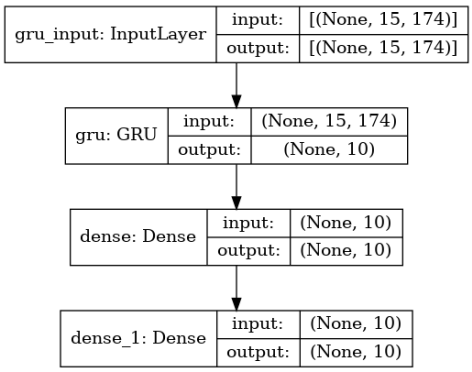


Figure 6. GRU architecture tested for sign language recognition.

3.5. Graphical User Interface Design

To improve user interaction with our translation system, we developed a graphical user interface (GUI). This GUI is aimed at enhancing the usability and accessibility of the system. A User Interface (UI) is essentially the point of interaction between the user and the system, enabling the user to input commands and data and to access the system's content. UIs are integral to a wide variety of systems, including computers, mobile devices, and games.

Beyond the UI, we also focused on User Experience (UX). UX is about the overall experience of the user, encompassing their emotions, thoughts, reactions, and behavior during both direct and indirect engagement with the system, product, or service. This aspect of design is critical because it shapes how users perceive and interact with the system.

The outcomes of our efforts to develop a compelling UI and UX for the translation system are detailed in Section 4.2.

4. Results

We compared the performance of the models described in Section 3.4 using our collected data for training and testing. We trained each model using early stopping with a patience of five epochs. Table 1 shows the epochs and accuracy per model.

To simulate a real environment, we tested our models under different perturbations:

1. Drop keypoints : in this test, we randomly remove keypoints to simulate real-life situations where the keypoints are incomplete.
2. Noise: in this test, we added Gaussian noise ($\mu = 0$, $\sigma = 0.3$) to the keypoints' coordinates to simulate noisy detections.
3. Drop keypoints + noise: in this test, we added both of the perturbations described.

We used the MacroF1, the unweighted mean of the F1 scores calculated per class, to compare the models. Equations (2)–(5) show the formulas used to compute this metric.

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{2}$$

$$\text{recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3}$$

$$\text{F}_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{4}$$

$$\text{MacroF}_1 = \frac{\sum(\text{F1 scores})}{\text{Number of classes}} \tag{5}$$

Table 1. Number of epochs and accuracy per model during training.

Model	Epochs	Accuracy
RNN	44	0.948
BRNN	49	0.970
GRU	83	0.988
LSTM	70	0.988
BLSTM	31	0.986
Transformer	46	0.942

Table 2 shows the MacroF1 score for the different models under the different test conditions. The model with the best performance overall was the Ensemble averaging followed by the Bidirectional LSTM.

Table 2. Comparison of Macro F1 scores across various models under diverse testing conditions, with the highest performing model in each scenario emphasized in bold.

Model	Baseline	Drop Keypoints	Noise	Drop Keypoints + Noise
RNN	0.9118	0.5648	0.5223	0.4181
BRNN	0.9542	0.7490	0.6140	0.4747
GRU	0.9750	**0.8977**	0.7205	0.5904
LSTM	0.9591	0.8044	0.8059	0.5935
BLSTM	0.9808	0.7358	0.8260	0.6455
Transformer	0.9333	0.6474	0.5999	0.3875
Ensemble	**0.9815**	0.8464	**0.8779**	**0.7224**

4.1. Ablation Studies

The hands' features are essential for sign language, not so much the body or face. In this section, we evaluate the accuracy of the models under different perturbations and with different combinations of features. We want to know how the body and facial features contribute to overall sign language recognition. Tables 3–9 show the accuracy for each model with different combinations of features.

Focusing on the best model results. Table 9 illustrates the performance of the ensemble averaging when applied to sign language recognition, with varying input features and under different conditions of data perturbation. The model's baseline accuracy is measured without any perturbations. When keypoints are dropped from the test data, simulating incomplete data, there is a noticeable decrease in accuracy across all input feature combinations, indicating that the model relies significantly on the complete set of keypoints to make accurate predictions. The addition of Gaussian noise, simulating variations in keypoint detection, also lowers the model's accuracy, but less dramatically than dropping points when using all the keypoints, suggesting that the model has some robustness to noise. However, when both perturbations are applied (dropping keypoints and adding noise), the accuracy declines substantially, underscoring that the integrity and quality of input data are critical for the model's performance. The most robust combination of input features against these perturbations is the "Hands + Face + Body", which achieves the best results under the most significant data perturbation. Combining the three sources of keypoints helps to obtain a more robust sign language recognition.

Table 3. Accuracy of the RNN with various input feature combinations, with the highest accuracy in each testing condition emphasized in bold.

Keypoints Selection	Baseline	Drop Keypoints	Noise	Drop Keypoints + Noise
Hands-only	0.9206	**0.6393**	0.4659	0.3833
Hands + Face	0.9260	0.5464	0.4276	0.3380
Hands + Body	**0.9530**	0.6020	0.4589	0.3822
Hands + Face + Body	0.9119	0.5583	**0.5248**	**0.4211**

Table 4. Accuracy of the BRNN with various input feature combinations, with the highest accuracy in each testing condition emphasized in bold.

Keypoints Selection	Baseline	Drop Keypoints	Noise	Drop Keypoints + Noise
Hands-only	0.9519	0.7154	0.5610	0.4168
Hands + Face	0.9460	0.4843	0.5826	**0.5421**
Hands + Body	**0.9703**	0.5399	0.5156	0.4551
Hands + Face + Body	0.9535	**0.7365**	**0.5939**	0.4551

Table 5. Accuracy of the LSTM with various input feature combinations, with the highest accuracy in each testing condition emphasized in bold.

Keypoints Selection	Baseline	Drop Keypoints	Noise	Drop Keypoints + Noise
Hands-only	0.9665	0.7429	0.6484	0.4492
Hands + Face	0.9719	0.5626	0.7224	**0.5809**
Hands + Body	**0.9730**	0.7143	0.6895	0.5248
Hands + Face + Body	0.9584	**0.7937**	**0.7991**	0.5712

Table 6. Accuracy of the BLSTM with various input feature combinations, with the highest accuracy in each testing condition emphasized in bold.

Keypoints Selection	Baseline	Drop Keypoints	Noise	Drop Keypoints + Noise
Hands-only	0.9719	0.8493	0.7899	**0.6976**
Hands + Face	0.9751	0.7375	**0.8596**	0.6409
Hands + Body	0.9751	**0.9190**	0.7861	0.6144
Hands + Face + Body	**0.9805**	0.7078	0.8191	0.6285

Table 7. Accuracy of the GRU with various input feature combinations, with the highest accuracy in each testing condition emphasized in bold.

Keypoints Selection	Baseline	Drop Keypoints	Noise	Drop Keypoints + Noise
Hands-only	**0.9778**	**0.9076**	0.6193	0.5259
Hands + Face	0.9719	0.8461	**0.7521**	0.5637
Hands + Body	0.9740	0.8061	0.6625	0.4298
Hands + Face + Body	0.9746	0.8930	0.7143	**0.5723**

Table 8. Accuracy of the Transformer with various input feature combinations, with the highest accuracy in each testing condition emphasized in bold.

Keypoints Selection	Baseline	Drop Keypoints	Noise	Drop Keypoints + Noise
Hands-only	0.9146	**0.7073**	0.5248	0.3207
Hands + Face	0.9114	0.5691	**0.6279**	0.3833
Hands + Body	0.9227	0.4076	0.5583	**0.3957**
Hands + Face + Body	**0.9314**	0.6268	0.5928	0.3650

Table 9. Accuracy of the ensemble with various input feature combinations, with the highest accuracy in each testing condition emphasized in bold.

Keypoints Selection	Baseline	Drop Keypoints	Noise	Drop Keypoints + Noise
Hands-only	0.9778	0.8558	0.7629	0.6069
Hands + Face	0.9805	**0.9497**	0.8639	0.6792
Hands + Body	**0.9827**	0.7786	0.8083	0.6987
Hands + Face + Body	0.9811	0.8336	**0.8752**	**0.7132**

4.2. User Interface

The system operates in two modes: translating from Mexican Sign Language to Spanish text (MSL-SPA) and from Spanish text to Mexican Sign Language (SPA-MSL). The user interface comprises three main views:

1. MSL-SPA Mode View: In this view, the system displays a live video feed from a user-facing camera. This feed shows the user's body keypoints before the system classifies the sign. This view is depicted in Figure 7.
2. SPA-MSL Mode View: This view is for the SPA-MSL mode. Here, the system displays the result of an MSL-SPA translation. The second user can then respond by typing on a keyboard. This typed text is used to generate a sign language animation for the other user. This view is illustrated in Figure 8.
3. Animation View: The third view presents the animation generated from the SPA-MSL mode. Users can see the sign language animation created from the text input. This view is shown in Figure 9.

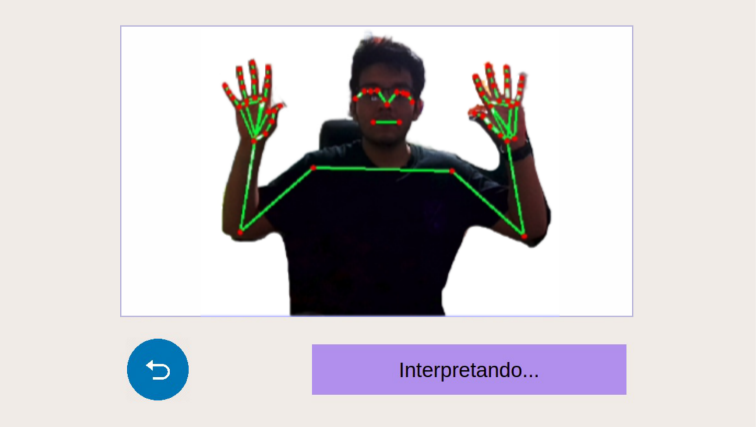


Figure 7. Mexican Sign Language to Spanish mode (MSL-SPA). The system shows the video from a front-facing camera with the body keypoints and the status *Interpretando* indicating that the system is interpreting the sign.

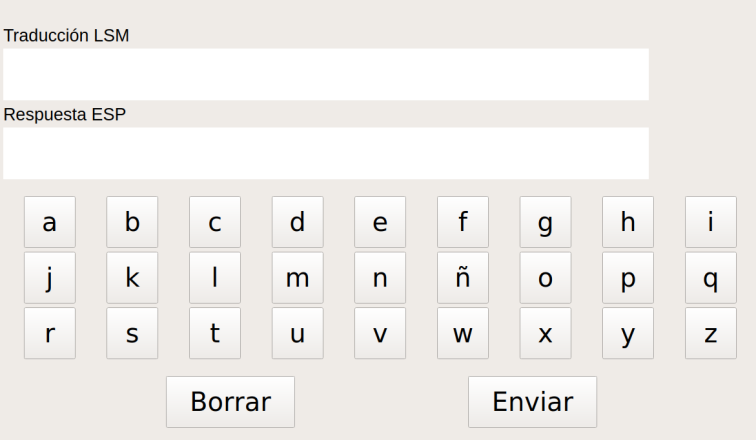


Figure 8. Interface for Spanish to Mexican Sign Language (SPA-MSL) Translation. The interface presents two input fields: 'Traducción LSM' for displaying the translation in Spanish and 'Respuesta ESP' for entering text to translate into MSL. A virtual keyboard is provided for the user to input text, which are then converted into MSL animations. The buttons 'Borrar' and 'Enviar' allow the user to delete the input or send it for translation, respectively.

Figure 9. SPA-MSL mode. The user can type text in Spanish and generate an animation showing the corresponding sign.

The back-end system consists of multiple parallel processes (daemons) running on independent CPU threads. Figure 10 shows the general pipeline with the back-end daemons for processing tasks in blue and interpreter tasks in yellow and green.

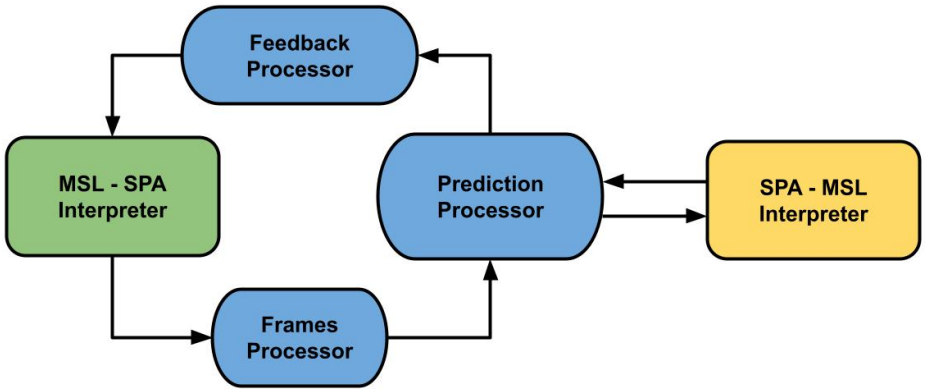


Figure 10. The back-end system consists of multiple parallel processes: frames, prediction, feedback processors, and the two interpreters.

The MSL-SPA mode has two main functions: it receives the video feed from the camera and displays the live output as shown in Figure 7, along with making sign predictions. Conversely, the SPA-MSL mode shows sign language translations from the MSL-SPA mode and can also accept text input to generate animations of sign language, which are then displayed to the partner. These modes are carried out through three parallel processes: the frames, prediction, and feedback processors.

The frames processor takes input frames and utilizes the Mediapipe keypoint detector to obtain the X, Y, and Z coordinates relative to the camera. Because the input is live video, the frames processor uses a sliding window consisting of 15 frames (as depicted in Figure 11) to create a feature vector that is placed into the prediction queue. Additionally, the frames processor performs an initial calibration with the user to ensure they are positioned at least two meters away from the camera. This distance ensures that the camera captures the necessary area and maintains consistency between the live data and the training data.

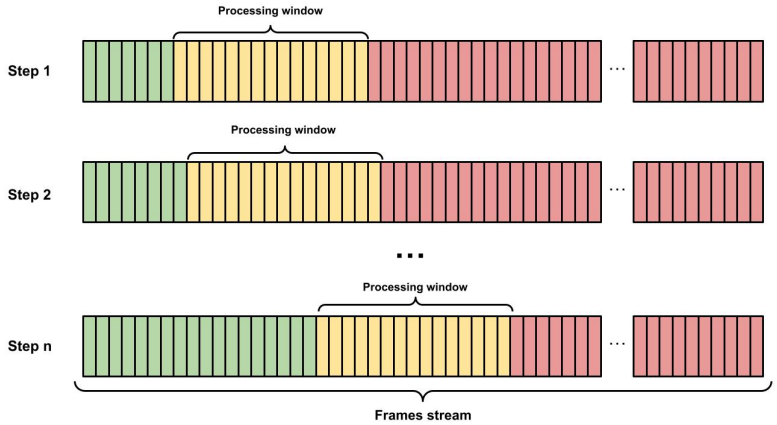


Figure 11. The frames processor applies a sliding window technique to the video stream, creating a feature vector for analysis. Green frames indicate those that have already been processed, yellow frames are currently undergoing processing, and red frames are queued for future processing.

The prediction processor can operate in two modes: MSL-SPA and SPA-MSL. In the MSL-SPA mode, it retrieves data from the frames queue produced by the frames processor and runs the prediction model to generate a list of results. The final prediction is based on the number of votes, and a notification is sent to the feedback processor. In contrast, in the SPA-MSL mode, it receives Spanish text and sends a notification to the feedback

processor. In the MSL-SPA mode, the feedback processor is responsible for displaying the text translation, while in the SPA-MSL mode, it displays an animation video. We have a database containing animation videos for each sign. The feedback processor searches for the corresponding video in the database and presents it to the user in the SPA-MSL mode, as shown in Figure 9.

We created the video animations using Blender3D with the free rig character Rain [47]. We chose to animate Rain (see Figure 9) to enhance the gestures for improved comprehension.

4.3. Prototype

The final prototype consists of a Jetson Nano board that has two seven-inch LCD touch displays. One of the screens displays the MSL-SPA mode, while the other shows the SPA-MSL mode. The MSL-SPA mode screen has an OAK-D camera attached to it for video capture. We used free-access 3D models from Thingiverse [48] to create the mounting structure of the board and screens and added the camera stand on top of one display. Finally, we 3D printed the structure. You can see the final prototype in Figure 12.

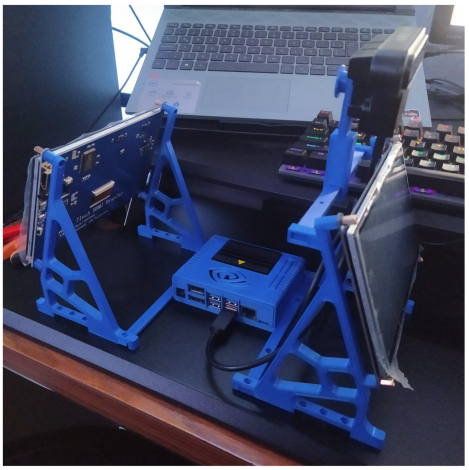


Figure 12. Final Prototype. The image shows the two touchscreens back-to-back with the main processor in the middle.

5. Conclusions

Our bidirectional Mexican Sign Language (MSL) translation system aims to bridge the communication gap between the deaf community and the hearing world. Utilizing machine learning, including recurrent neural networks, transformers, and keypoint detection, our system shows promise in enabling seamless communication, with the promise of integrating individuals with hearing disabilities into society and education more effectively. This innovation emphasizes the importance of inclusive technology and the role of artificial intelligence in surmounting language barriers. The project's key successes lie in its bidirectional translation system, characterized by efficiency and accuracy. The use of MediaPipe for keypoint detection, along with RNN, BRNN, LSTM, GRU, and Transformer architectures, facilitates accurate translation of signs into text and vice versa. The system's real-time functionality, adaptability to various sign language variations, and user-friendly interface make it a practical tool for everyday use.

However, the study faced challenges related to the variability and complexity of sign language gestures, and the scarcity of diverse sign language datasets, impacting the training accuracy of the models. The system's performance under different real-world conditions like varied lighting and backgrounds also presents ongoing challenges. These issues highlight the necessity for further research, particularly in dataset development and enhancing the system's adaptability. Future directions include expanding the system to more languages and sign language variants, refining algorithms for complex signs and non-manual signals, and collaborating with the deaf community for feedback and improvements.

This research opens pathways for more inclusive communication technologies, aiming to significantly reduce communication barriers for the deaf and hard-of-hearing, leading to a more inclusive society.

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

Funding: This work was supported by IPN-SIP Grant 20232841.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Acknowledgments: We extend our gratitude to the Autonomous University of Queretaro and the National Council of Humanities, Sciences, and Technologies (CONAHCYT) for their backing, facilitated via the National Research System (SNI). We acknowledge the Instituto Politecnico Nacional for its contribution through the IPN-SIP Grant 20232841. Additionally, we acknowledge the use two AI tools. Grammarly Assistant for improving the grammar, clarity, and overall readability of the manuscript and GPT-4 to help with the wording, formatting, and styling of the manuscript. It provided suggestions for phrasing concepts in a more comprehensible manner and helped in structuring the content effectively.

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

References

1. Wolfe, R.; Braffort, A.; Efthimiou, E.; Fotinea, E.; Hanke, T.; Shterionov, D. Special issue on sign language translation and avatar technology. *Univers. Access Inf. Soc.* **2023**, *22*, 1–3. [CrossRef]
2. Welcome to the SIGN-HUB Platform. 2020. Available online: https://thesignhub.eu/ (accessed on 25 May 2023).
3. Valli, C.; Lucas, C. *Linguistics of American Sign Language: An Introduction*; Gallaudet University Press: Washington, DC, USA, 2000.
4. Traxler, C.B. The Stanford Achievement Test: National norming and performance standards for deaf and hard-of-hearing students. *J. Deaf. Stud. Deaf. Educ.* **2000**, *5*, 337–348. [CrossRef] [PubMed]
5. Ruvalcaba, D.; Ruvalcaba, M.; Orozco, J.; López, R.; Cañedo, C. Prototipo de guantes traductores de la lengua de señas mexicana para personas con discapacidad auditiva y del habla. *Mem. Congr. Nac. Ing. Biomédica* **2018**, *5*, 350–353.
6. Hernández Samacá, S.F. Desarrollo de Guantes Traductores de Lengua de Señas Colombiana a Lengua Natural. Master's Thesis, Universidad Autónoma de Bucaramanga UNAB, Bucaramanga, Colombia, 2022.
7. Navarrete, P.M.B.; Giraldo, J.; Rodríguez, S. Kit didáctico para el aprendizaje del lenguaje de señas ecuatoriano. *Rev. InGenio* **2021**, *4*, 1–10. [CrossRef]
8. Bungeroth, J.; Ney, H. Statistical sign language translation. In Proceedings of the Workshop on Representation and Processing of Sign Languages, LREC, Lisbon, Portugal, 30 May 2004; Volume 4, pp. 105–108.
9. Stokoe, J.; William, C. Sign language structure: An outline of the visual communication systems of the American deaf. *J. Deaf. Stud. Deaf. Educ.* **2005**, *10*, 3–37. [CrossRef] [PubMed]
10. San-Segundo, R.; Barra, R.; Córdoba, R.; D'Haro, L.F.; Fernández, F.; Ferreiros, J.; Lucas, J.M.; Macías-Guarasa, J.; Montero, J.M.; Pardo, J.M. Speech to sign language translation system for Spanish. *Speech Commun.* **2008**, *50*, 1009–1020. [CrossRef]
11. Pichardo-Lagunas, O.; Partida-Terrón, L.; Martínez-Seis, B.; Alvear-Gallegos, A.; Serrano-Olea, R. Sistema de traducción directa de español a LSM con reglas marcadas. *Res. Comput. Sci.* **2016**, *115*, 29–41. [CrossRef]
12. Qiao, S.; Wang, Y.; Li, J. Real-time human gesture grading based on OpenPose. In Proceedings of the 2017 10th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Shanghai, China, 14–16 October 2017; pp. 1–6.
13. Barrera Melchor, F.; Alcibar Palacios, J.C.; Pichardo-Lagunas, O.; Martinez-Seis, B. Speech to Mexican Sign Language for Learning with an Avatar. In Proceedings of the Mexican International Conference on Artificial Intelligence, Mexico City, Mexico, 12–17 October 2020; pp. 179–192.

14. Sosa-Jiménez, C.O.; Ríos-Figueroa, H.V.; Solís-González-Cosío, A.L. A Prototype for Mexican Sign Language Recognition and Synthesis in Support of a Primary Care Physician. *IEEE Access* **2022**, *10*, 127620–127635. [CrossRef]
15. Kinect for Windows. Available online: https://learn.microsoft.com/en-us/windows/apps/design/devices/kinect-for-windows (accessed on 4 January 2024).
16. Martínez-Gutiérrez, M.E.; Rojano-Cáceres, J.R.; Benítez-Guerrero, E.; Sánchez-Barrera, H.E. Data Acquisition Software for Sign Language Recognition. *Res. Comput. Sci.* **2019**, *148*, 205–211. [CrossRef]
17. Martinez-Seis, B.; Pichardo-Lagunas, O.; Rodriguez-Aguilar, E.; Saucedo-Diaz, E.R. Identification of Static and Dynamic Signs of the Mexican Sign Language Alphabet for Smartphones using Deep Learning and Image Processing. *Res. Comput. Sci.* **2019**, *148*, 199–211. [CrossRef]
18. Carmona-Arroyo, G.; Rios-Figueroa, H.V.; Avendaño-Garrido, M.L. Mexican Sign-Language Static-Alphabet Recognition Using 3D Affine Invariants. In *Machine Vision Inspection Systems, Volume 2: Machine Learning-Based Approaches*; Wiley: Hoboken, NJ, USA, 2021; pp. 171–192.
19. Naranjo-Zeledón, L.; Chacón-Rivas, M.; Peral, J.; Ferrández, A. Architecture design of a reinforcement environment for learning sign languages. *PeerJ Comput. Sci.* **2021**, *7*, e740. [CrossRef] [PubMed]
20. Trujillo-Romero, F.; Bautista, G.G. Reconocimiento de palabras de la Lengua de Señas Mexicana utilizando información RGB-D. *ReCIBE Rev. Electrón. Comput. Inform. Bioméd. Electrón.* **2021**, *10*, C2–C23.
21. Jimenez, J.; Martin, A.; Uc, V.; Espinosa, A. Mexican Sign Language Alphanumerical Gestures Recognition using 3D Haar-like Features. *IEEE Lat. Am. Trans.* **2017**, *15*, 2000–2005. [CrossRef]
22. Cervantes, J.; García-Lamont, F.; Rodríguez-Mazahua, L.; Rendon, A.Y.; Chau, A.L. Recognition of Mexican sign language from frames in video sequences. In Proceedings of the Intelligent Computing Theories and Application: 12th International Conference, ICIC 2016, Lanzhou, China, 2–5 August 2016; Proceedings, Part II 12 ; pp. 353–362.
23. Camgoz, N.C.; Koller, O.; Hadfield, S.; Bowden, R. Sign language transformers: Joint end-to-end sign language recognition and translation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 10023–10033.
24. Zhang, H.; Sun, Y.; Liu, Z.; Liu, Q.; Liu, X.; Jiang, M.; Schafer, G.; Fang, H. Heterogeneous attention based transformer for sign language translation. *Appl. Soft Comput.* **2023**, *144*, 110526. [CrossRef]
25. Guo, Z.; Hou, Y.; Hou, C.; Yin, W. Locality-Aware Transformer for Video-Based Sign Language Translation. *IEEE Signal Process. Lett.* **2023**, *30*, 364–368. [CrossRef]
26. Narayanan, M.B.; Bharadwaj, K.M.; Nithin, G.; Padamnoor, D.R.; Vijayaraghavan, V. Sign Language Translation Using Multi Context Transformer. In Proceedings of the Advances in Soft Computing: 20th Mexican International Conference on Artificial Intelligence, MICAI 2021, Mexico City, Mexico, 25–30 October 2021; Proceedings, Part II 20; pp. 311–324.
27. De Coster, M.; D'Oosterlinck, K.; Pizurica, M.; Rabaey, P.; Verlinden, S.; Van Herreweghe, M.; Dambre, J. Frozen pretrained transformers for neural sign language translation. In Proceedings of the 18th Biennial Machine Translation Summit (MT Summit 2021), Macau, China, 4–8 September 2021; pp. 88–97.
28. Gibet, S.; Marteau, P.F. Signing Avatars-Multimodal Challenges for Text-to-sign Generation. In Proceedings of the 2023 IEEE 17th International Conference on Automatic Face and Gesture Recognition (FG), Waikoloa Beach, HI, USA, 5–8 January 2023; pp. 1–8.
29. Wolfe, R.; McDonald, J.C.; Hanke, T.; Ebling, S.; Van Landuyt, D.; Picron, F.; Krausneker, V.; Efthimiou, E.; Fotinea, E.; Braffort, A. Sign language avatars: A question of representation. *Information* **2022**, *13*, 206. [CrossRef]
30. Filhol, M.; McDonald, J.; Wolfe, R. Synthesizing sign language by connecting linguistically structured descriptions to a multi-track animation system. In Proceedings of the Universal Access in Human–Computer Interaction, Designing Novel Interactions: 11th International Conference, UAHCI 2017, Held as Part of HCI International 2017, Vancouver, BC, Canada, 9–14 July 2017; Proceedings, Part II 11; pp. 27–40.
31. Angelini, R. Contrasting Technologists' and Activists' Positions on Signing Avatars. In Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems, Hamburg, Germany, 23–28 April 2023; pp. 1–6; Extended Abstracts.
32. Moncrief, R.; Choudhury, S.; Saenz, M. Efforts to Improve Avatar Technology for Sign Language Synthesis. In Proceedings of the 15th International Conference on PErvasive Technologies Related to Assistive Environments, Corfu, Greece, 29 June–1 July 2022; pp. 307–309.
33. De Martino, J.M.; Silva, I.R.; Marques, J.G.T.; Martins, A.C.; Poeta, E.T.; Christinele, D.S.; Campos, J.P.A.F. Neural machine translation from text to sign language. *Univers. Access Inf. Soc.* **2023**, 1–14. [CrossRef]
34. Papadimitriou, K.; Potamianos, G.; Sapountzaki, G.; Goulas, T.; Efthimiou, E.; Fotinea, S.E.; Maragos, P. Greek sign language recognition for an education platform. *Univers. Access Inf. Soc.* **2023**, 1–18. [CrossRef]
35. Raspberry Pi 4. Product Description. Available online: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/ (accessed on 4 January 2024).
36. UP Squared Series Specifications. Available online: https://up-board.org/upsquared/specifications/ (accessed on 4 January 2024).
37. NVIDIA Developer. Jetson Nano Developer Kit. Available online: https://developer.nvidia.com/embedded/jetson-nano-developer-kit (accessed on 4 January 2024).
38. OAK-D—Product Information. Available online: https://shop.luxonis.com/collections/oak-cameras-1/products/oak-d (accessed on 4 January 2024).

39. OV9282—DepthAI Hardware Documentation. 2023. Available online: https://docs.luxonis.com/projects/hardware/en/latest/pages/articles/sensors/ov9282/#ov9282 (accessed on 26 December 2023).
40. Depth Accuracy—DepthAI Hardware Documentation. 2023. Available online: https://docs.luxonis.com/projects/hardware/en/latest/pages/guides/depth_accuracy/#p-75mm-baseline-distance-oaks (accessed on 26 December 2023).
41. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. *Learning Internal Representations by Error Propagation*; Defense Technical Information Center: For Belvoir, VA, USA, 1985.
42. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]
43. Schuster, M.; Paliwal, K.K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **1997**, *45*, 2673–2681. [CrossRef]
44. Graves, A.; Fernández, S.; Schmidhuber, J. Bidirectional LSTM networks for improved phoneme classification and recognition. In Proceedings of the International Conference on Artificial Neural Networks, Warsaw, Poland, 10–15 September 2005; pp. 799–804.
45. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
46. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, Ł.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2017; pp. 5998–6008.
47. Dzadik, D. Rain–Character Rig. Blender Studio. 2020. License: CC-BY. Available online: https://studio.blender.org/characters/5f1ed640e9115ed35ea4b3fb/v2/ (accessed on 4 January 2024).
48. Thingiverse. Ultimaker Thingiverse. 2023. Available online: https://www.thingiverse.com/ (accessed on 4 January 2024).