Smart Home Control through Sign Language Recognition

A. De Toro-Castro[1 [0009−0005−7681−0282]], M. Lupión[⋆1,2 [0000−0001−7697−8062]], V. González-Ruiz[1 [0000−0001−6495−4856]], J.F. Sanjuan[1 [0000−0002−2874−0903]], and P.M. Ortigosa[1 [0000−0001−6514−6543]]

[1] Dpt. of Informatics, ceiA3, Univ. of Almería, 04120, Almería, Spain
adc144@inlumine.ual.es {marcoslupion,vruiz,jsanjuan,ortigosa}@ual.es
[2] School of Computing, Ulster University, BT15 1ED, Belfast, UK
M.LupionLorente@ulster.ac.uk

Abstract. Sign language recognition plays a crucial role in improving accessibility for individuals with hearing impairments, particularly within smart home environments. This paper presents a novel real-time Spanish Sign Language (SSL) recognition system designed to facilitate interaction with smart home devices. The proposed system leverages a hybrid CNN-LSTM neural network architecture to process real-time videos. The keypoints required for recognition (hands and body) are extracted using MediaPipe, ensuring a privacy-preserving representation of user gestures. A newly recorded SSL dataset, specifically tailored for smart home control, is introduced to improve recognition accuracy. The system is fully integrated with a KNX-based smart home infrastructure at the University of Almería, enabling seamless control of household components, such as lights, shutters, and air conditioning, via sign language commands. Experimental results confirm the effectiveness of the proposed approach, achieving a high accuracy of 87.93% and an F1-score of 0.78 in gesture recognition across different users under a leave-one-subject-out validation, while maintaining real-time performance.

Keywords: Language sign recognition · Smart Environments · Artificial Intelligence

1 Introduction

Hearing impairment, whether partial or total, imposes significant barriers to effective communication. In Europe, it is estimated that around 190 million people—approximately 20% of the population—experience some level of hearing loss or deafness [16]. Although many individuals with hearing loss benefit from assistive technologies like hearing aids and cochlear implants to enhance auditory input, a distinct subset of the deaf community, estimated at roughly 750,000 people in Europe, relies exclusively on sign language for communication [4]. Given the limited number of hearing individuals proficient in sign language, researchers

⋆ Corresponding author

have developed automated Sign Language Recognition (SLR) systems that employ artificial intelligence (AI) and computer vision to interpret sign language from visual or sensor-based inputs [4,14]. These systems convert sign language into text or speech in real time, promising enhanced inclusivity and reduced reliance on human interpreters. SLR typically involves the segmentation of sign language into basic units, known as "glosses," each corresponding to a specific word or phrase.

Due to their higher recognition accuracy and ease of deployment, vision-based approaches have gained increasing attention. Architectures such as convolutional neural networks (CNNs) [3], recurrent neural networks (RNNs) [**?**], and transformers [15] have proven effective in automatically extracting and fusing spatial and temporal features from video sequences, thus significantly enhancing recognition performance.

Ambient intelligence within smart environments (SEs) is designed to assist individuals (particularly the elderly and those with disabilities or illnesses) in their daily activities [12]. SE applications also include fall detection and assessment [10], and medication reminders [9]. Interaction with SEs is commonly facilitated by voice-controlled assistants, such as Amazon Alexa or Google Assistant, which have been shown to improve usability compared to traditional mobile or Web interfaces [6]. For instance, fall detection systems may assess the condition of a user post-fall [11], while platforms like OpenHab or HomeAssistant enable voice-activated control of home devices.

Despite the clear need for inclusive technology, few SE solutions have been tailored to accommodate sign language users [7]. Efforts to merge SLR with SEs have focused on developing systems that perform real-time sign language interpretation using wearable or visual inputs [1]. One pioneering study in this field is presented in [8], where the authors designed a dynamic SLR system based on 3D CNNs for smart home interaction. Their system translates video sequences into home automation commands (e.g., "Turn on air conditioner in the bathroom"), recognizing up to 25 gestures from the Croatian Sign Language Dataset. While numerous studies report high recognition performance in controlled environments [14], many of these approaches are impractical for real-time deployment due to their heavy computational demands. Moreover, discrepancies between the perspectives in training datasets and those encountered in real-world applications can further compromise recognition accuracy.

Summarizing, we introduce an innovative DL solution for real-time Spanish sign language recognition. Our research work has focused mainly on determining the performance difference (making an objective comparison) between two AI models, one based on a CNN, which only uses spatial information, and another model which is an extension of the previous one through an LTSM module, which also exploits temporal information. The models have been trained on a SSL dataset specifically created to include vocabulary and commands relevant to smart home operations. Apart from this, we have implemented a software application that acts on the different elements of the smart home (such as windows, shutters, air conditioning devices, etc.) ensuring a seamless interaction of

the user with the environment. The principal contributions of this work are as follows:

- The development of a robust, real-time Spanish sign language recognition system for smart environments.
- The creation of a novel SSL dataset specifically designed for smart home interaction commands.
- The implementation of a hybrid CNN-LSTM model optimized for deployment on resource-constrained devices.

The remainder of this paper is organized as follows. Section 2 details the proposed methodology, Section 3 describes the experimental setup and presents the evaluation results, and Section 4 concludes the paper with a discussion of future research directions.

2 Methodology

2.1 Architecture

The system architecture (shown in Figure 1) consists of several interconnected components for efficient sign language translation and smart home control. Following, the components and their task in the overall systems are described:

- **Camera**, that captures real-time video of the user's sign language gestures, at a rate of 4 Frames Per Second (FPS).
- The **window splitting module**, which divides the video into subsequence of 12 frames (3-seconds windows) for temporal gesture analysis [13].
- A **Pose recognition component (Mediapipe)** processes each frame to detect and extract three different types of keypoints (hands and body).
- **Image generator**, that organizes the extracted keypoints into images (of 100×100 pixels) with black backgrounds, structuring the AI model's sequence data.
- The **AI model**, which analyzes the batch of images, classifying the sequences into gestures.
- The **smart home integration script**, that summarizes the gestures recognized by the AI model and builds the commands that allow to control the different smart home components. A Spacelynk server receives the commands and triggers actions to control devices such as lights, windows, and air conditioners.

2.2 Keypoints Extraction

Keypoints detection is performed using MediaPipe models [2] to capture the spatial positions of essential landmarks in human pose estimation and hand tracking. The extracted keypoints serve as a fundamental input representation for downstream tasks such as gesture classification. We extract keypoints from two primary regions:

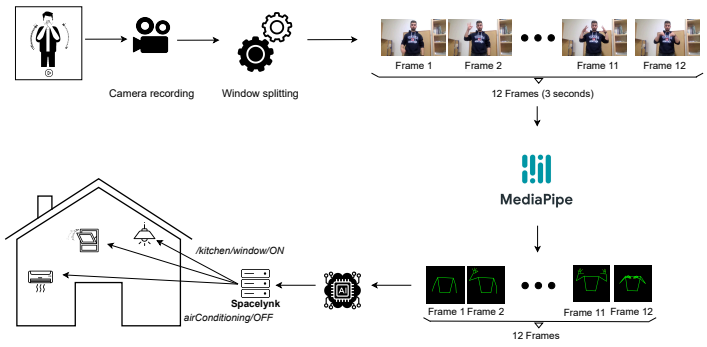

Fig. 1: System Architecture.

- **Pose**: A subset of 8 upper-body landmarks for tracking arms and torso movements. These keypoints are selected to track the shoulders, elbows, wrists, and hips, which are essential for interpreting sign language gestures. The focus on upper body movements allows for efficient and accurate recognition of gestures involving arm positioning and torso orientation.
- **Hands**: Twenty-one keypoints covering the entire hand structure for precise gesture tracking. With 21 keypoints per hand, the total reaches 42 when both hands are detected. Hand gestures are central to sign language, and these keypoints track the wrist, finger positions, and joint movements, allowing for detailed and accurate gesture recognition. This level of detail enables the system to detect complex hand movements and finger configurations that are critical for understanding sign language.

Once these keypoints are extracted, they are used to create a 100x100 pixel image that accurately represents the sign speaker or user. This image is constructed by mapping the (x, y) coordinates of the detected keypoints onto a grid. The background is set to black to homogenize different backgrounds, and the keypoints are normalized, placing the user in the centre of the image to avoid changes in the scale of different users. The result is a privacy-preserving representation of the user, as it avoids using any personal identifiable features such as faces or full body images. Instead, the image keeps enough detail to recognize and classify sign language gestures while ensuring the privacy of the individual, as shown in Figure 2.

2.3 Models

Batches of twelve images (captured in a 3-second window [13]) are processed in order to recognize a gesture on them. The input captures both spatial information (e.g., the person, the background, and location) and temporal relations (e.g., the movement of the person across images). In this case, two different neural network configurations have been designed and evaluated:

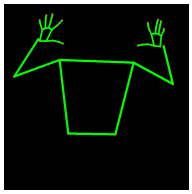

Fig. 2: Keypoint-based structural pose representation.

- **CNN**: The CNN is particularly adept at capturing spatial relationships in images, thanks to its convolutional layers. In this case, given that the input is a batch of data, each CNN layer is applied to each layer separately. A total of 4 convolutional layers (with filter size of 3, and number of filters ranging from 32 to 128) are incorporated, followed by a global average pooling layer that flattens the data, and two fully connected layers (with 64 and 11 neurons) that map the spatial features into the different gestures. Furthermore, dropout (20% of neurons) and batch normalization layers are incorporated between the convolutional layers to reduce overfitting and accelerate the training, respectively.
- **CNN + LSTM**: Since the input images also contain temporal information, we process image sequences (12 images/sequence) to recognize each gesture. Recurrent neural networks, particularly those incorporating LSTM units, are well-suited for temporal data processing. However, these networks cannot directly handle raw images, so we propose combining them with convolutional layers that first extract spatial features. For this reason, the CNN+LSTM architecture proposed is the same as the CNN described before but incorporates a LSTM layer with 12 units before the fully connected layers.

In both cases, the models have been trained using Adam as optimizer, with a decaying learning rate starting at 0.0005. The number of epochs was set to 200, and an early stopping of 20 epochs avoided evaluating the full training.

2.4 Smart Home Integration

The smart home at the University of Almería [12] is a laboratory specifically designed to facilitate research, experimentation, and educational activities related to ambient intelligence solutions. This environment enables both university staff and students to explore, develop, and learn about innovative smart home technologies. The facility is fully interconnected through a KNX bus system, which allows wired communication among various devices. A comprehensive map of the installed devices within the laboratory is presented in Figure 3.

The system incorporates an array of sensors capable of monitoring environmental parameters such as humidity, temperature, and CO_2 levels. Additionally, actuators are integrated into the environment, including window blinds, air conditioning units, lighting systems, and shutters, allowing comprehensive automation and control functionalities.

Fig. 3: Diagram of sensors and actuators in the Smart Home environment.

The central control system of the smart home is managed by a Web server operating on the SpaceLynk LSS100200 platform. This server functions as the primary interface for the KNX system, allowing users to manage and monitor device statuses through either a Web-based interface or a RESTful API. This API facilitates programmatic interactions via HTTP requests, enabling integration with external applications and automation scripts.

Each sensor and actuator within the system is uniquely identified by a device address structured in a three-level hierarchy (e.g., 3/1/1 corresponds to the kitchen light). Through the API, it is possible to both query the status of devices and modify their operational state.

The primary objective of this study is to implement device control using a SLR system integrated with a Language-Specific Engine. The control logic relies on accurately identifying commands composed of four distinct elements. An API call is triggered only when all components of the command are correctly recognized. The command structure is defined as follows: **Room** (*Kitchen*, *Bedroom*, *Bathroom*, *Living Room*), **Item** (*Light*, *Shutter*, *Air Conditioning*), and **Action** (*Turn on*, *Turn off*, *Set to the middle (50%)*). This framework ensures precise and efficient interaction with the smart home system, facilitating natural language-based control while ensuring operational accuracy and minimizing potential errors in device activation.

3 Experimentation

3.1 HPC infrastructure

The experimentation has been carried out in the cluster of the SAL research group at the University of Almería, Spain. Specifically, one node equipped with

a single AMD EPYC 7513 32-Core Processor, 128 GB of DDR4 RAM at 3200 MT/s, and an NVIDIA Tesla A30 GPU with 24 GB of HBM2 memory.

3.2 Dataset

As mentioned in the introduction, there is no large SSL dataset that incorporates all the possible gestures that allow to control the smart home. For this reason, a new specific dataset has been recorded. Its details are the following: (1) *Number of actors*: 4, (2) *Number of words*: 11, (3) *Total number of sequences per user*: 110 = 10 times each sign and 11 signs. Augmented synthetically to 2750 for each user, (4) *Total number of sequences*: 11000 in total. 2750 for each user and 4 users, and (5) *Number of locations*: 3.

The recording and annotation of the dataset is a very costly and time-consuming process. To automate the recording, a Web tool has been designed. The main functionality can be appreciated in Figure 4. In such a screen, the camera is activated and the Web system requires the user to perform a gesture in a 3-second time window. At the right part, the ground truth gesture performed by a specialist actor is provided. Once the videos are recorded, these are uploaded into a server, executing the Mediapipe framework to extract the keypoints.

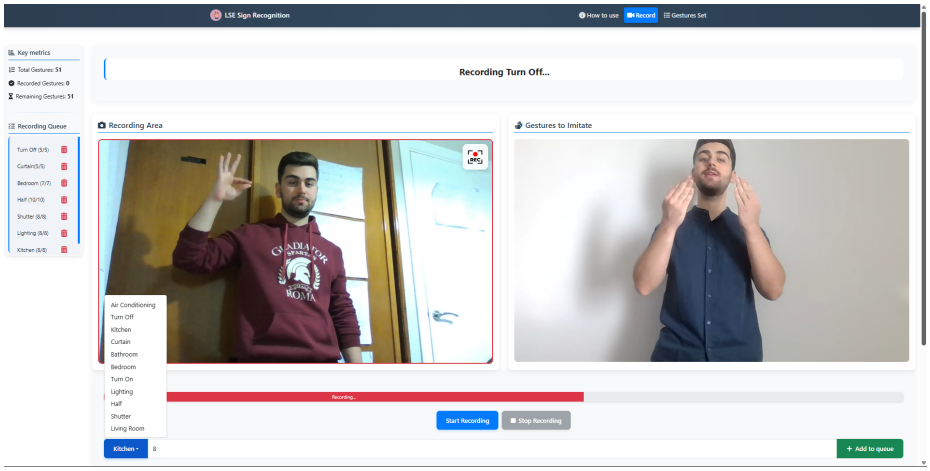

Fig. 4: Web tool to create the dataset.

3.3 Model Performance

In this section, the evaluation of the models is included. It has been carried out using a cross-validation Leave-One-Subject-Out approach [5]. Taking into account the 4 users included in the dataset, the experimentation consisted of 4

training processes for each model. In each training, 3 users were used to train/-validate the ANN, and one user for testing. This validation process can determine the generalization capabilities of the model with new and unseen users.

In the first experiment, two ANN (Artificial Neural Network) architectures were compared: CNN and CNN-LSTM (see Section 2.3). Results are shown in Table 1, which includes the training time, the number of epochs, and the train, validation and test accuracies of both models. It is important to notice that the different values are the average of the training for the different (4) users.

As can be appreciated, the CNN-LSTM model achieves a higher accuracy in validation and testing, which demonstrates that it is more convenient for a real environment. However, the training time (and the number of epochs) are slightly higher than the CNN model, which includes fewer parameters. It is also important to note that training and validation datasets achieve 100% accuracy, while the test dataset (with the unseen user) achieves 87.93%, demonstrating the ability of the model to generalize with new users. In case of following a standard validation by shuffling data of all users, the test accuracy reaches 100%.

			Accuracy		
Model	**Training Time (s)**	**Num. epochs**	**Train**	**Validation**	**Testing**
CNN	518	44	1.0000	0.9000	0.8145
CNN-LSTM	664	53	1.0000	1.0000	0.8793

Table 1: Performance of model architectures.

Second, the F1-score metric of the gestures for the test dataset is included in Table 2. It is important to note that the 4 different trainings are included to appreciate the differences between users. It can be appreciated that mean F1-score values range from 0.71 (in the first user) to 0.87 in the case of the second user, with a mean value of 0.78. Although the data from the different users is balanced, the differences in the execution of the different gestures can produce such disparities in the accuracy of the model.

Regarding the different gestures, the maximum F1-score value is achieved by the "Air Conditioning" gesture, with a F1-score value of 0.91. It is important to remark that this gesture is very different from the others, being the only that has both hand movements at head height throughout the entire execution of the gesture. In contrast, the "Half" gesture achieves a lower F1-score of only 0.67 and fails to be recognized in the case of the first user. This poor performance is primarily due to its high visual similarity with the "Kitchen" gesture; both gestures involve similar hand positions and motion trajectories. Consequently, the model often confuses one with the other, leading to a misclassification of "Half" as "Kitchen".

Finally, the inference speed of the model was evaluated in the final hardware, i.e. an MSI Modern 15 series equipped with an Intel Core i7 Ultra processor (Meteor Lake, 1.4 GHz). On this configuration, the model achieves an average inference speed of 6.33 FPS, with an average frame processing time of 135.26 ms and an average model inference time of 133.06 ms. These values yield a theoretical maximum of approximately 6.83 FPS, based on the combined processing and inference time per frame.

Gesture		F1-Score				
ID	**Meaning**	**1**	**2**	**3**	**4**	**Mean**
1	Air Conditioning	0.9482	0.9879	0.9470	0.7924	0.9188
2	Turn Off	0.7368	0.7939	0.1011	0.8908	0.6306
3	Kitchen	0.6568	0.9718	0.7974	0.9859	0.8454
4	Curtain	0.8460	0.9443	0.7814	0.8208	0.8481
5	Bedroom	0.8556	0.8609	0.8861	0.8104	0.8532
6	Turn On	0.7205	0.7817	0.7778	0.8714	0.7878
7	Lighting	0.7893	0.9384	0.5569	0.7911	0.7689
8	Half	0.0000	0.9280	0.7911	0.9676	0.6716
9	Shutter	0.8039	0.8914	0.9153	0.8235	0.8585
10	Living Room	0.6737	0.7717	0.5476	0.9843	0.7443
11	Bathroom	0.7924	0.8000	0.8372	0.5994	0.7572
All		0.7112	0.8790	0.7217	0.8489	0.7894

Table 2: F1-score for the different users in test dataset.

4 Conclusions and future works

This work has introduced a real-time system for Spanish sign language recognition integrated within a KNX-based smart home environment. The proposed neural network combines spatial and temporal modelling via a CNN-LSTM architecture, using MediaPipe to extract pose and hand-based keypoints and transform them into privacy-preserving image representations. The results obtained through cross-user validation demonstrate the model's ability to generalize across different individuals, achieving an average F1-score of 0.789 in new unseen users.

Despite these promising outcomes, certain gestures such as "Half" and "Turn Off" exhibit reduced recognition performance, due to high similarity between signs and variability in user execution. These findings highlight the need for improved training data and model refinement.

Future work will focus on several directions. Firstly, the dataset will be expanded to include a larger number of users, recording conditions, and sign vocabulary. Ultimately, usability studies with sign language users will be conducted to assess the practical impact of the system and identify additional accessibility needs.

Acknowledgements This work has been funded by the projects R+D+i PID2021-123278OB-I00 and PDC2022-133370-I00 from MCI-N/AEI/10.13039/501100011033/ and ERDF funds; and the Department of Informatics of the University of Almería.

References

1. Adeyanju, I.A., Bello, O.O., Adegboye, M.A.: Machine learning methods for sign language recognition: A critical review and analysis. Intelligent Systems with Applications **12**, 200056 (2021)

2. Bazarevsky, V.: Blazepose: On-device real-time body pose tracking. arXiv preprint arXiv:2006.10204 (2020)
3. Cui, R., Liu, H., Zhang, C.: A deep neural framework for continuous sign language recognition by iterative training. IEEE Transactions on Multimedia **21**(7), 1880–1891 (2019)
4. European Centre for Modern Languages: Día europeo de las lenguas. https://edl.ecml.at/Facts/FAQsonsignlanguage/tabid/2741/language/Default.aspx (February 2025), (Accessed on 04/02/2025)
5. Gholamiangonabadi, D., Kiselov, N., Grolinger, K.: Deep neural networks for human activity recognition with wearable sensors: Leave-one-subject-out cross-validation for model selection. Ieee Access **8**, 133982–133994 (2020)
6. Hoy, M.B.: Alexa, siri, cortana, and more: an introduction to voice assistants. Medical reference services quarterly **37**(1), 81–88 (2018)
7. Kate, J.B.: Smart home system for the deaf and hard-of-hearing (2015), https://api.semanticscholar.org/CorpusID:33454053
8. Kraljević, L., Russo, M., Pauković, M., Šarić, M.: A dynamic gesture recognition interface for smart home control based on croatian sign language. Applied Sciences **10**(7), 2300 (2020)
9. Liu, Z., Zhang, C., Peng, H., Xu, Q., Gao, Y.: Drug distribution management system based on iot. KSII Transactions on Internet and Information Systems (TIIS) **16**(2), 424–444 (2022)
10. Lupión, M., González-Ruiz, V., Sanjuan, J.F., Medina-Quero, J., Ortigosa, P.M.: Detection of unconsciousness in falls using thermal vision sensors. In: The International Conference on Innovations in Computing Research. pp. 3–12. Springer (2022)
11. Lupión, M., González-Ruiz, V., Sanjuan, J.F., Ortigosa, P.M.: Privacy-aware fall detection and alert management in smart environments using multimodal devices. Internet of Things p. 101526 (2025)
12. Lupión, M., Medina-Quero, J., Sanjuan, J.F., Ortigosa, P.M.: Dolars, a distributed on-line activity recognition system by means of heterogeneous sensors in real-life deployments—a case study in the smart lab of the university of almería. Sensors **21**(2), 405 (2021)
13. Monfort, M., et al.: Moments in time dataset: One million videos for event understanding. IEEE Transactions on Pattern Analysis and Machine Intelligence **42**(2), 502–508 (2020)
14. Rastgoo, R., Kiani, K., Escalera, S.: Sign language recognition: A deep survey. Expert Systems with Applications **164**, 113794 (2021)
15. Saunders, B., Camgoz, N.C., Bowden, R.: Continuous 3d multi-channel sign language production via progressive transformers and mixture density networks. International journal of computer vision **129**(7), 2113–2135 (2021)
16. (WHO), W.H.O.: Ear and hearing care. https://www.who.int/europe/news-room/questions-and-answers/item/ear-and-hearing-care (January 2023), (Accessed on 07/09/2024)