Real-time Sign Language Recognition and Multimodal Translation

Si Qiang Wu Yang, Aitor Perez Vela, Mikhail Gromov, and Jafar Saniie
Embedded Computing and Signal Processing (ECASP) Research Laboratory (http://ecasp.ece.iit.edu/)
Department of Electrical and Computer Engineering
Illinois Institute of Technology, Chicago, IL, 60616

***Abstract*— Sign languages are vital for an estimated 466 million users globally, yet significant communication barriers remain, particularly in virtual environments. This paper proposes a lightweight, multi-modal computer vision system for real-time American Sign Language (ASL) translation into text and speech. The system processes local webcam feeds to extract body keypoints, utilizing a hybrid recurrent neural network (RNN) architecture that combines Bidirectional LSTM and GRU layers to capture complex gesture dynamics. A custom dataset of 3,780 samples was developed to train and validate the model. Experimental results demonstrate a testing accuracy of 96.83% and real-time performance with a model size of only 3.24 MB (740,031 parameters). This work offers a highly efficient solution for accessible communication, with potential for integration into video conferencing and VR environments**.

Keywords— Sign language; Multimodal translation; Computer Vision; Recurrent Neural Network (RNN); real-time; video feed

I. INTRODUCTION

Sign languages are rich, complex visual languages that convey meaning through manual gestures, facial expressions, and body movements. They serve as the primary means of communication for many deaf and hard-of-hearing individuals worldwide. Historical evidence of sign language dates back to the fifth century BC, notably in Plato's Cratylus [1]. Today, an estimated 466 million people experience disabling hearing loss, underscoring the global importance of sign languages in daily communication. Despite this widespread reliance, significant barriers remain between sign language users and the hearing community, particularly in digital and virtual environments. These challenges limit accessibility, inclusion, and effective interaction, highlighting the need for improved technologies and systems that better support sign language communication [2].

While American Sign Language (ASL) is one of the most widely used languages in North America, Deaf individuals often face challenges in real-time communication due to a lack of accessible, high-speed translation tools[3]. The development of automated Sign Language Recognition (SLR) systems faces two primary technical hurdles: The requirement for temporal analysis to capture the dynamic nature of gestures[4], as opposed to simple static image classification, and the need for computational efficiency to allow models to run locally on devices with limited resources (e.g., laptops, mobile systems) without relying on high-latency cloud processing. This paper presents a multi-modal computer vision algorithm capable of real-time translation from a standard webcam feed. Our contribution includes the creation of a diverse, 63-class dataset and the implementation of a hybrid Bidirectional LSTM-GRU architecture. This system achieves high accuracy with a minimal memory footprint (3.24 MB), facilitating seamless integration into daily communication. By delivering a highly accurate (96.83%), lightweight, and real-time ASL translation system, the proposed work directly addresses practical deployment constraints while advancing accessible, scalable communication technologies.

The architecture of our sign language recognition system follows a modular pipeline designed for low-latency execution. It integrates real-time video acquisition, spatial feature optimization, and a sequential deep learning architecture. As shown in Figure 1, the system converts consecutive video frames into a series of spatio-temporal keypoints. This mapping effectively filters out environmental noise and background clutter, focusing the model's attention exclusively on the human pose. This structured sequence serves as the primary input for our recurrent layers, enabling the capture of long-term dependencies within each sign language gesture.

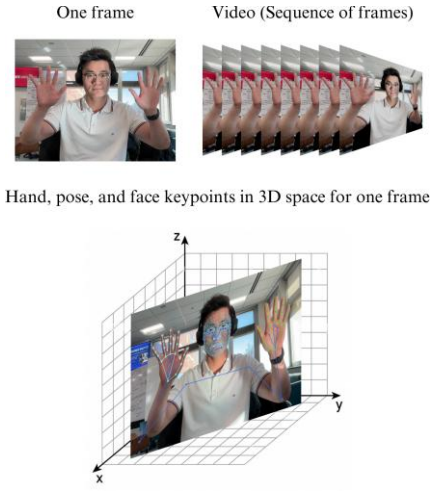
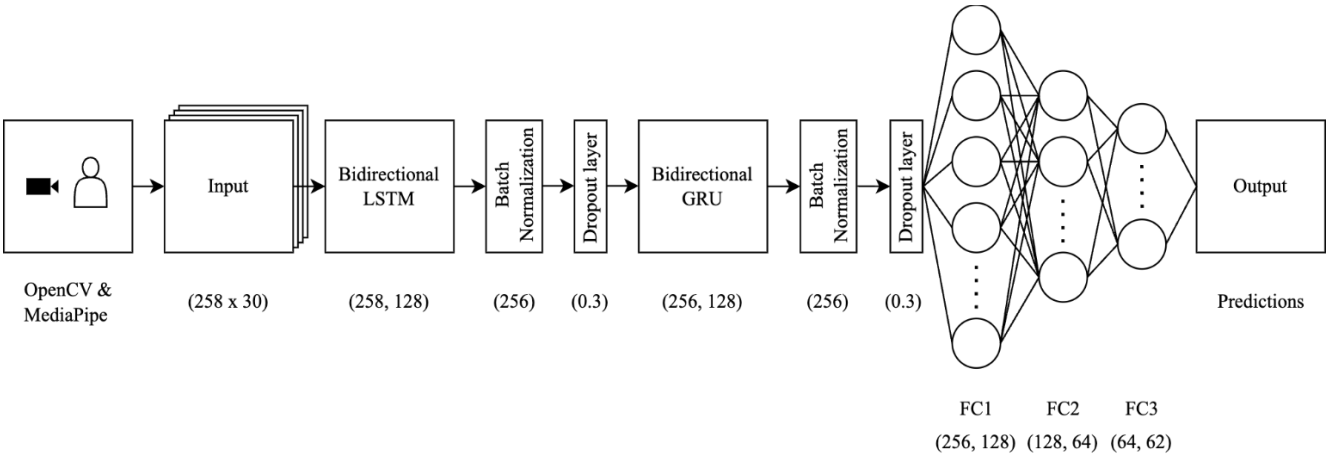


Fig. 1. Sequence of frames and keypoints of hands and body in 3D space using MediaPipe framework

979-8-3315-6026-3/26/$31.00 ©2026 IEEE

II. ASL Recognition Method Based on LSTM-GRU Architecture

The proposed architecture follows a modular pipeline specifically engineered for high-throughput and low-latency execution on consumer-grade hardware to successfully translate ASL [5][6]. To address the limitations of existing sign language recognition frameworks, the system integrates real-time video acquisition, spatial feature optimization, and a sequential deep learning model based on a hybrid recurrent neural network (RNN)[7][8]. As illustrated in the overarching pipeline[9] in Figure 2, this hybrid approach capitalizes on the complementary strengths of Bidirectional Long Short-Term Memory (Bi-LSTM) and Bidirectional Gated Recurrent Unit (Bi-GRU) layers. Specifically, the Bi-LSTM[10] components capture complex, long-term temporal dependencies inherent in dynamic gestures, while the Bi-GRU layers provide the computational efficiency required for real-time inference. This integration is designed to strike an optimal balance between categorical accuracy and system performance, facilitating a seamless end-to-end translation process from raw video feed to multimodal output.



.

Fig. 2. Architecture Diagram Depicting the Dat Flow though the ASL Translation Pipeline, from Video Input to Guesture Classification

A. Data Collection

Following an extensive review of publicly available sign language datasets, it was determined that none provided both the necessary vocabulary breadth and the dynamic recording conditions required for a robust real-time system. Consequently, a custom corpus of 3,780 video samples was curated, encompassing 63 classes: 26 alphabet letters, 14 numerical signs, and 23 common conversational phrases. To ensure environmental robustness, samples were recorded under heterogeneous conditions, including varying illumination, camera angles, and subject orientations.

The vocabulary was specifically selected to cover essential communication:

- **Conversational Phrases:** Includes common interactions such as *"Hello," "Thank you," "How are you? "* and identity markers like *"Deaf"* or *"Hearing."*
- **Alphabet and Numeric Signs:** Covers letters A–Z and numbers 1–20. Notably, numbers 0, 6, and 9 were intentionally excluded due to their high visual similarity to letters O and W, reducing categorical ambiguity.

A critical phase in optimizing the system for edge computing was the selection of the most relevant spatial features. While holistic tracking initially provided 1,662 landmarks per frame, facial landmarks accounted for 84.5% of this input data while contributing marginally to the classification accuracy of the target ASL vocabulary. To maximize computational efficiency and reduce overhead, the input dimensionality was optimized by isolating only the manual and postural coordinates. This streamlined extraction process, which facilitates a 6x reduction in data complexity, is illustrated in Figure 3.

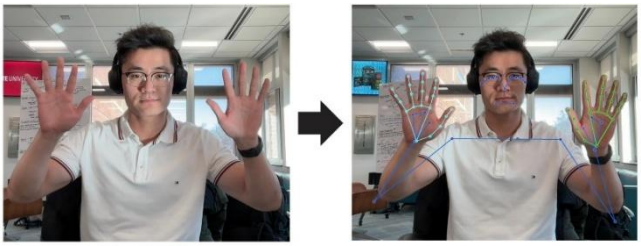


Fig. 3. Keypoint Extraction Stage of Image Preprocessing.

B. Data Processing

The preprocessing pipeline was designed to transform raw video data into structured, high-dimensional tensors suitable for recurrent neural network training. This process involved feature extraction, stochastic data augmentation, and dataset partitioning.

For each video sequence, spatial features were extracted using the MediaPipe[11][12][13] holistic model at a temporal resolution of 30 frames per second. The resulting feature set has 258 elements per frame, consisting of:

- **Postural Dynamics:** 33 body landmarks including (x, y, z) coordinates and a visibility metadata field, totaling 132 features.

Identify applicable funding agency here. If none, delete this text box.

- **Manual Kinematics:** 21 landmarks for each hand (left and right) recorded in (x, y, z) space, contributing 21* 3 * 2 = 126 features.

Each processed sample was structured into a spatio-temporal tensor of shape (30, 258), representing a one-second window of gesture activity. These sequences were normalized and stored as NumPy arrays before being mapped to their respective categorical labels for supervised learning.

To improve model generalization and mitigate overfitting due to the limited size of the initial corpus, a stochastic augmentation pipeline was implemented. Each original sequence underwent four random transformations to introduce realistic spatial and environmental variability:

1) Rotation: Applied to (x, y) coordinates within a range of($\pm 10°$).

2) Scaling: Uniform scaling factors sampled from [0.9, 1.1] to simulate varying camera distances.

3) Spatial Shifting: Translational offsets applied to the (x, y) plane up to ± 0.1

4) Gaussian Noise: Additive zero-mean noise with a standard deviation of $\sigma = 0.01$ to account for sensor precision errors.

This augmentation expanded the custom dataset from 3,780 original samples to a robust training set of 23,010 sequences.

C. Hybrid RNN Architecture

The core classification engine is built upon a Hybrid Bidirectional Recurrent Neural Network (RNN) architecture, specifically designed to leverage the complementary strengths of different gated cell types for dynamic gesture recognition. The detailed configuration, layer dimensions, and parameter distribution of this sequential pipeline are presented in Table 1, illustrating an architecture optimized for both predictive depth and computational efficiency.

- **Bidirectional LSTM Layer:** The initial stage utilizes 128 Bi-LSTM[14] units (397,312 parameters) to extract global temporal context. By processing sequences in both forward and backward directions, the model captures long-term dependencies from past and future states simultaneously. This is critical for disambiguating signs with similar hand shapes that rely on specific motion trajectories to convey meaning.
- **Bidirectional GRU Layer:** A subsequent layer of 128 Bi-GRU[15] units (296,448 parameters) acts as a temporal refiner. GRUs provide a more compact parameter space than LSTMs, reducing the risk of overfitting while effectively refining the sequential features extracted in the previous stage.
- **Comparison to Unidirectional Models:** While unidirectional layers are simpler, bidirectional processing was selected to ensure the system captures the full context of a gesture within the 30-frame window, which is essential for the nuances of ASL grammar and syntax.
- **Classification Head:** The architecture concludes with three Fully Connected (FC) layers (32,896, 8,256, and 4,095 parameters, respectively). These layers employ ReLU activation to introduce non-linearity, mapping the learned embeddings into the 63 target class probabilities.

To ensure stability during training and high generalization across different users, the recurrent stages are interleaved with Batch Normalization and Dropout (p=0.2). This hierarchical integration results in a highly efficient model with a total of 740,031 trainable parameters and a compact memory footprint of 3.24 MB, facilitating high-performance deployment on resource-constrained edge devices.

TABLE I. MODEL SUMMARY: ARCHITECTURE AND PARAMETERS

Layer (type:depth-idx)	Output Shape	Param #
SignLanguageModel	[1, 63]	–
LSTM: 1-1	[1, 30, 256]	397,312
BatchNorm1d: 1-2	[1, 256, 30]	512
Dropout: 1-3	[1, 30, 256]	–
GRU: 1-4	[1, 30, 256]	296,448
BatchNorm1d: 1-5	[1, 256, 30]	512
Dropout: 1-6	[1, 30, 256]	–
Linear: 1-7	[1, 128]	32,896
Dropout: 1-8	[1, 128]	–
Linear: 1-9	[1, 64]	8,256
Linear: 1-10	[1, 63]	4,095
Total params		740,031
Trainable params		740,031
Non-trainable params		0
Total mult-adds (M)		20.86
Input size (MB)		0.03
Forward/backward pass size (MB)		0.25
Params size (MB)		2.96
Estimated Total Size (MB)		3.24

D. Model Evaluation

This section presents the performance of the proposed sign language recognition system through both training metrics and real-time testing. Quantitative results are reported in terms of accuracy, loss, and inference time, while qualitative performance is assessed based on real-time[16] usage and generalization across different individuals.

Using 258 keypoints (hand and pose landmarks) per frame, the model was trained over 35 epochs. Training loss was defined as the average error between predicted values and ground-truth labels, providing a measure of how well the model fits the training data. To monitor generalization, both training and testing metrics were tracked across all epochs (Figure 4). This allowed us to detect any signs of overfitting, which typically appear as a widening gap between training and testing curves. In our case, the loss and accuracy curves for both datasets followed each other closely, stabilizing after approximately 20 epochs. This behavior indicates effective generalization and training stability.

The system achieved very high training and testing accuracies, 99.89% and 99.07% respectively, while completing training in ~35 minutes (2084.51 seconds). This confirms the model's ability to converge efficiently without compromising predictive performance. Batch Normalization and Dropout

layers were crucial: Batch Normalization facilitated stable and rapid convergence, while Dropout enhanced generalization to unseen data. Additionally, the model operated at 54.19 FPS, enabling predictions approximately every 0.018 seconds per sample, which makes it highly suitable for real-time applications. Below, in Table II, the class-wise evaluation results are reported, showing the model's performance for each sign with an average of 75 testing samples per class.

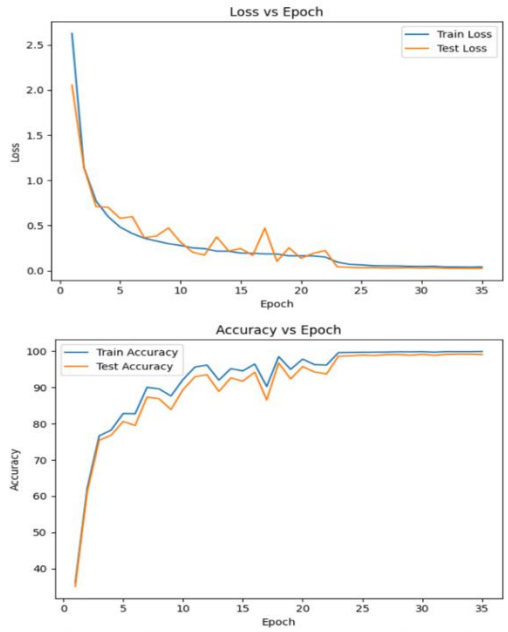


Fig. 4. Loss and Accuracy Curves over 35 Epochs.

E. Per-Class Accuracy of the Model

The detailed breakdown of the model's accuracy per class is summarized in Table II. As shown, the system maintains high performance levels across all 'Words & Phrases,' while also exhibiting strong mean accuracy for the 'Numbers' and 'Alphabet' categories.

To evaluate the generalizability of our model, we performed an additional test using a public Kaggle dataset, WLASL (Word-Level American Sign Language). This dataset was not used for training, but exclusively for independent testing. We selected only the signs overlapping with our custom dataset to ensure a fair comparison. For compatibility with our model, each video was processed with MediaPipe[14] Holistic to extract 258 keypoints[13] per frame (33 pose landmarks with visibility + 21 left-hand + 21 right-hand). Sequences were normalized to a fixed length of 30 frames: shorter videos were padded with zero vectors, while longer ones were truncated. The processed data was stored in the same flattened format as our training set, enabling direct evaluation with the existing pipeline. As shown in Table III, performance is slightly lower than on our own test set due to differences in recording conditions (e.g., camera, lighting, signers). Nevertheless, the model achieved perfect accuracy on several classes (busy, goodbye, hard of hearing, help, hungry, tired) and maintained high performance on most of the vocabulary.

TABLE II. PER-CLASS BINARY CLASSIFICATION ACCURACY ON OUR DATASET

Class	Accuracy (%)
bad	98.61
busy	100.00
deaf	98.61
excuse me	100.00
fine	100.00
good	97.30
good morning	100.00
good night	100.00
goodbye	96.15
hard of hearing	100.00
hearing	100.00
hello	100.00
help	98.61
How are you?	100.00
hungry	98.73
I love you	100.00
Nice to meet you!	98.61
please	98.75
See you later	100.00
sorry	98.75
thank you	97.50
tired	100.00
What's your name?	100.00
1-20	Mean 99.40
A-Z	Mean 97.16

TABLE III. PER-CLASS BINARY CLASSIFICATION ACCURACY ON PUBLIC DATASETS

Class	Accuracy (%)
bad	96.67
busy	100.00
deaf	98.48
fine	98.15
good	85.00
goodbye	100.00
hard of hearing	100.00
hearing	97.92
hello	75.00
help	100.00
hungry	100.00
please	88.10
sorry	76.19
thank you	88.10
tired	100.00

III. Conclusion

We developed a lightweight system capable of real-time sign language recognition and translation, centered on a hybrid recurrent neural network trained on a curated dataset of 63 signs. Through data augmentation, the dataset was expanded to 23,010 labeled sequences, enabling robust training and improving generalization across users. The final model achieved 99.07% test accuracy and operated at 54 FPS (0.018s per sample), demonstrating its suitability for real-time communication on consumer-grade hardware. Beyond the custom dataset, the model was also evaluated on a publicly available ASL dataset (WLASL, Kaggle). Despite differences in recording conditions, it maintained strong performance, confirming its generalization capacity.

The developed sign language recognition system is customizable and reconfigurable, and can be adapted to expand the vocabulary, integrate context-aware language models for sequential interpretation, and embed the system into real-time platforms such as video conferencing. By combining recognition with speech synthesis, the system paves the way for multimodal translation and greater accessibility. Overall, the approach was deliberately optimized for efficiency and inclusivity, with the ultimate goal of empowering underserved populations and fostering more inclusive digital interactions.

References

[1] "Sign Language" Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/Sign_language

[2] World Health Organization. "Deafness and hearing loss". [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss

[3] Hand Talk. "Deaf people in the digital world: learn their main challenges and how to overcome them." [Online]. Available: https://www.handtalk.me/en/blog/deaf-people-challenges/

[4] National Deaf Children's Society. "The three major barriers deaf people face in the community." [Online]. Available: https://www.ndcs.org.uk/deaf-child-worldwide/our-blog/the-three-major-barriers-deaf-people-face-in-the-community/

[5] Wikipedia. "American Sign Language." [Online]. Available: https://en.wikipedia.org/wiki/American_Sign_Language

[6] Commission on the Deaf and Hard of Hearing, State of Rhode Island. "American Sign Language" Social impact of profound hearing loss. [Online]. Available: https://en.wikipedia.org/wiki/Social_impact_of_profound_hearing_loss

[7] IEEE Conference publication | IEEE Xplore. "Network and Deep Learning." [Online]. Available: https://doi.org/10.1109/TENCON.2018.8650524

[8] Donahue, J.; Hendricks, L.A.; Guadarrama, S.; Rohrbach, M.; Venugopalan, S.; Saenko, K.; Darrell, T. "Long-Term Recurrent Convolutional Networks for Visual Recognition and Description." [Online]. Available: https://ieeexplore.ieee.org/document/7298878

[9] Proceedings of the 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) [Online]. Available: https://doi.org/10.1109/

[10] Samaan, G.H.; Wadie, A.R.; Attia, A.K.; Asaad, A.M.; Kamel, A.E.; Slim, S.O.; Abdallah, M.S.; Cho, Y.-I. "MediaPipe's Landmarks with RNN for Dynamic Sign Language Recognition." [Online]. Available: https://doi.org/10.3390/electronics111

[11] Cooper, H.; Ong, E.-J.; Pugeault, N.; Bowden, R. "Sign Language Recognition Using Sub-Units." Journal of Machine Learning Research 2012, 13, 2205–2231. [Online]. Available: https://ai.google.dev/edge/mediapipe/solutions/guide.

[12] Google AI. "MediaPipe solutions guide." [Online]. Available: https://ai.google.dev/edge/mediapipe/solutions/guide.

[13] Siva, L. Mr. Pose: "An easy guide for pose estimation with Google's MediaPipe." Medium. [Online]. Available: https://logessiva.medium.com/an-easy-guide-for-pose-estimation-with-googles-mediapipe-a7962de0e944.

[14] Vennerød, C.B.; Kjærran, A.; Bugge, E.S. "Long Short-Term Memory RNN." *arXiv* 2021, arXiv:2105.06756. [Online]. Available: https://doi.org/10.48550/arXiv.2105.06756.

[15] Rana, R. "Gated Recurrent Unit (GRU) for Emotion Classification from Noisy Speech." *arXiv* 2016, arXiv:1612.07778. [Online]. Available: https://doi.org/10.48550/arXiv.1612.07778

[16] Kartynnik, Y.; Ablavatski, A.; Grishchenko, I.; Grundmann, M. "Real-Time Facial Surface Geometry from Monocular Video on Mobile GPUs." *arXiv* 2019, arXiv:1907.06724. [Online]. Available: https://doi.org/10.48550/arXiv.1907.06724.