Signify: A Real-Time Sign to Text and Text to Sign Mobile Application for Dynamic Filipino Sign Language Translation using Transformer Architecture Deep Learning Model

Aliyah Ayco
Angeles University Foundation
Angeles City
Pampanga, Philippines

Kaye Anne Mirador
Angeles University Foundation
Angeles City
Pampanga, Philippines

Glaiza Mei Natividad
Angeles University Foundation
Angeles City
Pampanga, Philippines

Noah Andrea Pagba
Angeles University Foundation
Angeles City
Pampanga, Philippines

James Esquivel
Angeles University Foundation
Angeles City
Pampanga, Philippines

ABSTRACT

This study presents Signify, a real-time, bidirectional mobile application for dynamic Filipino Sign Language (FSL) translation designed to bridge communication gaps between the Deaf and Hard of Hearing (DHH) community and hearing individuals. Utilizing Long Short-Term Memory (LSTM) and Transformer architectures, the system enables Sign-to-Text (S2T) and Text-to-Sign (T2S) functionalities. To improve model robustness, the researchers expanded the FSL-105 dataset by adding a "Directions" category and recording 80 additional videos per gesture, resulting in a total of 11,530 videos. For S2T recognition, hand landmarks were extracted via MediaPipe. Comparative analysis revealed that the Transformer model significantly outperformed the LSTM baseline, achieving a test accuracy of 98.73%. This was further improved to 99.60% through data augmentation techniques including Gaussian noise injection and temporal jitter. The T2S module utilizes a direct mapping approach to retrieve pre-recorded FSL video segments validated by a certified interpreter for linguistic accuracy. Integrated into an Android application using TensorFlow Lite, the system supports real-time, offline inference. User usability testing yielded a Grand Overall Mean of 4.57 (Excellent), reflecting high satisfaction among signers and non-signers. This research advances inclusive communication in alignment with SDGs 4 and 10.

General Terms

Artificial Intelligence, Deep Learning, Computer Vision, Mobile Computing, Pattern Recognition, Human-Computer Interaction

Keywords

Filipino Sign Language (FSL), Transformer Architecture, Long-Short Term Memory (LSTM), MediaPipe, Real-time Translation.

1. INTRODUCTION

Sign language is a complete linguistic system essential for cognitive development, social integration, and education within the Deaf and Hard of Hearing (DHH) community [1]. It serves as a vital bridge between the DHH and hearing communities. However, overcoming communication barriers remains challenging, as most hearing individuals do not understand sign language [2].

Recent studies have explored vision-based Sign Language Recognition (SLR) systems utilizing deep learning. For instance, combining MediaPipe for hand tracking and Long Short-Term Memory (LSTM) networks has yielded up to 99% accuracy for American Sign Language (ASL) alphabets [3]. While previous studies predominantly focused on ASL, recent models have addressed real-time dynamic Filipino Sign Language (FSL) recognition via mobile applications [4]. However, these systems often cover limited vocabularies and primarily focus on one-way recognition (sign-to-text), failing to support the natural, two-way communication required by DHH individuals. Advanced dual-learning transformer models have demonstrated pose-to-text and text-to-pose capabilities [5], but integrating these robust architectures into accessible, real-time mobile applications remains a rare achievement.

In response to the limitations in existing technologies, this study developed a real-time, two-way mobile application for FSL translation. It implements a comparative analysis between LSTM and Transformer deep learning architectures to improve real-world usability and S2T performance, alongside a robust T2S module mapping text to validated sign representations.

2. RELATED LITERATURE

2.1 Dataset Augmentation and Limitations

Data collection challenges in SLR systems include variations in hand appearance, backgrounds, and lighting conditions, making diverse data collection vital for model robustness [8]. Existing systems trained on datasets like FSL-105 demonstrate high real-time accuracy but are restricted by small, fixed vocabulary categories [4]. Expanding vocabulary and dynamically recognizing both static and dynamic gestures require rigorous data scaling [7].

2.2 Sign-to-text Recognition Models

Studies combining LSTM networks with MediaPipe have established strong baselines for dynamic FSL recognition [9]. Similar LSTM architectures have effectively recognized ASL [3], Indian Sign Language [10], and Czech Sign Language [11] by extracting spatio-temporal features. However, Transformer networks are increasingly utilized for their superior sequence modeling. Studies exploring Transformer networks for isolated SLR demonstrate that multimodal Transformers often outperform traditional LSTM models by efficiently handling global dependencies in complex gestures [12], offering faster learning and processing for real-time applications [13].

2.3 Text-to-Sign Generation Models

T2S translation has been implemented through dual neural machine translation models, converting text into visual sign representations bypassing gloss annotations [14]. Other studies emphasize temporal feature fusion in video frames utilizing BERT and ResNet-50 models [15]. Additionally, generative systems using Generative Adversarial Networks (GANs) and OpenPose have shown high motion accuracy in translating text to sign glosses [16], [17]. Conversely, for rapid mobile processing, deterministic direct-mapping techniques using pre-rendered avatars or recorded databases processed via Natural Language Processing (NLP) ensure expressive, low-latency sign language output without heavy computational overhead [18], [19].

2.4 Mobile Application Integration

Integrating complex SL models into mobile applications necessitates lightweight frameworks. Utilizing TensorFlow Lite or Google's MediaPipe reduces computational complexity, allowing high-precision inference directly on edge devices [14], [20]. This allows real-time two-way translation bridging NLP tokenization and gesture detection entirely offline, highly effective for daily communicative scenarios [21], [22].

3. METHODOLOGY

3.1 Data Acquisition and Augmentation

The primary dataset used was FSL-105 [6], comprising 4-second video clips of 105 labeled FSL gestures. To ensure a comprehensive evaluation across varying scenarios and enrich vocabulary, the researchers augmented the dataset by introducing a new "Directions" category containing 10 specific signs: Right, Left, North, South, West, East, Up, Down, Forward, and Backward. To improve model generalization, the dataset was expanded by recording 80 additional videos per sign across different users, maintaining 640x360 resolution and 30 frames per second (fps). This resulted in a baseline dataset of 11,530 videos.

3.2 Data Preprocessing and Feature Extraction

To standardize input for the Transformer model, all captured video sequences were resized to a resolution of 640x360 pixels. To maintain temporal consistency, sequences were fixed at 120 frames, corresponding to a 4-second duration at 30fps. For feature extraction, the system utilized MediaPipe to identify 21 key landmarks per hand. By focusing on the x and y coordinates of these points, the raw video data was transformed into a structured input shape of (120, 84), representing the temporal flow of hand coordinates across the sequence.

3.3 Keypoints Augmentation and Training Hyperparameters

To enhance the model's robustness against real-world variability, two primary augmentation techniques were applied: Gaussian Noise Injection (with a standard deviation of 0.01) to simulate sensor noise, and Temporal Jitter (shifting sequences by a maximum of 12 frames) to account for variations in sign speed. The augmentation resulted to 69, 162 samples. The Transformer model was trained using the AdamW optimizer with a learning rate of 0.0003. Training was conducted for up to 350 epochs, utilizing an early stopping mechanism that restored weights from the optimal epoch—typically around epoch 70 for the final augmented dataset—to prevent overfitting.

3.4 Dataset Partitioning

The augmented dataset was partitioned to ensure robust model training and evaluation. A standard splitting ratio of 70% for training, 15% for validation, and 15% for testing was employed. This distribution was managed using the train_test_split() utility from the scikit-learn library, which facilitated a stratified approach to maintain class balance across all 105 Filipino Sign Language (FSL) gestures.

3.5 Sign-to-Text Model Development

To present a detailed comparative analysis, two distinct architectures were developed and evaluated:

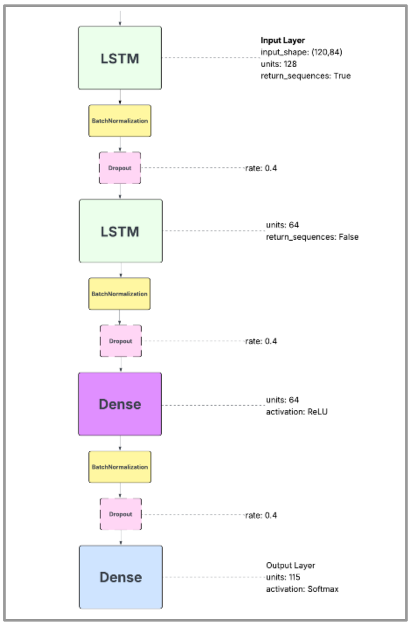

Figure 1. LSTM Model Architecture

As illustrated in Figure 1, the network processed the (120, 84) input through an initial LSTM layer (128 units, return sequences), followed by Batch Normalization and 0.4 Dropout for regularization. A second LSTM layer (64 units) summarized the sequence into a final hidden state, which was then mapped via dense layers to a 115-unit Softmax output for final gesture classification.

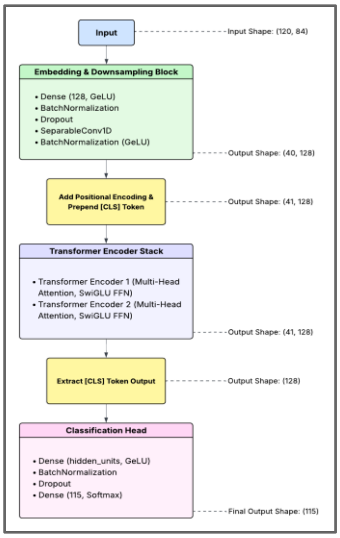
Input
Input Shape: (120, 84)
Embedding & Downsampling Block
• Dense (128, GeLU)
• BatchNormalization
• Dropout
• SeparableConv1D
• BatchNormalization (GeLU)
Output Shape: (40, 128)
Add Positional Encoding & Prepend [CLS] Token
Output Shape: (41, 128)
Transformer Encoder Stack
• Transformer Encoder 1 (Multi-Head Attention, SwiGLU FFN)
• Transformer Encoder 2 (Multi-Head Attention, SwiGLU FFN)
Output Shape: (41, 128)
Extract [CLS] Token Output
Output Shape: (128)
Classification Head
• Dense (hidden_units, GeLU)
• BatchNormalization
• Dropout
• Dense (115, Softmax)
Final Output Shape: (115)

Figure 2. Transformer Model Architecture

The Transformer architecture, shown in Figure 2, projects input features into a 128-dimensional latent space. A separable convolutional layer downsamples the sequence to preserve salient motion while reducing computational load. After integrating sine positional encodings and a learnable [CLS] token, the data passes through two Transformer encoder blocks. Each block utilizes 8 multi-head self-attention heads and a SwiGLU Feed-Forward Network with a hidden dimension of 256. This architecture prioritizes global dependencies, allowing the model to focus on critical hand trajectories over environmental clutter.

3.6 T2S Module Architecture

The T2S component bypassed predictive generation to ensure zero-latency mobile execution on resource-constrained hardware. It employed a direct mapping approach linking English and Tagalog text inputs to a structured JSON lookup file. This system retrieves standardized 1920x1080, H>264 encoded FSL video segments from a library of “proper signs” validated by a certified interpreter from the Persons with Disability Affairs Office (PDAO).

3.7 System Requirements and Experimental Environment

To ensure the real-time performance of the Signify application, the following system environment was established for testing and deployment:

Table 1. Minimum and Recommended system Requirements

Specification	Minimum Requirements	Recommended Requirements
Chipset	Qualcomm Snapdragon 660 / MediaTek Helio G95 / Exynos 9611 or equivalent	Snapdragon 888 / Tenor G2 / MediaTek Dimensity 9000 / Exynos 2200 or equivalent
RAM	6 gb	8 gb or higher
Available Storage	200 mb	500 mb or higher
Android Version	Android 10 (Q)	Android 12 (S) or higher
Camera	8 MP rear or front, 30 fps	12 MP or higher, 60 fps

- Hardware Requirements: The application requires a minimum of a Qualcomm Snapdragon 660 chipset and 6 GB of RAM. For optimal performance involving concurrent MediaPipe tracking and Transformer-based inference, a Snapdragon 888 chipset with 8 GB of RAM is recommended.
- Software Specifications: The system is compatible with Android 10 (API level 29) or higher. Development was conducted using Python 3.x for model training with TensorFlow Lite and the Flutter framework for the mobile interface.
- Camera Specifications: A camera resolution of at least 8 MP at 30 fps is necessary to maintain the 120-frame input window effectively.

3.8 Mobile Application Deployment of S2T and T2S

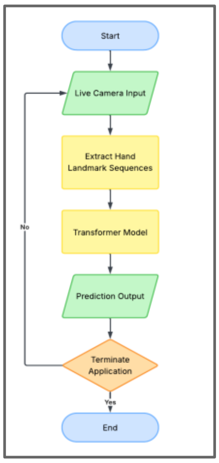

Figure 3. S2T Data Processing Flow

The live S2T processing pipeline is designed for high-performance, on-device execution. As shown in Figure 3, the system continuously captures a live camera feed and utilizes the MediaPipe framework to extract hand landmark sequences. These structured features are fed into the Transformer-Encoder model for real-time gesture classification. This streamlined data flow ensures that the predicted FSL translation is displayed as text with minimal latency, providing a seamless user experience.

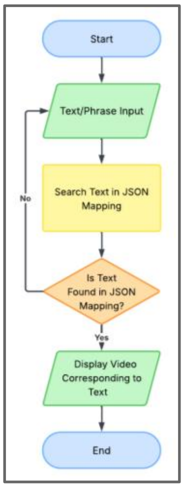
Start
Text/Phrase Input
Search Text in JSON Mapping
Is Text Found in JSON Mapping?
No
Yes
Display Video Corresponding to Text
End

Figure 4. T2S Data Processing Flow

The logic for the T2S module follows a deterministic retrieval-based approach, as depicted in Figure 4. Upon receiving a text or phrase input in either English or Tagalog, the system queries a structured JSON mapping file that serves as a lookup table. If a corresponding gesture is found, the application retrieves and plays the pre-recorded, expert-validated FSL video segment. This direct-mapping strategy ensures absolute linguistic fidelity and zero-latency performance on mobile hardware.

3.9 Evaluation Scenarios and Deployment

The S2T models were optimized for mobile edge environments via TensorFlow Lite. The Transformer model utilized FlexDelegate to ensure compatibility with specialized operations and the XNNPACK delegate to accelerate CPU computation. Comprehensive evaluations were conducted using varied datasets (Expert Signers vs. Non-Signers) and environmental scenarios (front vs. back mobile cameras) to assess real-world usability and the impact of lighting on pose extraction

4. RESULTS AND ANALYSIS

4.1 Comparative S2T Model Analysis

The performance of the S2T engine was evaluated by comparing the baseline LSTM model against the proposed Transformer-Encoder architecture. As shown in Table 2, the Transformer model significantly outperformed the LSTM across all key metrics, particularly after data augmentation.

Table 2. Comparative Performance Metrics of the Best LSTM and Transformer Models

Model	Training Accuracy	Test Accuracy	Training Loss	Test Loss
LSTM (Baseline)	96.18%	93.29%	0.1105	0.2351
Transformer (Initial)	99.74%	98.73%	0.0067	0.0784
Transformer (Augmented)	99.71%	99.60%	0.0073	0.0097

The results indicate that while the LSTM provided a solid baseline, it struggled with the complexity of dynamic FSL gestures. The Augmented Transformer achieved a near-perfect test accuracy of 99.60%, proving that the self-attention mechanism is superior at identifying hand landmark trajectories.

4.2 Statistical Validation

To ensure the superiority of the Transformer was not due to chance, an independent-samples t-test (two-tailed, $\alpha = 0.05$) was conducted. The Transformer (Mean = 98.58%, SD = 0.13) significantly outperformed the LSTM (Mean = 91.87%, SD = 1.28), with $t(4) = -9.03$, $p < 0.01$. This statistical evidence confirms that the self-attention mechanism is more effective than recurrent units for FSL translation.

4.3 Error Analysis and Matrix Findings

Error matrix analysis highlighted distinct architectural behaviors:

- **LSTM Weaknesses**: The baseline model consistently struggled with "Months" (e.g., *December, February*), demonstrating an inability to capture subtle temporal nuances in hand speed.
- **Transformer Performance:** The Transformer's rare errors were strictly confined to minimal gesture pairs with high spatial similarity (e.g., "*Correct*" vs. "*Wrong*").

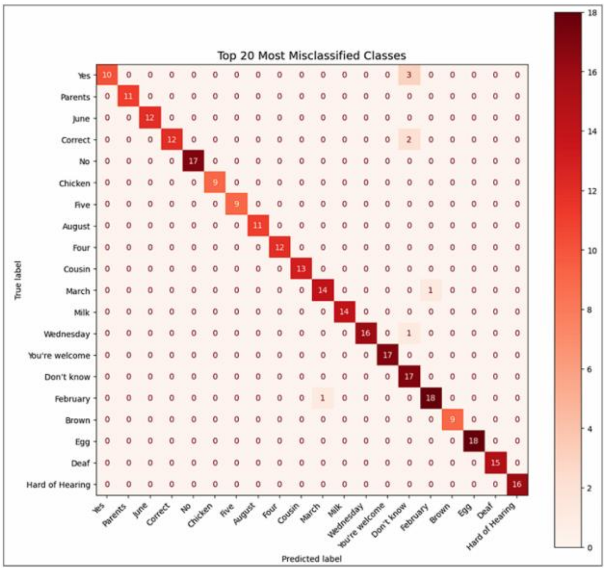

Figure 5. Top 20 most misclassified classes from the final Transformer model

4.4 Technical Challenges and Observed Limitations

Despite the high performance of the Transformer architecture, specific technical hurdles were identified during the evaluation phase. A primary challenge involves "overlapping hands," which occurs during complex gestures where the hands cross or obscure one another. This was notably observed in the phrase "Ikinagagalak kong makilala ka" (Nice to meet you), where the proximity of the hands can cause landmark tracking interference. The Transformer's self-attention mechanism allows it to maintain focus on the dominant hand's trajectory even during these temporary occlusions, enabling more robust sequence modeling compared to recurrent architectures.

Furthermore, the system's accuracy is influenced by external environmental factors, including suboptimal lighting conditions and variations in camera angles. The researchers also noted challenges with "minimal pairs"—signs that possess nearly identical hand shapes or movements—which require the model to discern highly subtle temporal differences to ensure an accurate translation.

4.5 Data Augmentation and Real-World Robustness

To strengthen the system against diverse environments, the best Transformer architecture underwent a secondary training phase on an extensively augmented dataset of 69,162 samples. The following techniques were applied:

1. **Gaussian Noise Injection (SD = 0.01):** Successfully simulated sensor inaccuracies typical of mid-range mobile cameras.
2. **Temporal Jitter (±12 frames)**: Simulated natural variations in human signing speeds, ensuring the model remains robust during fast-paced conversation.

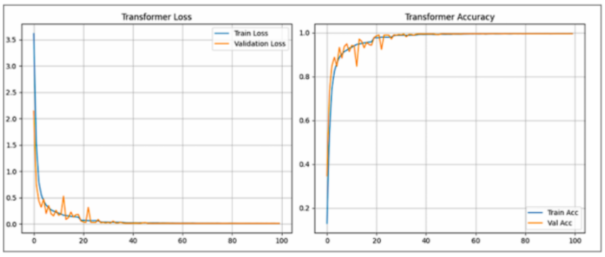

Figure 6. Training and validation loss and accuracy curves of the Final Augmented Transformer model

This comprehensive protocol improved the final test accuracy to 99.60% and a remarkably low test loss of 0.0097, completely resolving previous misclassifications of directional minimal pairs.

4.6 Qualitative Validation of the T2S Module

Unlike the S2T model, the T2S module utilizes a deterministic retrieval system rather than a predictive model. This design choice ensures zero-latency performance and high linguistic fidelity on mobile hardware by mapping text inputs directly to an indexed library of expert-validated FSL videos via a JSON lookup system.

The module's efficacy was established through a qualitative validation process conducted by a professional Sign Language Interpreter certified by the PDAO. The results of this expert review are summarized below:

- **Linguistic Accuracy:** The expert issued a formal 'Certification of Validation', attesting that all 115 FSL signs within the application's dataset are "proper signs".
- **Cultural Appropriateness:** The evaluation confirmed that the signs selected represent the most widely used and common forms of FSL, ensuring they are broadly recognizable and comprehensible within the Filipino Deaf community.
- **Natural Variation:** While acknowledging inherent regional or stylistic variations in FSL, the expert affirmed that the system's repertoire remains linguistically sound and appropriate for cross-demographic communication.

4.7 System Latency and Real-Time Performance

To meet the requirements of a real-time mobile tool, the final Transformer model was optimized via TensorFlow Lite. Utilizing the XNNPACK delegate for CPU acceleration, the system achieved an average inference speed of 30ms to 45ms per sequence. This ensures that the translation is displayed at a rate compatible with the 30 fps camera input, providing a seamless "live" experience for the user.

4.8 Live Implementation and Usability Testing

To evaluate the real-world readiness of the Signify application, a comprehensive live test was conducted on a mobile device. The researchers performed five trials per gesture, utilizing both front and back cameras to verify the hand landmark model's performance. The live gesture recognition achieved an overall accuracy of 90.96%, with individual class accuracies ranging from 10% to 100 %. Observations during these trials indicated that the application encountered difficulties with certain directional signs and colors; specifically, "Down" achieved 20% accuracy, while "North" scored 30% due to misinterpretation as "gray." Additionally, "Green" was frequently misidentified as "white," resulting in an accuracy of 50%. These tests were conducted under optimal conditions to establish a baseline for technical performance.

Following the technical trials, the user usability of the application was rigorously assessed using a five-point Likert scale. This evaluation involved a panel of nine strategic testers: three expert Signers from the Angeles Elementary School Special Education Faculty and six Non-Signers from the Angeles University Foundation College of Education, specializing in Special Needs Education. The assessment focused on three major criteria: Perceived Ease of Use, Performance and Accuracy, and User Satisfaction.

Table 3. Summary of Overall Usability Assessment by User Category

Category	Overall Mean Score	Interpretation
Signers	3.92	Very Good
Non-signers	4.90	Excellent
Overall	4.57	Excellent

The application demonstrated a high level of usability, yielding a Grand Overall Mean (GOM) of 4.57, interpreted as "Excellent." As shown in Table 2, a comparative analysis revealed that Non-Signers rated the application significantly higher (4.90) than the expert Signers (3.92).

Table 4. Detailed Usability Assessment by Evaluative Criteria

Usability Criteria	Mean Score	Interpretation
Perceived Ease of Use	4.67	Excellent
Performance and Accuracy	4.39	Very Good
User Satisfaction	4.78	Excellent

A criterion-specific breakdown, detailed in Table 4, provides further insight: Perceived Ease of Use achieved a mean of 4.67 (Excellent), indicating an intuitive interface; Performance and Accuracy registered a mean of 4.39 (Very Good); and User Satisfaction received the highest score of 4.78 (Excellent), confirming a high degree of user contentment and a high probability of recommendation.

Despite these positive quantitative results, qualitative feedback from the expert signers highlighted areas for enhancement. Specifically, signers recommended refining the sign lexicon to prioritize authentic signs commonly used by the Deaf community to ensure better linguistic fidelity. Furthermore, feedback indicated that while the S2T feature is satisfactory, it requires further tuning to improve the recognition of signs essential for everyday communication.

5. CONCLUSION AND RECOMMENDATIONS

This study effectively addresses communication barriers by deploying Signify, a real-time, bidirectional FSL translator on a mobile edge environment. Detailed comparative analysis confirmed that Transformer architectures (98.73% baseline accuracy) significantly outperform traditional LSTM models (93.29%) in dynamic FSL recognition ($p < 0.01$). Furthermore, subjecting the Transformer to extensive evaluation scenarios utilizing Gaussian noise and temporal jitter augmentations elevated field-ready accuracy to 99.60%, successfully resolving ambiguities in directional minimal pairs. The combination of an optimized S2T Transformer and a deterministic, expert-validated T2S mapping module resulted in a highly accessible offline application that achieved an "Excellent" (4.57) usability rating across both Deaf expert signers and hearing non-signers.

To further enhance the research, it is recommended to conduct extensive data collection encompassing varied environmental scenarios (low lighting, extreme angles) to mitigate camera-sensor failures. Future iterations should focus on integrating non-manual signals—such as facial expressions and head orientations—to capture the full grammatical depth of FSL. Additionally, transitioning from isolated word recognition to continuous sentence-level translation and integrating Generative AI (e.g., GANs) for fluid, avatar-based sign rendering would further improve the naturalism of the T2S module.

6. ACKNOWLEDGMENTS

The authors extend their gratitude to the faculty of the College of Computer Studies at Angeles University Foundation, the Persons with Disability Affairs Office (PDAO), and the Special Education Department of Angeles Elementary School for their invaluable expertise and validation.

7. REFERENCES

[1] Hand Talk. 2023. The Benefits of Sign Language for Children with Hearing Loss. Hand Talk - Learn ASL today.

[2] Tolentino, L. K., Serfa Juan, R., Thio-ac, A., Pamahoy, M. A., Forteza, J. R. and Garcia, X. J. 2019. Static Sign Language Recognition Using Deep Learning. International Journal of Machine Learning and Computing 9(6), 821-827.

[3] Sundar, B. and Bagyammal, T. 2022. American Sign Language recognition for alphabets using MediaPipe and LSTM. Procedia Computer Science 215, 642-651.

[4] Canlas et al. 2024. Real-time dynamic Filipino Sign Language recognition application. Angeles University Foundation.

[5] Chaudhary, L., Ananthanarayana, T., Hoq, E. and Nwogu, I. 2023. SignNet II: A transformer-based two-way sign language translation model. IEEE Transactions on Pattern Analysis and Machine Intelligence 45(11), 12896-12907.

[6] Tupal, J. 2023. FSL-105 Dataset. Mendeley Data repository.

[7] Evangelista, C. L. L., Geli, C. J. R., Castillo, M. M. V. and Macabagdal, C. B. G. 2023. Long Short-Term Memory-based Static and Dynamic Filipino Sign Language Recognition. IEEE.

[8] Nerlekar, A. 2021. Sign Language Recognition Using Smartphones. California State University.

[9] Caya, M. V. C., Madrid, G. K. R. and Villanueva, R. G. R. 2022. Recognition of Dynamic Filipino Sign Language using MediaPipe and Long Short-Term Memory. 13th ICCCNT.

[10] Kothadiya, D., Bhatt, C., Sapariya, K., Patel, K., Gil-González, A.-B. and Corchado, J. M. 2022. Deepsign: Sign Language Detection and Recognition Using Deep Learning. Electronics 11(11), 1780.

[11] Do Long, V. 2021. Mobile Application for Sign Language Recognition. Czech Technical University.

[12] De Coster, M., Van Herreweghe, M. and Dambre, J. 2020. Sign language recognition with transformer networks. LREC, 6018-6024.

[13] Abdul, W., Alsulaiman, M., Amin, S. U., Faisal, M., Muhammad, G., Albogamy, F. R. and Ghaleb, H. 2021. Intelligent real-time Arabic sign language classification using attention-based inception and BiLSTM. Computers & Electrical Engineering 95, 107395.

[14] Ananthanarayana, T. 2021. A Comprehensive Approach to Automated Sign Language Translation. Rochester Institute of Technology.

[15] Wei, S. and Lan, Y. 2023. A two-way translation system of Chinese sign language based on computer vision. arXiv preprint.

[16] Stoll, S., Camgoz, N. C., Hadfield, S. and Bowden, R. 2020. Text2Sign: Towards sign language production using neural machine translation and generative adversarial networks. International Journal of Computer Vision 128(6), 891–908.

[17] toll, S., Hadfield, S. and Bowden, R. 2020. Signsynth: Data-driven sign language video generation. ECCV, 353-370.

[18] Latkar, H., Wasker, S., Vashistha, A. and Kanse, A. 2024. Real-time conversion of sign language to text and speech, and vice-versa. TIJER 11(4).

[19] Faisal, M., Alsulaiman, M., Mekhtiche, M., Abdelkader, B. M., Algabri, M. and Alrayes, T. B. S. 2023. Enabling two-way communication of deaf using Saudi sign language. IEEE Access 11, 135423-135434.

[20] Kavana and Suma. 2022. Real-time sign language recognition.

[21] Manoj Kumar, D., Bavanraj, K., Thavananthan, S., Bastiansz, G. M. A. S., Harshanath, S. M. B. and Alosious, J. 2020. EasyTalk: A translator for Sri Lankan Sign Language using machine learning. ICAC.

[22] Saleem, M. I., Siddiqui, A., Noor, S., Luque-Nieto, M. A. and Otero, P. 2022. A novel machine learning based two-way communication system for deaf and mute. Applied Sciences 13(1), 453.

[23] Ramadhan, M. F., Samsuryadi, S. and Primanita, A. 2024. American sign language translation to display the text (subtitles) using a convolutional neural network. EMACS Journal 6(3).

[24] Osman et al. 2020. Hearing Assistive Technology: Sign Language Translation Application for Hearing-Impaired Communication.

[25] Kautsar et al. 2024. Transformer-based sequence models.