GESTURA: A real time Gesture to Speech Interpreter

Archana V. Bhamare, Aditya Gaikwad, Chinmay Bhandare, Harshal Dharmapurikar
Pimpri Chinchwad College of Engineering, Pune, India

Abstract

This work presents a novel approach to sign language translation by integrating Electromyography (EMG) signals with Natural Language Processing (NLP) to enhance contextual understanding. The primary challenge addressed is the communication barrier faced by individuals who are deaf or hard of hearing, particularly the limited ability of non-sign language users to comprehend sign language. Current systems often require both parties to understand sign language, which is not always feasible. The proposed solution involves the development of a real-time machine learning model, portable device that translates hand gestures into text, thereby facilitating clear communication. The literature survey highlights the reliability of EMG-based systems in capturing muscle activity, while also acknowledging their limitations, such as signal noise and the need for precise Gesture recognition. By incorporating NLP, the system aims to improve the contextual accuracy of translations, overcoming the shortcomings of existing gesture recognition systems. The expected outcome is a robust assistive communication tool that significantly enhances interaction for individuals with communication disorders, ultimately contributing to the field of assistive technology.

Keywords — Sign language translation, Electromyography (EMG), Natural Language Processing (NLP), Gesture recognition, Machine learning

Introduction

In today's digital age, human-computer interaction (HCI) plays a pivotal role in shaping user experiences across various domains, from virtual reality gaming to assistive technologies for individuals with disabilities. Traditional input methods such as keyboards and mice have limitations in terms of intuitiveness and accessibility. As such, there is a growing demand for more natural and immersive interfaces that can bridge the gap between humans and machines seamlessly.

These gloves are designed to capture intricate hand movements and translate them into actionable commands, revolutionizing the way we interact with computers and digital environments.

The objectives for this work is to accurately detect finger movements, map these specific movements to corresponding text, and develop a machine learning model capable of classifying input gestures with precision. Additionally, the work aims to convert the generated text into voice signals to enhance accessibility. The overarching goal is to create a portable, cost-effective, and user-friendly system that can perform real-time sign language translation, making it both practical and efficient for users who need instantaneous communication assistance.

Literature Review

Sr. No.	Title	Publisher	Year	Methodology	Conclusion
1	Sign Language Interpreter Using Machine Learning	IEEE	2024	Combined wearable flex sensors and a microcontroller to collect gesture data; implemented supervised machine learning to classify gestures into corresponding text.	Showcased the feasibility of real-time gesture interpretation, improving accessibility for individuals unfamiliar with sign language.
2	Real-time Sign Language Recognition using Machine Learning and Neural Network	IEEE	2023	Used neural networks to process input signals from multiple sensors, including accelerometers and gyroscopes, and trained the model with labeled gesture data.	Provided a scalable and efficient solution for dynamic gesture recognition, applicable in wearable technology.
3	Sign Language Prediction using Machine Learning Techniques: A Review	IEEE	2023	Surveyed various ML techniques such as SVM, decision trees, and neural networks, comparing their performance in gesture recognition across different datasets.	Identified challenges in data availability and emphasized the importance of robust ML models to ensure diverse and inclusive recognition systems.
4	Sign Language Recognition using Deep Learning	IEEE	2024	Developed a deep learning pipeline utilizing CNNs to extract spatial and temporal features from gesture images or signal data	Improved recognition accuracy and robustness, demonstrating potential for deployment in real-world scenarios.
5	EMG-Based Gesture Recognition for Sign	IEEE	2022	Collected EMG data using wearable sensors; preprocessed signals to reduce noise and used machine learning for classification	Established reliable methods for using EMG signals to classify gestures,

	Language Interpretation				suitable for assistive technologies.
6	Accelerometer -Based Gesture Classification in Wearable Devices	IEEE	2021	Implemented a wearable device with sensors mounted on specific positions to precisely measure the gestures.	Demonstrated precise motion tracking, laying the foundation for enhanced gesture-based communication devices.

Table 1.1 Review of the previous works

S. Anthoniraj et al. [1] discussed the ability of EMG signals to capture real-time gesture patterns by assessing skeletal muscle activity. In the proposed model, hand gesture recognition accuracy was achieved at high levels in real-time applications. Signal variability and noise during such applications posed a problem, thereby requiring improved signal processing and machine learning techniques to ensure higher consistency.

R. Matlani et al. [2] conducted a study on vision-based systems for sign language recognition, based on image recognition techniques for detecting hand and body gestures. Though non-invasive, they suffer from several drawbacks, which include poor performance in low light conditions, privacy issues, and inability to capture minimal movement. The authors suggested the use of EMG-based systems instead.

D. Aggarwal et al. [3] discussed the portability of EMG-based systems for assistive communication devices, demonstrating their effectiveness in capturing sign language gestures for daily use. However, the lack of sensor positioning and inconsistency of muscle signals restrict their use and there is a need for developing adaptive sensors and strong training algorithms.

D. Kothadiya et al. [4] made use of supervised learning models applied to EMG data improve the accuracy of gesture recognition. However, the contextual unawareness confines their system to interpreting only complex or nuanced gestures; hence, they suggest adding Natural Language Processing to understand the gestures in a wider conversational perspective.

S.K. Singh et al. [5] had focused on EMG-based sign language translation wherein the EMG signals are mapped to particular signs. Even though it has demonstrated that EMG can be useful in sign language translation, some issues are still encountered, such as contextual accuracy and ambiguity resolution. They, too, recommended NLP integration to improve the quality of translations.

Finally, T. Marasović et al. [6] emphasized the importance of contextual awareness in improving gesture recognition accuracy. By integrating NLP techniques with EMG signals, they proposed a system incorporating context mapping, which significantly reduced errors and enhanced performance, particularly during complex conversations.

This integration highlights the potential for creating more accurate and meaningful gesture-based communication systems.

I. Proposed System

The proposed system consists of four key components:

1. Gesture Recognition Module
   - Captures real-time hand movements using EMG sensors and accelerometers.
   - Maps gestures to individual words using a trained ML model.
2. Context-Aware Text Generation Module (T5-Base)
   - Takes the predicted words and paraphrases them into coherent and grammatically correct sentences.
   - Utilizes a Persistent KV Cache Mechanism to retain context across multiple calls.
3. Persistent KV Cache Mechanism *(Proposed Innovation)*
   - Stores the transformer's Key-Value (KV) cache in an external memory buffer (RAM or Flash).
   - Updates this cache dynamically to reuse past context for real-time sentence formation.
4. Text-to-Speech (TTS) Conversion Module
   - Converts the final grammatically corrected sentence into speech.

Methodology

The proposed system follows a structured pipeline for real-time gesture recognition and speech synthesis, integrating hardware-based signal acquisition, machine learning-driven classification, and natural language processing (NLP) for grammatically correct speech output.

1. Hardware Design and Data Acquisition
   The system is built around a custom-designed wearable device equipped with accelerometers and an ESP32 for real-time motion and electromyography (EMG) signal acquisition. The device captures:
   - EMG signals (2 channels per hand)
   - Accelerometer data (6 dimensions per hand)

   This results in a 16-dimensional feature space (8 per hand) that is transmitted via a wired connection to a Raspberry Pi for further processing.

2. Data Processing and Preprocessing Pipeline
   Once received by the Raspberry Pi, the raw data undergoes several preprocessing steps:
   - Noise filtering using bandpass filtering to remove unwanted EMG artifacts
   - Normalization & feature scaling to maintain consistency
   - Dimensionality reduction (if needed) to optimize computational efficiency.

3. Gesture Recognition Using Hybrid Machine Learning Model

The processed 16-dimensional data is then fed into a gesture classification model, which consists of:

- Artificial Neural Network (ANN) trained on the 16D feature set for initial gesture detection.
- Ensemble Learning Approach combining ANN with strong classifiers (Random Forest, Gradient Boosting, SVM) to enhance accuracy and robustness.

Once a gesture is detected, it is mapped to its corresponding text representation.

4. Context-Aware Paraphrasing for Indian Sign Language (ISL)
   Since Indian Sign Language (ISL) sentences often follow a syntactically different structure (e.g., "TONIGHT HOME LATE NOT." instead of "Don't be late coming home tonight."), a custom-trained T5 paraphraser is introduced to generate grammatically correct text. To maintain linguistic coherence, we implement a Persistent Context-Aware KV Cache Mechanism, allowing the paraphraser to retain context across multiple text generations:

   - Persistent KV Cache for Long-Term Context

   Instead of discarding the KV cache after each generation, past key-value states are stored in structured external memory (RAM, NVMe SSD, or Flash storage) for retrieval.

   - Modifying the T5 Generation Process

   The generate() function is modified to accept an external past_key_values parameter.

   When a new word is detected, the system retrieves previous KV states and injects them into the next generate() call, ensuring grammatical and contextual continuity.

   - Efficient KV Storage

   Circular buffer in RAM stores recent KV cache for fast retrieval.

   Periodic Flash memory storage is used for longer retention, ensuring a balance between performance and memory constraints.

   - Rolling Context Window

   Instead of indefinitely storing all tokens, a sliding window mechanism keeps only N past tokens, dynamically pruning older, irrelevant context to prevent excessive memory usage.

5. Emotion-Aware Speech Synthesis
   Simultaneously, a sentiment analysis model evaluates the emotional tone of the paraphrased sentence. Emotional tagging is applied to different parts of the sentence, which then influences the Mini-Parler TTS module to generate human-like speech with appropriate tone, pitch, and cadence.

6. Real-Time Feedback Loop for Adaptive Learning

To improve accuracy over time, the system incorporates a user feedback mechanism, enabling corrections that fine-tune both:

- The gesture recognition model (retraining with new samples)
- The paraphraser and speech synthesis module (adjusting linguistic structures and sentiment mapping)

This ensures continuous adaptation and refinement, making the system more precise and context-aware over repeated usage.

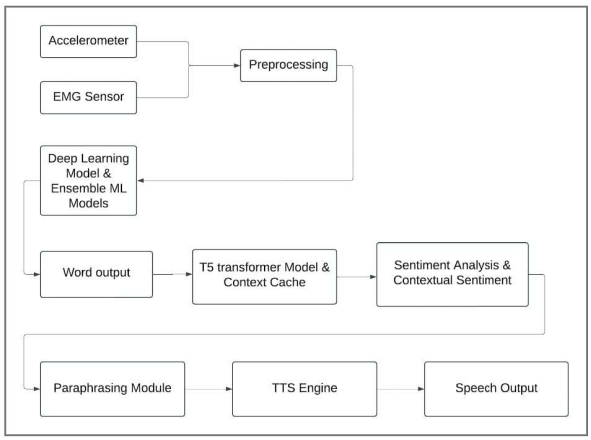

Fig. 1. Block diagram for gesture interpretation and speech generation

Implementation

The implementation of the proposed gesture interpretation system involves a seamless integration of hardware and software components, ensuring real-time processing and accurate translation of hand gestures into meaningful speech. The system is built on a Raspberry Pi platform, where necessary libraries such as Adafruit ADS1x15 for analog EMG signal acquisition, MPU6050 for accelerometer data, and TensorFlow Lite for efficient machine learning inference are installed.

The first step is enabling I²C communication, which facilitates data transfer between sensors and the microcontroller. The EMG sensor and accelerometer are then connected to capture muscle activity and hand movements, respectively. Once the sensors are initialized, signal acquisition begins, where raw data undergoes preprocessing, including noise filtering using band-pass and low-pass filters and normalization for consistency. The preprocessed signals are then segmented using a sliding window approach to create manageable data chunks, which are subsequently used for feature extraction.

Feature extraction plays a crucial role in gesture recognition, where time-domain and frequency-domain features such as root mean square (RMS), variance, and power spectral density are derived from the EMG and accelerometer data. A deep learning

model, primarily an artificial neural network (ANN), is trained alongside an ensemble classifier, incorporating techniques such as Random Forest, to enhance recognition accuracy.

The classified gestures are then mapped to predefined words and transmitted to the next processing stage, where a modified T5-based transformer model performs grammar correction. This transformer model receives words sequentially, updates a context buffer, and dynamically adjusts sentence structure to maintain grammatical coherence and context relevance. The model refines tenses, adds missing words, and restructures the sentence based on an autoregressive approach, ensuring a natural flow of language.

Simultaneously, a sentiment analysis module processes the evolving sentence, extracting emotional cues using a pre-trained sentiment classifier. This module assigns a sentiment score—positive, neutral, or negative—by analyzing embeddings from the grammar correction system. The sentiment score is then passed to the text-to-speech (TTS) engine, which employs an advanced speech synthesis model such as Tacotron 2 or FastSpeech to generate emotion-aware speech output. Based on the sentiment score, the TTS engine modulates tone, pitch, and speed to reflect the user's emotional intent, ensuring an expressive and natural speech synthesis.

Finally, the system incorporates a feedback loop, allowing users to validate and refine gesture recognition accuracy over time. User-specific gestures are incrementally learned, optimizing the model's adaptability and robustness. This comprehensive implementation strategy ensures that the system operates efficiently, providing a real-time, user-friendly solution for converting hand gestures into grammatically and contextually accurate speech.

4.1 Data Flow and Processing Pipeline

1. Gesture Prediction → Initial Word Formation
   - The system predicts words from gestures using EMG sensor input and a trained classification model.
2. Context-Aware Paraphrasing (T5 with Persistent KV Cache)
   - The predicted words are tokenized and fed into the T5 model for paraphrasing.
   - The model generates grammatically correct text using a stored KV cache from previous interactions.
3. KV Cache Storage and Retrieval
   - The KV cache from previous generate() calls is stored in an external buffer.
   - When a new word arrives, the system retrieves and injects stored KV pairs into the next generation call.

4. Text-to-Speech Conversion
    - The final corrected output is converted into speech using Parler-TTS/Coqui-TTS.

4.2 Key Technical Implementations

Modifying the T5 Generation Process to Accept:

1. External KV Cache
    - Modify the generate() function to accept an external past_key_values parameter.
    - Before running generate(), retrieve stored KV cache from memory and inject it into the new input sequence.
2. Efficient KV Storage Mechanism
    - Implement a circular buffer in RAM to store recent KV cache for fast retrieval.
    - For longer retention, periodically store and retrieve KV cache from NVMe SSD or Flash memory.
3. Rolling Context Window
    - Instead of storing all tokens indefinitely, keep a sliding window of past N tokens to avoid excessive memory usage.
    - Ensure old, irrelevant context is pruned dynamically.
4. Optimizing Cache for Low Latency
    - KV cache retrieval should be non-blocking to maintain real-time processing speed. Using CUDA pinned memory for fast GPU access reduces overhead.

Results and Analysis

Feature	**Our Product (EMG+Accelerometer)**	**SignaLoud (Glove-Based)**
Sensor Technology	Uses EMG sensors to capture muscle activity and MPU6050 accelerometers for motion tracking	Uses flex sensors for finger bending and IMUs for motion detection
Wearable Form Factor	Fabric-based bands with Velcro straps for easy attachment on forearm	Gloves that cover the entire hand, possibly restricting natural movement
Communication Medium	Uses ESP32 & Raspberry Pi wired communication like I2C	Uses Bluetooth to transmit data to a smartphone
Data Processing	Uses Machine Learning (ML) model to	Uses predefined sign-to-text

	predict gestures	conversion
Context Understanding	Supports NLP-based translation for better context recognition	No NLP integration, works with a predefined sign-to-speech mapping
Accuracy	Higher accuracy due to muscle activation + movement tracking	Lower accuracy as flex sensors only detect finger bending
Language Flexibility	Can be customized for different sign languages & dialects	Limited to predefined sign language database
Application Scope	Can be used for assistive communication, prosthetics, and smart wearables	Mainly for basic sign-to-text conversion
Latency in Data Transmission	High speed compared to Bluetooth and minimum data loss	Higher latency as Bluetooth communication relies on external devices
Power Efficiency	Optimized power consumption with Raspberry Pi and ESP32 sleep modes	Consumes more power due to continuous Bluetooth transmission
User Comfort	Lightweight & flexible, does not restrict hand movements	Can be restrictive, especially during prolonged use

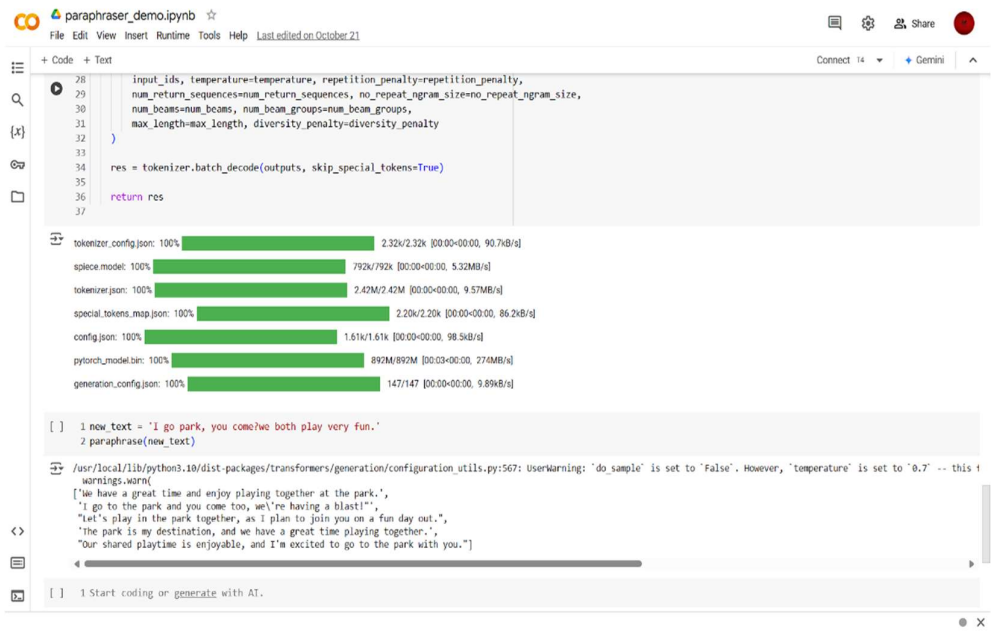

Fig. 2. T5 Model Paraphrasing Output Grammar correction and paraphrasing:

1. A transformer model (humarin / chatgpt_paraphraser_on_T5_base) was chosen because of its fast sequence to sequence transfer with contextual understanding, making it reliable for emotion matching required for speech generation.
2. Below are some demo results, not much refined, as the data is going to be huge.
3. You can see that the above model, helps adding helping verbs and paraphrases the input text.

ANN:

A deep neural network was trained with some hyperparameter tuning namely SGD, RMSProp, Adam as optimizer, etcetera.

```
#SGD #RMSprop #Adam #Adadelta #Adagrad ##Adamax ###Nadam #Ftrl
opt = optimizers.Nadam(lr=1e-3)
model.compile(optimizer = opt,
              loss = "categorical_crossentropy",
              metrics = ["accuracy"])

model.summary()
```

```
Model: "functional_5"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         [(None, 8)]               0
_________________________________________________________________
functional_3 (Functional)    (None, 32)                610016
_________________________________________________________________
dense_10 (Dense)             (None, 8)                 264
=================================================================
Total params: 610,280
Trainable params: 610,280
Non-trainable params: 0
_________________________________________________________________
```

Fig. 3. ANN model summary of parameters

The Image below represents the accuracy and loss of Training and Validation data, which resulted in

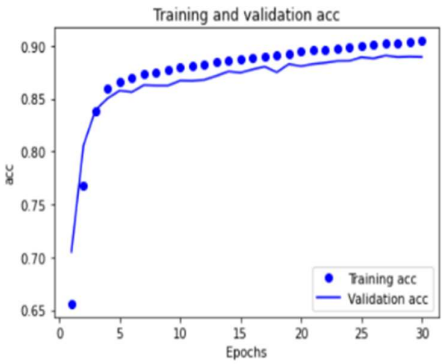

Fig. 4. Training (28.76%) & validation loss (37.28%)

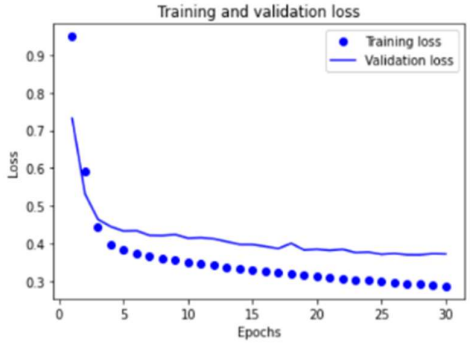

Fig. 6.3.4 Training (90.4%) & validation accuracy (88.94%)

Conclusion

Communication between sign language users and nonusers is greatly improved by the Sign Language Interpreter system, which offers an efficient way to convert sign language motions into text. This system achieves high precision in gesture identification by precisely capturing and interpreting hand gestures using EMG signals. Context aware translations, made possible by the incorporation of Natural Language Processing (NLP), produce output that is both correct and pertinent to the conversational context. The system is a useful, accessible tool that lowers communication barriers and promotes inclusion for people with speech and hearing impairments because of its real-time processing and user- friendly design. This research proposes a Persistent KV Cache Mechanism for real-time gesture-to-speech conversion, allowing context-aware text generation using the T5 model. By retaining past context across multiple generate() calls, the system ensures grammatical correctness and fluency in generated speech.

Future work includes:

1. Exploring Retrieval-Augmented Generation (RAG) to dynamically fetch contextual information.
2. Optimizing cache storage for minimal latency with hardware acceleration.
3. Evaluating on different transformer architectures like GPT-based models.

References

[1] Anthoniraj, S., et al. "Sign Language Interpreter Using Machine Learning." 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA). IEEE, 2021.

[2] Matlani, Roshnee, et al. "Real-time Sign Language Recognition using Machine Learning and Neural Network." 2022 International Conference on Electronics and Renewable Systems (ICEARS). IEEE, 2022.
[3] Aggarwal, D., Ahirwar, S., Srivastava, S., Verma, S., & Goel, Y. (2023, March). Sign Language Prediction using Machine Learning Techniques: A Review. In 2023 Second International Conference on Electronics and Renewable Systems (ICEARS) (pp. 1296-1300). IEEE.
[4] Kothadiya, D., Bhatt, C., Sapariya, K., Patel, K., Gil-González, A. B., & Corchado, J. M. (2022). Deepsign: Sign language detection and recognition using deep learning. Electronics, 11(11), 1780.
[5] Singh, Shashank Kumar, and Amrita Chaturvedi. "A reliable and efficient machine learning pipeline for american sign language gesture recognition using EMG sensors." Multimedia Tools and Applications 82.15 (2023): 23833-23871.
[6] Marasović, Tea, and Vladan Papić. "Accelerometer-based gesture classification using principal component analysis." SoftCOM 2011, 19th International Conference on Software, Telecommunications and Computer Networks. IEEE, 2011.
[7] Chen, Zhigang, et al. "C ${^ 2} $ RL: Content and Context Representation Learning for Gloss-free Sign Language Translation and Retrieval." arXiv preprint arXiv:2408.09949 (2024).
[8] Kudrinko, Karly, et al. "Wearable sensor-based sign language recognition: A comprehensive review." IEEE Reviews in Biomedical Engineering 14 (2020): 82-97.
[9] Sriharsha, A. V., et al. "An Adaptive Learning Method for Sign Language Detection." 2024 5th International Conference for Emerging Technology (INCET). IEEE, 2024.
[10] De Fazio, Roberto, et al. "Human–machine interaction through advanced haptic sensors: a piezoelectric sensory glove with edge machine learning for gesture and object recognition." Future Internet 15.1 (2022): 14.
[11] Wen, Feng, et al. "AI enabled sign language recognition and VR space bidirectional communication using triboelectric smart glove." Nature communications 12.1 (2021): 5378.
[12] Balaji, Pranav, and Manas Ranjan Prusty. "Multimodal fusion hierarchical self-attention network for dynamic hand gesture recognition." Journal of Visual Communication and Image Representation 98 (2024): 104019.
[13] Kim, Jae-Myeong, Min-Gu Kim, and Sung-Bum Pan. "Study on noise reduction and data generation for sEMG spectrogram-based user recognition." Applied Sciences 12.14 (2022): 7276.
[14] Kankipati, Dileep, et al. "tinyRadar for Gesture Recognition: A Low-power System for Edge Computing." 2023 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS). IEEE, 2023.
[15] Alzubaidi, Mohammad A., Mwaffaq Otoom, and Areen M. Abu Rwaq. "A novel assistive glove to convert arabic sign language into speech." ACM Transactions on Asian and Low-Resource Language Information Processing 22.2 (2023): 1-16.