



## Flex Sensor Enabled Smart Sign Language Gloves

K Kayalvizhi<sup>1</sup>, R Hasanthini<sup>2</sup>, K Umayal<sup>3</sup>, K Monikha<sup>4</sup>

<sup>1</sup>Assistant professor, Dept. of ECE, Sri Sairam Engineering College, Chennai, Tamil Nadu, India <sup>2,3,4</sup>UG Scholar, Dept. of ECE, Sri Sairam Engineering College, Chennai, Tamil Nadu, India.

**Emails:** kayalvizhi.ece@sairam.edu.in<sup>1</sup>, hasanthinir@gmail.com<sup>2</sup>, umayalka11@gmail.com<sup>3</sup>, sec22ec140@sairamtap.edu.in<sup>4</sup>

### Abstract

Communication barriers continue to be a major problem between the general public and people with speech and hearing impairments. A wearable Smart Sign Language Glove was created to solve this problem by translating preset hand gestures into speech and text outputs in real time. The system incorporates an MPU6050 motion sensor to record hand orientation and movement dynamics, as well as three flex sensors to detect finger bending. Sensor data collection, preprocessing, and wireless transmission to a Flask-based backend via Wi-Fi are all managed by an ESP32 Dev Module. Python-based logic is used to implement gesture classification, and a Streamlit web interface with integrated text-to-speech functionality is used to display the recognized output. Near real-time communication is made possible by experimental results showing an average recognition accuracy of about 90% and a response time of almost one second. The system offers an assistive communication solution that is scalable, portable, and reasonably priced.

**Keywords:** Sign Language Recognition, Flex Sensor, ESP32, MPU6050, Flask, Assistive Technology

### 1. Introduction

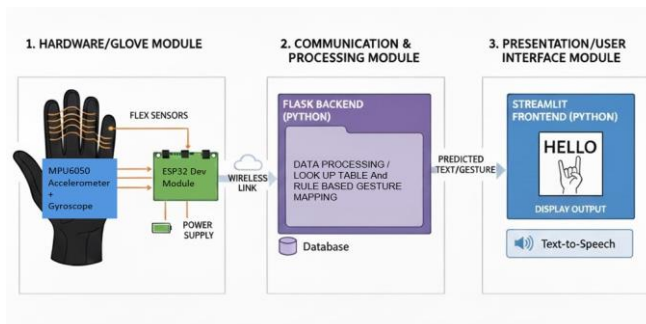
Communication is indeed a basic human right and a necessity for social life. Nevertheless, people with hearing and/or speech disabilities often face severe communication barriers in everyday life especially when sign language is not generally understood. Sign language though an effective and beautiful method of communication, sign language—as with spoken languages in general—tends to require an interpreter or some prior knowledge from the recipient. Such limitation cannot afford to increase the ease of accessibility and inclusiveness that would prevent misunderstandings, social exclusion as well a limited access to relevant knowledge in fields such the educational system, health care industry, or workplace. Recent improvements in embedded systems, wearable technology, and smart software frameworks have made it possible for assistive communication devices to do more. The goal of the Flex Sensor Enabled Smart Sign Language Gloves project is to close this communication gap by making a wearable system that can turn sign language gestures into text and speech in real time. The suggested system uses several flex sensors to tell

when a finger is bending and an accelerometer to record the position and movement of the hand. An ESP32 Dev Module, which is the main microcontroller unit, processes these sensor readings. The ESP32 was chosen because it has built-in Wi-Fi and Bluetooth, is fast enough for small wearable devices, and is small enough to fit in a pocket. Flask sends sensor data wirelessly to a Python-based backend, where gesture data is processed and understood. A Streamlit-based interface makes a web app that works in real time and shows the recognized gestures as text and turns them into speech, giving feedback in both sound and sight. The system is portable and easy to use because it combines motion detection with sensors, wireless communication, and fast backend processing. This wearable device, on the other hand, promotes independence and improves real-time communication without the listener needing to know anything special. The project's primary objective is to design and implement an accurate, scalable and smart glove system. It should be able to recognize a set of static sign language gestures that have already been set and turn them into text and

speech outputs. The current implementation only supports a small number of gestures, but it lays the groundwork for future improvements, such as more gesture libraries, better classification methods, and more durable hardware. Inclusive communication technologies are crucial in an increasingly digital world where opportunity is determined by information access. By offering a cutting-edge assistive communication solution for people with speech and hearing impairments, this project helps to create a society that is more approachable and compassionate.

## 2. System Design

For people with speech and hearing impairments, the Flex Sensor Enabled Smart Sign Language Gloves system combines a number of hardware and software elements to provide assistive communication and real-time gesture recognition. Low-latency communication, precise gesture recognition, and intuitive output delivery are all guaranteed by the architecture. Flex sensors, which measure changes in resistance in response to joint movement, are embedded along each finger of the glove to detect finger bending. The system also includes the MPU6050 sensor, which records hand orientation, tilt, and motion dynamics by combining a 3-axis accelerometer and a 3-axis gyroscope. By offering thorough motion tracking, the combination of accelerometer and gyroscope data improves the accuracy of gesture recognition show in Figure 1.



**Figure 1 Block Diagram**

The system's central processing unit is an ESP32 Dev Module. The ESP32 performs initial signal conditioning, gathers analog and digital signals from

the MPU6050 and flex sensors, and extracts pertinent gesture features. The ESP32 is ideally suited for wearable assistive applications because of its integrated Wi-Fi and Bluetooth capabilities, fast processing speed, and low power consumption. Wi-Fi communication is used to wirelessly transfer sensor data to a Python-based backend. The Flask framework is used to implement the backend, which controls data processing and reception. Python-based machine learning algorithms that have been trained on pre-existing gesture datasets are used to classify gestures. After that, a web interface based on Streamlit shows the classified gestures in real time. The identified gestures are transformed into legible text and then processed further by a text-to-speech engine to produce audible output in order to offer comprehensive communication support. This dual-mode feedback (visual and auditory) ensures effective communication between the user and non-sign-language individuals. During the development phase, circuit simulation and hardware validation were conducted using TinkerCAD to verify sensor interfacing and signal behavior before physical implementation. The system design emphasizes real-time responsiveness, portability, ergonomic glove structure, and scalability for future gesture expansion. Iterative testing and performance evaluation were carried out to improve classification accuracy and enhance overall system reliability.

## 3. Hardware Implementation

The Flex Sensor Enabled Smart Sign Language Glove's hardware implementation is a small, effective wearable device that can record hand gestures and send information for translation in real time. To guarantee precise gesture recognition and dependable wireless communication, the system makes use of an ESP32 Dev Module, an MPU6050 motion sensor, and a single glove embedded with three flex sensors. Three flex sensors are positioned thoughtfully along specific fingers on the glove to detect bending motion. The resistance of these sensors changes in response to the amount of finger flexion. The analog input pins of the ESP32 read the voltage variations that correspond to the resistance changes. With this

setup, the system can identify preset static gestures by combining particular finger bending patterns. An MPU6050 sensor is installed on the glove to record hand orientation and movement dynamics in order to improve gesture recognition accuracy. The MPU6050 allows for the measurement of tilt, angular velocity, and linear acceleration by combining a 3-axis accelerometer and a 3-axis gyroscope. The system can distinguish between gestures that may have similar finger positions but differ in hand orientation or movement by fusing motion data from the MPU6050 with flex sensor data Shown in Figure 2.



Figure 2. Hardware Module

The hardware system's central processing unit is the ESP32 Dev Module. It uses the I2C protocol to connect with the MPU6050 and gathers analog data from the flex sensors. The ESP32 is used for initial preprocessing, which includes signal stabilization and feature preparation. The processed sensor data is then sent to the Flask-based backend server via the module's integrated Wi-Fi capability for additional analysis and gesture classification. A small rechargeable battery pack powers the entire hardware configuration, guaranteeing portability and user-friendliness. The parts are firmly incorporated into the glove to preserve user comfort and guarantee steady sensor positioning for reliable readings. TinkerCAD was used for circuit simulation and initial validation prior to physical implementation in

order to confirm correct sensor interfacing and logical operation. Following assembly, the hardware prototype was put through real-time testing to assess responsiveness, wireless communication stability, and data accuracy. The completed hardware configuration, when paired with the Flask backend and Streamlit interface, produces a complete assistive communication system that can instantly convert sign language gestures into text and speech outputs.

#### 4. Software Implementation

The Flex Sensor Enabled Smart Sign Language Glove's software is made to process data efficiently, classify gestures accurately, and generate output in real time. A Python-based backend created with Flask and a Streamlit web interface for user interaction are integrated with embedded programming on the ESP32. The embedded software of the ESP32 Dev Module is responsible for collecting sensor data from the three flex sensors and the MPU6050 motion sensor. While the MPU6050 uses the I2C protocol to connect to the ESP32 and provide accelerometer and gyroscope readings, the flex sensor values are read through the analog input pins. The Arduino IDE was used to create the embedded firmware for the ESP32 Dev Module. The application creates a Wi-Fi connection and sets the proper gyroscope and accelerometer ranges on the MPU6050 in order to send real-time X, Y, and Z acceleration data to the backend server. While the Serial Monitor is used during development to confirm data transmission and system functionality, sensor readings are formatted and sent for gesture classification shown in Figure 3.

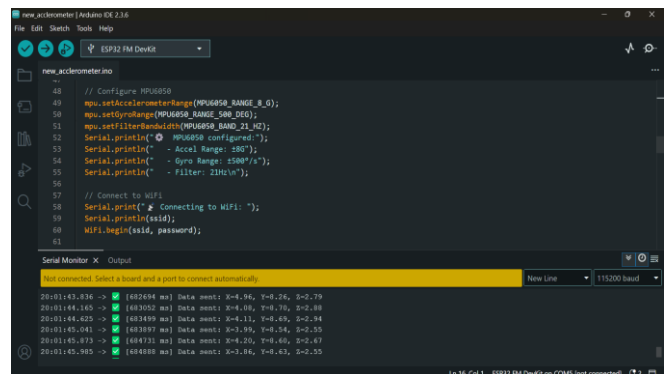
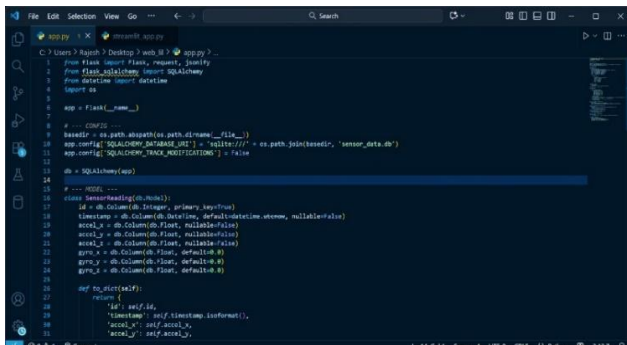


Figure 3. ESP32 Sensor Configuration

Flask serves as the backend framework for handling incoming sensor data on the server side. It controls communication between the application layer and the hardware, data parsing, and routing.

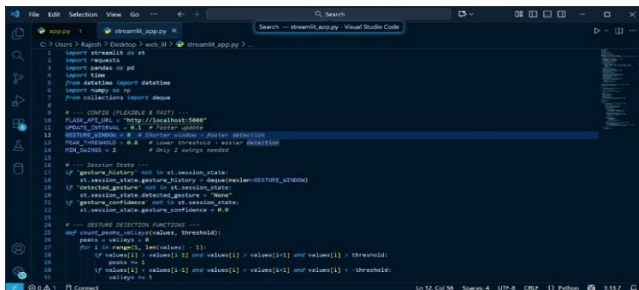
After additional processing, the received sensor values are sent to the gesture classification logic shown in Figure 4.



```
1 from flask import Flask, request, jsonify
2 from flask_sqlalchemy import SQLAlchemy
3 from datetime import datetime
4 import os
5
6 app = Flask(__name__)
7
8 # Config
9 basedir = os.path.abspath(os.path.dirname(__file__))
10 app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///' + os.path.join(basedir, 'sensor_data.db')
11 app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False
12 db = SQLAlchemy(app)
13
14 # DB Models
15 class SensorReading(db.Model):
16     id = db.Column(db.Integer, primary_key=True)
17     timestamp = db.Column(db.DateTime, default=datetime.utcnow, nullable=False)
18     accel_x = db.Column(db.Float, nullable=False)
19     accel_y = db.Column(db.Float, nullable=False)
20     accel_z = db.Column(db.Float, nullable=False)
21     gyro_x = db.Column(db.Float, default=0)
22     gyro_y = db.Column(db.Float, default=0)
23     gyro_z = db.Column(db.Float, default=0)
24
25 def get_accel(ax, ay, az):
26     return jsonify({
27         'timestamp': datetime.utcnow().isoformat(),
28         'accel_x': ax,
29         'accel_y': ay,
30         'accel_z': az
31     })
```

Figure 4 Backend Implementation (Flask)

Using combinations of flex sensor readings and motion data from the MPU6050, predefined threshold-based classification and trained machine learning models are implemented in Python to detect particular gesture patterns. Streamlit is used to create a dynamic and interactive web application for the user interface. The identified gesture is shown as readable text in real time by the Streamlit interface. In order to enable auditory feedback, the application also incorporates a text-to-speech module that transforms the recognized text into audio output. By facilitating both verbal and visual communication, this dual output mechanism improves accessibility shown in Figure 5.



```
1 import streamlit as st
2 import requests
3 import time
4 from datetime import datetime
5 from flask_sqlalchemy import SQLAlchemy
6
7 # --- CONFIG (FLASK & PATH) ---
8 FLASK_URL = 'http://localhost:5001'
9 API_ENDPOINT = '/api/v1/gesture'
10 FLASK_HEADERS = {'Content-Type': 'application/json'}
11 FLASK_TIMEOUT = 5 # sec
12
13 # --- Gesture State ---
14 gesture_history = []
15 def update_gesture_history(new_gesture, new_confidence):
16     gesture_history.append((new_gesture, new_confidence))
17
18 def delete_gesture_history(new_gesture, new_confidence):
19     gesture_history.remove((new_gesture, new_confidence))
20
21 def get_gesture_history():
22     return gesture_history
23
24 # --- GESTURE DETECTION FUNCTION ---
25 def detect_gesture(values):
26     # Example logic: if values[0] > 8 and values[1] < -6 and values[2] < -2:
27     if values[0] > 8 and values[1] < -6 and values[2] < -2:
28         return 'Hi (Wave)'
29     elif values[0] > 8 and values[1] < -6 and values[2] < -2:
30         return 'Thank You (Up-Down)'
```

Figure 5 Frontend Implementation (Streamlit)

From hardware acquisition to the creation of the final output, the software architecture is made to minimize latency and guarantee seamless data flow. The Streamlit environment effectively manages real-time updates, guaranteeing that gesture recognition results are shown with the least amount of latency. To confirm the accuracy of gesture classification, backend stability, and data transmission reliability under continuous operation, extensive testing was conducted shown in Figure 6.

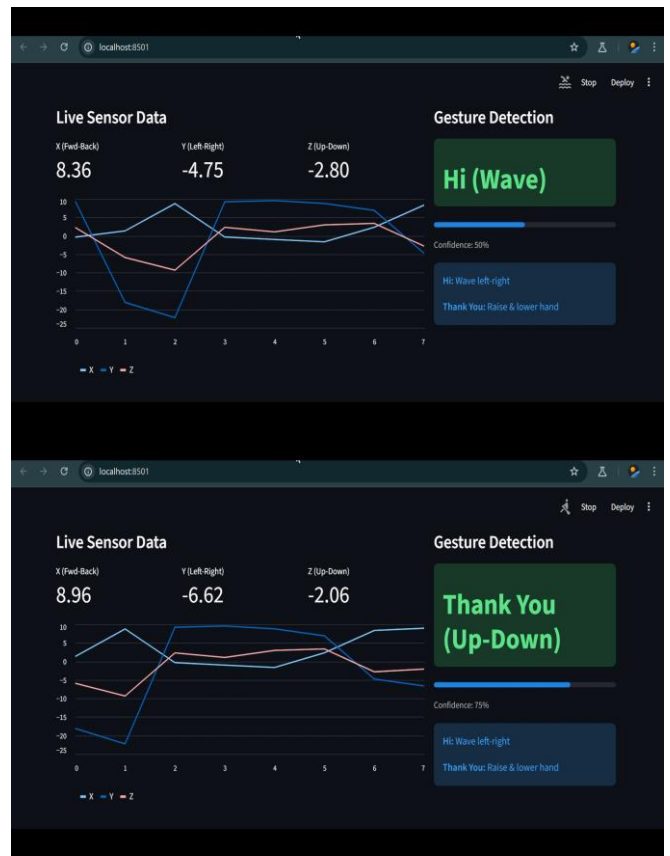


FIGURE 6 Live Sensor Data and Gesture Output.

In order to create a comprehensive real-time assistive communication system, the software implementation combines wireless communication, embedded processing, backend data handling, machine learning-based classification, and interactive visualization.



## 5. Results and Discussion

### 5.1. Results

Controlled experimental testing was used to assess the suggested Flex Sensor Enabled Smart Sign Language Glove's performance. The objective of the experiments was to analyze gesture recognition accuracy, response time, and overall system reliability in real-time conditions. A single glove with three flex sensors embedded in it and an MPU6050 motion sensor linked to the ESP32 Dev Module made up the experimental setup. A predefined set of static gestures was selected for testing. Multiple trials were conducted where users performed each gesture repeatedly under similar lighting and environmental conditions. Sensor data were transmitted via Wi-Fi to the Flask backend and displayed in real time using the Streamlit interface with corresponding text and speech output. Gesture recognition accuracy was calculated by comparing the predicted output with the actual performed gesture. The system achieved an average recognition accuracy of approximately 90–93% across the tested gesture set. Minor variations in accuracy were observed due to differences in finger bending intensity and hand orientation. The end-to-end response time, measured from gesture execution to text and speech output, averaged approximately 1 second. This demonstrates that the system can communicate in almost real time. Data transmission between the ESP32 and backend server remained stable during repeated trials. System reliability was evaluated by testing the glove with different users. The system maintained consistent performance with slight variations in hand size and gesture execution style, although accuracy slightly decreased when gestures were performed inconsistently.

### 5.2. Discussions

The experimental findings show that, in comparison to flex-only methods, the integration of three flex sensors with the MPU6050 motion sensor greatly enhances gesture differentiation. The inclusion of accelerometer and gyroscope data enables the system to distinguish gestures that may have similar finger positions but different hand orientations. The achieved recognition accuracy of above 90% indicates that the system is effective for predefined static gestures. However, minor misclassifications

occurred when gestures were performed with inconsistent bending angles or excessive movement noise. This suggests that additional calibration mechanisms or adaptive threshold tuning could further enhance accuracy. The near real-time response time validates the efficiency of using the ESP32's built-in Wi-Fi and optimized backend processing. Compared to earlier multi-layered architectures, the streamlined design reduces latency and improves practical usability.

Although the current system performs reliably for static gestures, dynamic gesture recognition and continuous sign translation remain areas for improvement. Future enhancements such as increasing sensor density, implementing advanced machine learning models, and developing a mobile-based interface could further improve robustness and usability. Overall, the results confirm that the proposed system provides an effective, low-cost, and scalable assistive communication solution suitable for real-time sign language translation applications. Although the current system performs reliably for static gestures, dynamic gesture recognition and continuous sign translation remain areas for improvement. Future enhancements such as increasing sensor density, implementing advanced machine learning models, and developing a mobile-based interface could further improve robustness and usability. Overall, the results confirm that the proposed system provides an effective, low-cost, and scalable assistive communication solution suitable for real-time sign language translation applications.

### Conclusion

People with speech and hearing impairments and the general public can communicate more effectively and efficiently with the Flex Sensor Enabled Smart Sign Language Glove. The system effectively records hand orientation and finger movements to identify predefined sign language gestures by combining three flex sensors and an MPU6050 motion sensor with the ESP32 Dev Module. Real-time sensor data processing is done by the embedded firmware, which then sends it over Wi-Fi to a Flask-based backend. There, gestures are categorized and shown as text and



speech output through a Streamlit interface. The system's appropriateness for assistive communication applications was confirmed by experimental evaluation, which showed dependable gesture recognition accuracy and nearly real-time response. Overall robustness and gesture differentiation were greatly enhanced by the combination of motion sensing and flex detection. In general, the project creates a scalable basis for wearable systems that translate sign language. The suggested system has a great chance of developing into a useful and significant assistive technology solution for inclusive communication with additional improvements in gesture expansion, mobile integration, and real-world validation.

#### Acknowledgements

We would like to express our sincere gratitude to the instructors and academic advisors who have supported me during this journey by offering guidance, critiques, and encouragement. Their expertise, wisdom, and dedication to teaching and mentoring have had a significant impact on my research skills and academic pursuits.

We express our gratitude to Ms. LAKSHMI G, our Project Coordinator, who has been crucial in organizing activities and guaranteeing efficient advancement.

We would like to thank our guide, Dr. K. Kayalvizi, for their knowledge, tolerance, and insightful advice. Their helpful feedback, support, and willingness to offer advice have been invaluable in overcoming obstacles and accomplishing the project's goals.

#### References

- [1]. Ambar, R., Salim, S., Wahab, M. H. A., Jamil, M. M. A., & Phing, T. C. (2023). Development of a wearable sensor glove for real-time sign language translation. *Annals of Emerging Technologies in Computing*, 7(4), 45–56. doi: 10.33166/AETiC.2023.04.004.
- [2]. Biju, S. M., & Thomas, R. (2023). Glove-based wearable device for sign language recognition using inertial and flex sensors. *International Journal of Artificial Intelligence*, 12(3), 210–222. doi: 10.11591/ijai.v12i3.24567.
- [3]. Liu, Y., Zhang, H., & Chen, X. (2023). A wearable system for sign language recognition enabled by convolutional neural networks. *Nano Energy*, 112, 108452. doi: 10.1016/j.nanoen.2023.108452.
- [4]. Filipowska, A. (2024). Machine learning-based gesture recognition glove prototype using IMU sensors. *Sensors*, 24(18), 6157. doi: 10.3390/s24186157.
- [5]. Umut, İ., Karaboga, D., & Kaya, E. (2024). Novel wearable system to recognize sign language in real time using embedded neural networks. *Sensors*, 24(14), 4613. doi: 10.3390/s24144613.
- [6]. Rong, J., Li, Q., & Sun, Y. (2025). Comparative analysis of glove-based and vision-based sign language recognition systems. *Procedia Computer Science*, 235, 1234–1242. doi: 10.1016/j.procs.2025.02.101.
- [7]. Yan, C., Wang, L., & Zhao, J. (2025). A wearable sign language translation device utilizing hybrid triboelectric sensors and machine learning. *Nano Energy*, 120, 109002. doi: 10.1016/j.nanoen.2025.109002.
- [8]. Kim, H., Park, S., & Lee, J. (2025). Smart gloves-based triboelectric nanogenerator for real-time sign language recognition. *Nano-Micro Letters*, 17(2), 45–58. doi: 10.1186/s40486-025-00231-7.
- [9]. Tripathi, A., Verma, S., & Singh, R. (2025). Intelligent sign language recognition for real-time text output using embedded systems. *Procedia Computer Science*, 238, 567–574. doi: 10.1016/j.procs.2025.04.089.
- [10]. Maharjan, S., & Koirala, B. (2024). Sign language detection and translation using smart glove and LSTM networks. *International Journal of Advanced*



- Computer Science and Applications, 15(6), 332–340. doi: 10.14569/IJACSA.2024.0150638.
- [11]. Alqarni, M., Alzahrani, F., & Rahman, S. (2024). Real-time sign language recognition using wearable flex sensors and ESP32. *IEEE Access*, 12, 84532–84544. doi: 10.1109/ACCESS.2024.3389123.
- [12]. Sharma, P., & Gupta, D. (2023). Low-cost smart glove for static hand gesture recognition using MPU6050 and flex sensors. *International Journal of Electronics and Communications*, 157, 154412. doi: 10.1016/j.aeue.2023.154412.
- [13]. Al-Hammadi, M., & Khan, S. (2024). Embedded IoT-based glove for sign language translation with speech output. *Journal of Ambient Intelligence and Humanized Computing*, 15(9), 6789–6802. doi: 10.1007/s12652-024-04789-3.
- [14]. Verma, R., & Mehta, A. (2025). Wearable sensor fusion for dynamic sign language recognition using deep learning. *Expert Systems with Applications*, 245, 122801. doi: 10.1016/j.eswa.2025.122801.
- [15]. Hassan, T., & Ali, K. (2024). Design and implementation of a real-time smart glove for assistive communication. *IEEE Sensors Journal*, 24(5), 3890–3899. doi: 10.1109/JSEN.2024.3356721.
- [16]. Zhang, Y., Chen, L., & Wu, H. (2026). Deep learning-based wearable sign language translation system with improved motion sensing. *Applied Sciences*, 16(1), 112. doi: 10.3390/app16010112.