Asian Research Association
INTERNATIONAL RESEARCH JOURNAL OF
MULTIDISCIPLINARY TECHNOVATION

DOI: 10.54392/irjmt26415

Wearable Glove: Sign Language Interpretation with AI-Enabled Finger - Mounted Sensors

R. Valarmathi [a], K. Ghousiya Begum [a,*], R.D. Manoj [a]

[a] School of Electrical and Electronics Engineering, SASTRA Deemed to be University, Thanjavur- 613401, Tamil Nadu, India.
* Corresponding Author Email: ghousiyabegum@eie.sastra.edu
DOI: https://doi.org/10.54392/irjmt26415
Received: 14-03-2026; Revised: 29-06-2026; Accepted: 02-07-2026; Published: 18-07-2026

Abstract: Understanding sign language can facilitate communication between deaf people and hearing people. Nevertheless, a considerable communication gap still exists between sign language users and those who are unfamiliar with its gestures. To address this challenge, the present study combines sensor technology with machine learning (ML) algorithms to develop an AI-enabled wearable glove (AIEWG) for real-time American Sign Language (ASL) recognition and text translation. Two MPU6050 inertial measurement units and three flex sensors were used to capture finger-bending movements, hand motions, and wrist orientations. The proposed system recognizes individual letters (A–Z) and expressions such as "hello," "help," "thank you," "me," and "please." The collected sensor readings were preprocessed for machine learning analysis to interpret static gestures. Several ML and DL models were employed, among which the Particle Swarm Optimization (PSO)-tuned Extra Trees classifier (EXT) demonstrated superior performance, with precision and F1-score values of 99.87%, an AUC-ROC of 99.9%, and recall and accuracy of 99.86%. For real-time deployment, the AIEWG captures sensor data, which are then classified using a PSO-tuned EXT model deployed on an ESP32. The trained model is embedded in the firmware as a C-compatible (.h) file to enable real-time inference in the Arduino IDE. The system operates at a sampling frequency of 100 Hz, with each gesture captured in a continuous 3-second window. The recognized gestures are converted into text and transmitted via Wi-Fi to a web server, where they are displayed through an accessible graphical user interface (GUI). This enables uninterrupted interaction between deaf users and people who are unfamiliar with sign language.

Keywords: American Sign Language, Static Gestures, Wearable Glove, Machine Learning, Graphical User Interface.

1. Introduction

Sign language is an essential form of communication for millions of people who are deaf or hard of hearing worldwide. Sign language recognition systems have potential applications in educational resources, public service access, and assistive devices that enable interaction between hearing people and deaf people. This communication gap affects the educational opportunities, social integration, and communication accessibility of members of the deaf community. Signs are the primary mode of communication used by deaf or mute people to express their thoughts and interact with others. Nevertheless, the deaf community often experiences communication failures and social isolation because non-ASL speakers are unable to understand these signs.

Recent advances in wearable technology have expanded this field [1]. Sensor-based sign language recognition systems use strain sensors, pressure or tactile sensors, surface electromyography (sEMG) sensors, and inertial sensors such as magnetometers, accelerometers, and gyroscopes [2]. In such wearable systems, flex sensors and inertial measurement units (IMUs) enable the recording and analysis of hand and arm motions associated with sign language gestures. Gestures can be tracked in real time by placing sensors on the hands or on other wearable devices [3-5].

Many studies in the literature use ML and DL approaches to automate sign language recognition. Videos and images of static hand gestures are identified using Convolutional Neural Networks (CNNs), whereas dynamic motion sequences that change over time are recognized using recurrent neural networks (RNNs). Hybrid frameworks have also been developed to capture both the contours and movement of the hands. This shows that AI techniques can considerably improve sign language recognition (SLR). ML/DL frameworks are applied to process data collected from these sensors in order to identify and interpret sign language gestures [6].

These models enhance the accuracy and efficiency of real-time sign language recognition systems [7]. By training supervised learning algorithms such as Support Vector Machines (SVMs) and using histogram of oriented gradients (HOG) features on labeled datasets, diverse sign gestures can be classified using sensor data [8]. CNNs are used to recognize complex patterns, whereas RNNs identify temporal correlations in sign language sequences [9-11]. ML techniques have been widely explored for gesture classification [12]. Sign language recognition (SLR) approaches can be divided into three groups: computer vision (CV)-based models, wearable sensors, and hybrid systems. Computer vision-based models require image processing algorithms and cameras to categorize gestures [13]. A suitable target is typically located within the field of view, tracked as it moves, and then classified using DL models [14]. Leap Motion sensors and Kinect cameras have also been used to record hand movements for sign language gesture recognition [15]. However, environmental factors such as lighting, occlusion, shadows, background conditions, and camera orientation affect the performance of CV-based techniques [16].

Although these sensors are widely used, they may cause discomfort or restrict the user's movement. To overcome the drawbacks mentioned earlier, these sensors can be integrated into wearable electronics such as wristbands, gloves, and rings. However, real-time SLR based on wearable technology presents numerous challenges, including inconsistencies in users' signing actions, environmental noise that affects sensor data quality, and the need for reliable algorithms that can handle diverse signing styles. Therefore, it requires multifaceted research combining sensor technologies, machine intelligence, signal preprocessing, and robust design techniques to address these issues [17, 18]. Even so, wearable SLR systems can help deaf individuals become more independent and better understand events in social and professional contexts. Therefore, wearable-technology-based real-time SLR systems represent a promising approach to supporting the deaf community. Further innovations and improvements in wearable-sensor-based SLR systems and ML frameworks can enhance communication and quality of life for people with hearing impairments.

The present study aims to develop an AIEWG system equipped with sensors capable of recognizing real-time American Sign Language (ASL) gestures in order to overcome communication barriers for individuals with hearing and speech impairments. The study integrates a hardware module with ML models to classify and interpret ASL. The main hardware components include MPU6050 inertial measurement unit (IMU) sensors, flex sensors, and an ESP32 microcontroller. The ESP32 serves as the processing unit that collects and processes raw real-time signals and interfaces with the other hardware components and the web server. The subject's hand position and motion in three dimensions are recorded by the IMU sensors, which are essential for gesture recognition. Additionally, three flex sensors are positioned on the glove's fingers to detect flexion and finger movement, thereby distinguishing intricate ASL signs. Therefore, the primary contributions of this study include:

- Design and development of a wearable glove system integrating IMU and flex sensors for real-time ASL gesture acquisition.
- Implementation and evaluation of multiple ML models for ASL classification, with a PSO-tuned Extra Trees Classifier (EXT) achieving superior performance.
- Deployment of the trained model on an ESP32 microcontroller for real-time embedded inference.
- Development of an end-to-end real-time translation pipeline that converts ASL gestures into text and transmits outputs via Wi-Fi.
- Design of a web-based graphical user interface (GUI) for accessible and continuous communication between deaf and non-deaf users.

The following sections describe the operation of the AIEWG system, including hardware assembly, real-time data collection and processing, ML model training, deployment on a hardware device, accurate ASL recognition, wireless communication, and the design of an intuitive text-display interface.

2. System Architecture and Methodology

2.1 Hardware Setup

As discussed earlier, the hardware consists of IMU and flex sensors incorporated into a glove that the user wears so that hand movements can be captured by the sensors. Specifically, two IMU sensors are placed on the thumb and index finger of the glove, and three flex sensors are mounted on the ring finger, little finger, and middle finger. Each IMU sensor records detailed information about finger orientation, direction, acceleration, and angular velocity related to the alphabets and words (gestures). The MPU6050 sensor module includes a 3-axis gyroscope and a 3-axis accelerometer and operates at 3.3 V to 5 V. The gyroscope measures angular velocity in degrees per second, enabling the detection of hand rotation, while the accelerometer captures linear acceleration and gravitational forces across the X, Y, and Z axes, thereby measuring hand orientation and movement in different directions. In addition, the degree of finger bending is detected by the flex sensors attached to the glove. As the degree of flexion changes, the resistance of these sensors also changes, enabling quantitative measurement of finger flexion.

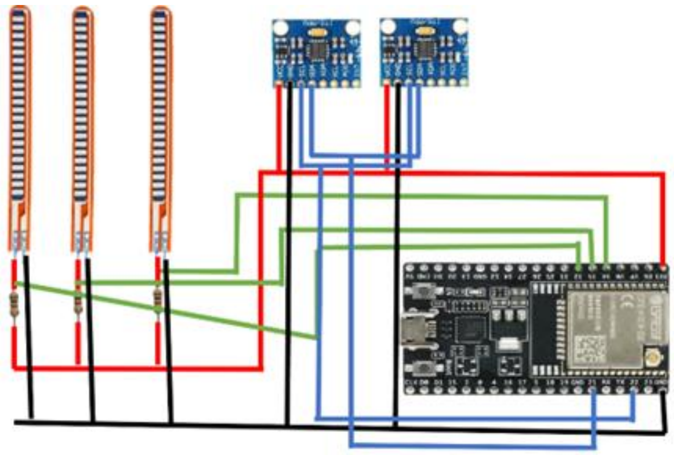


Figure 1. Off-glove electronic circuitry used for data acquisition and processing in the proposed AIEWG system.

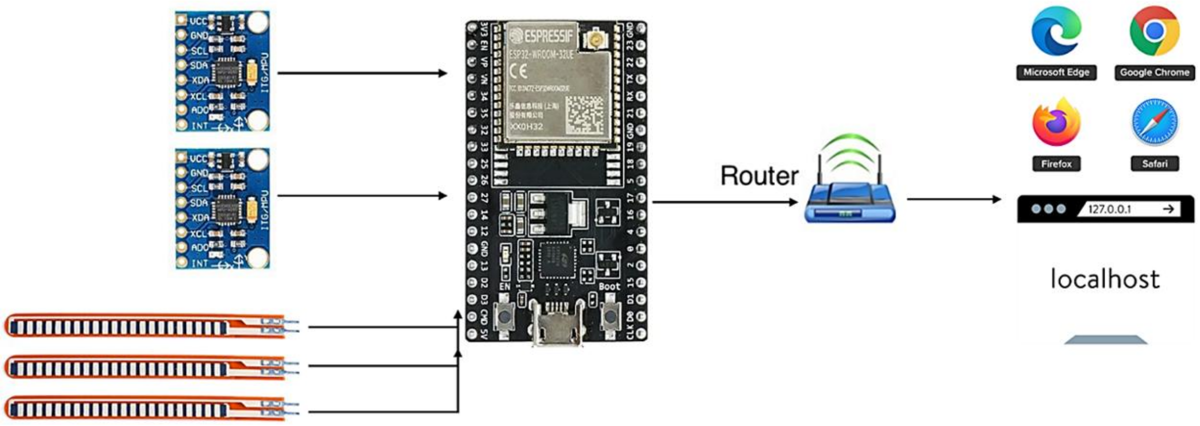


Figure 2. Wearable glove electronic unit transmitting real-time sensor data to a web browser via ESP32 Wi-Fi communication (localhost-based interface).

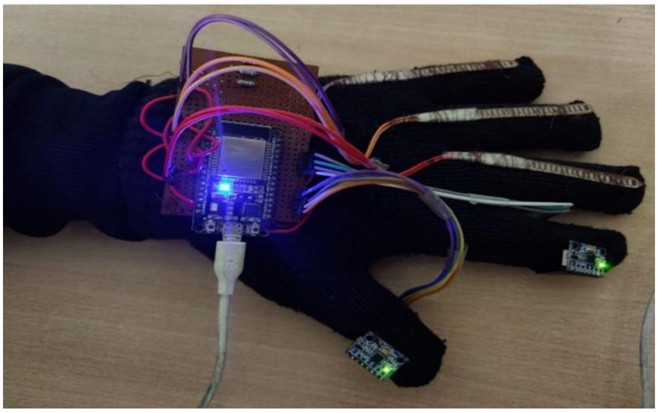


Figure 3. Complete architecture of the proposed AI-enabled Wearable Glove (AIEWG) system for ASL gesture recognition.

The flex sensors are attached at precise points on the glove, allowing finger flexion to be recorded accurately. By combining the two sensor types, the wearable electronics can detect subtle variations in finger position and curvature. Therefore, the overall hardware arrangement comprises a glove circuit, a computing unit, and interconnected components. This arrangement enables a compact, lightweight design that is easy to wear and does not interfere with hand movements during signing. The complete electronic hardware circuit mounted on the wearable glove provides a convenient solution for capturing, analyzing, and interpreting hand movements to identify ASL gestures in real time. Figure 1 shows the off-glove circuitry.

The MPU6050 and the flex sensors complement each other in capturing the full range of hand and finger movements required for ASL gesture recognition. Integrating data from both types of sensors improves the accuracy and reliability of the gesture recognition system, enabling more accurate interpretation of ASL gestures for effective communication. Since each flex sensor has a resistance that varies with finger curvature, a voltage divider circuit is used to obtain the corresponding voltage [2].

Figure 2 displays the overall hardware setup of the glove, which integrates the sensors with the ESP32 microcontroller. The flex sensors are connected to the ESP32 through analog pins, with each flex sensor paired with a 56 kΩ resistor to form a voltage divider. This enables the controller to measure changes in resistance when a flex sensor bends. The voltage divider circuit outputs a voltage proportional to the degree of finger bending, which is read by an analog-to-digital converter (ADC) in the ESP32. In contrast, the MPU6050 IMU sensor communicates with the ESP32 via the Inter-Integrated Circuit (I2C) protocol, facilitating data transfer between the microcontroller and the IMU. This allows the retrieval of acceleration, orientation, and angular velocity data from the IMU sensors. The I2C interface simplifies wiring by allowing multiple sensors to connect to the microcontroller through only two shared data lines (SDA and SCL). Thus, the glove design achieves its data-collection capability by incorporating both flex and IMU sensors, as shown in Figure 3.

2.2 Data Acquisition and Signal Collection

Data were collected using a custom-designed glove equipped with sensors to record hand movements during American Sign Language (ASL) gestures. For data collection, the glove was worn by 10 different subjects, who were asked to perform a series of gestures (each lasting 3 s) representing the 26 letters of the alphabet and five common ASL words in a randomized manner. The sensors operated at a sampling frequency of 100 Hz, and each gesture recording was captured as a continuous trial of approximately 3 seconds, corresponding to 300 time steps per gesture window. First, the system was calibrated, and the two MPU6050 sensors were configured to acquire accelerometer readings in g-force and gyroscope readings in degrees per second, which were then processed to calculate pitch, roll, and yaw angles for each finger. The voltage from each flex sensor was then converted into a digital value using the ESP32's built-in ADC. The sensors were calibrated to convert raw ADC values into sensor-angle values representing finger flexion.

The collected data included several parameters, such as angle values from the flex sensors and pitch, yaw, and roll angles from the MPU6050, which indicate hand orientation. By combining accelerometer and gyroscope data, a complementary filter can produce stable and accurate estimates that are formatted and sent serially to the computer for analysis and model training. Figures 4 and 5 display the data collection protocol and the real-time collection of data points from the glove circuitry.

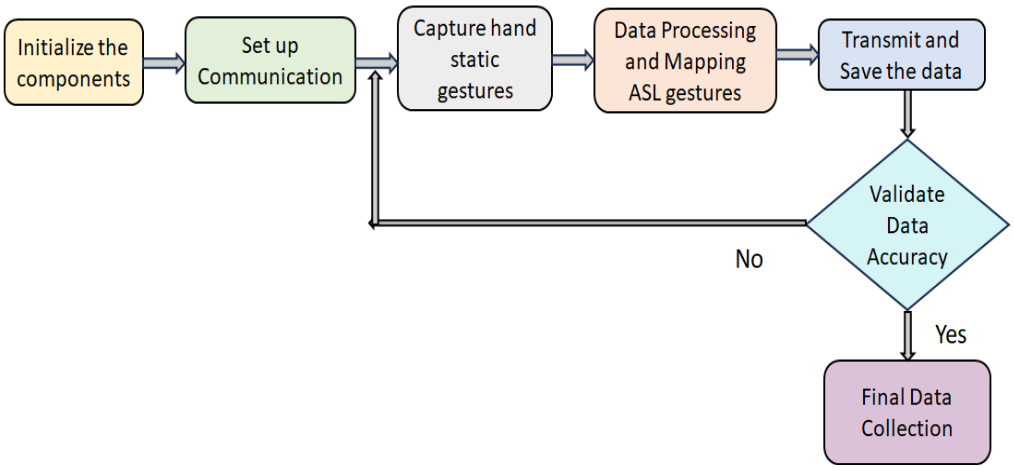


Figure 4. Flowchart illustrating the data collection protocol, including gesture acquisition, preprocessing stages and validation.

Figure 5. Sample ASL gestures representing letters “I”, “A”, and “Y” used in dataset construction.

These 10 participants (8 males and 2 females), aged 18 to 22, were volunteers from the university student population. To collect data, participants were instructed to perform predetermined ASL movements while wearing the proposed AIEWG device on their left hand. We ensured that all participants were comfortable throughout the experiment and assured them that their information would remain completely confidential. The study was approved by our university's Ethical Review Board, and participation was voluntary. An informed consent form detailing the study's goals, methods, potential risks and benefits, and participants' rights was read and signed by each subject before participation. The experiments were conducted in our Measurement and Transducers Laboratory in accordance with the university's guidelines.

Thus, data collection involved wearing the AIEWG, performing ASL gestures, recording the raw real-time readings, and extracting the dataset for training the ML models. This real-time data collection using the AIEWG ensured accurate and reliable data across multiple ASL gestures.

2.3 Data Preprocessing and Classifier Models

The data collected from participants while they performed ASL gestures served as the input dataset for training and testing the ML models. The dataset was collected from 10 subjects, each of whom performed a single continuous recording for each gesture class.

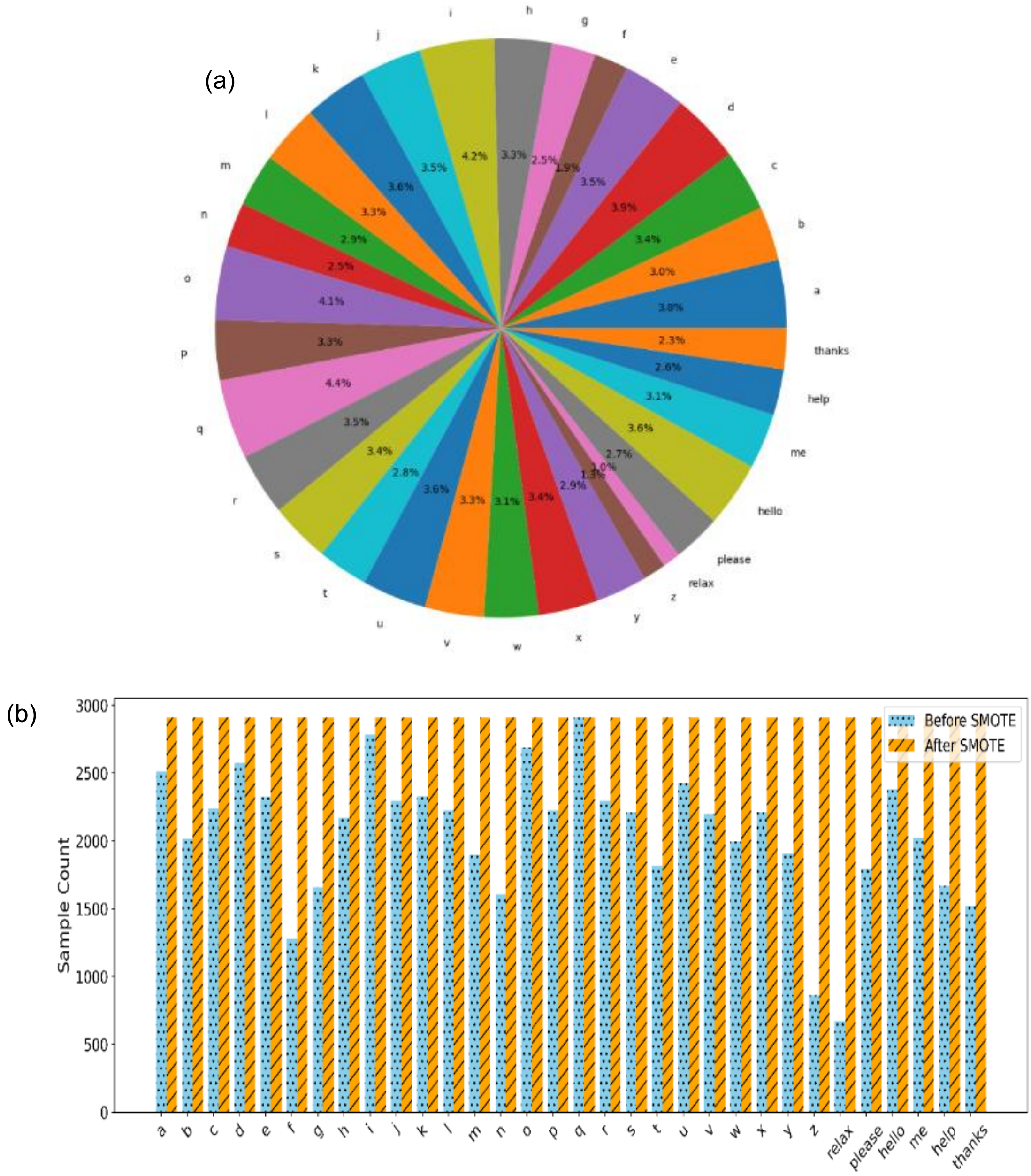


Figure 6. Dataset distribution analysis: (**a**) class-wise distribution of ASL gestures, (**b**) data balancing before and after applying SMOTE.

All recordings were segmented, preprocessed, and then combined to form a unified dataset consisting of 82,155 valid samples, each containing 23 features representing hand movements and orientations. These samples were obtained after removing corrupted or incomplete segments. A subject-independent evaluation protocol was then applied, in which the dataset was split at the participant level. Data from 8 subjects (80%) were used for training, while data from the remaining 2 unseen subjects (20%) were used for testing. This ensured that no samples from the same subject appeared in both the training and testing sets, thereby guaranteeing a subject-independent evaluation and eliminating subject overlap between the two sets. SMOTE was then applied to balance the training dataset. We trained different ML algorithms to determine the most effective model for ASL gesture recognition. Specifically, we evaluated the performance of eight ML algorithms and five DL algorithms: Random Forest (RF), Categorical Boosting (CB), Support Vector Classifier (SVC), K-Nearest Neighbor (KNN), Extreme Gradient Boosting (XGB), Decision Tree (DT), Logistic Regression (LR), and Extra Trees Classifier (EXT), along with Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Bidirectional Long Short-Term Memory (BiLSTM), Convolutional Neural Network + Long Short-Term Memory (CNN-LSTM), and Convolutional Neural Network + Bidirectional Long Short-Term Memory (CNN-BiLSTM). To address class imbalance, the oversampling technique SMOTE was applied. Figures 6a and 6b display the class-wise distribution percentages and the number of data samples before and after SMOTE. After SMOTE, the training sample size increased to 93,088.

2.4 PSO-Based Hyperparameter Optimization

Particle Swarm Optimization (PSO), a population-based metaheuristic method inspired by the social behavior of fish and birds, is useful for hyperparameter tuning (HT) of ML models [19]. Each swarm particle in PSO-based HT represents a potential solution, in this case a particular set of ML model hyperparameters. To identify the hyperparameters that maximize a predetermined objective function, these particles "fly" through the search space, updating their positions according to both the swarm's best-known position (global best) and each particle's personal best.

Depending on the task, this objective function is usually a performance metric, such as accuracy, a cross-validated F1-score, or another appropriate evaluation measure. The cross-validated F1-score is an excellent option, particularly for imbalanced classification problems, and can therefore be used as the objective function in PSO for ML hyperparameter tuning.

During each iteration, the ML model is trained and evaluated using each particle's current hyperparameters, and the resulting metric guides the swarm's movement. This procedure continues until a stopping criterion is satisfied, such as convergence to a solution or reaching the maximum number of iterations.

Because PSO does not require gradient information and can effectively explore complex, non-convex regions, it is well suited for HT. This makes it ideal for optimizing continuous and discrete hyperparameters in black-box ML models. When the class distribution is skewed, the F1-score is more informative than accuracy because it balances precision and recall. When cross-validation is used, the measured performance is more robust and less affected by a single train-test split. Figure 7 shows the convergence graph for the hyperparameter optimization of the EXT model.

The configuration parameters used for PSO are as follows: number of particles = 10, dimensions = 4, c1 (cognitive coefficient) = 0.5, c2 (social coefficient) = 0.3, w (inertia weight) = 0.9, and iterations = 30. At iteration t, each particle updates its velocity and position using the following equations:

$$v_i(t+1) = w.v_i(t) + c_1.r_1.\left(p_i - x_i(t)\right) + c_2.r_2.\left(g - x_i(t)\right) \quad (1)$$

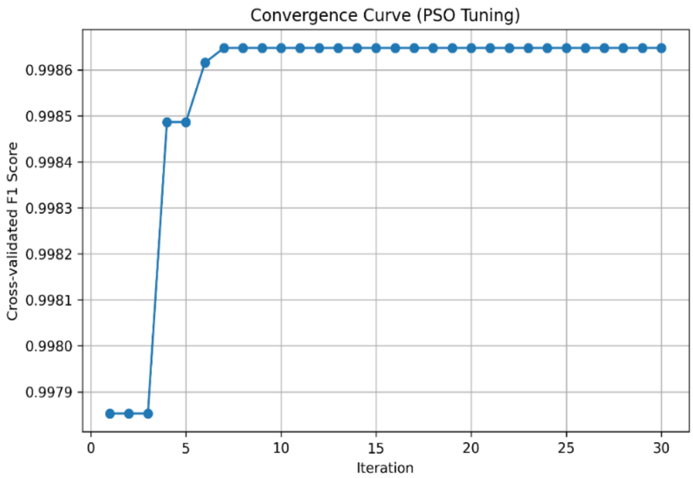


Figure 7. Convergence behavior of the PSO algorithm for tuning the EXT model parameters

Where, x_i is the position vector representing a candidate hyperparameter configuration of the Extra Trees classifier defined as,

$$x_i = [n_{estimators}, max_{depth}, min_{samples_{split}}, min_{samples_{leaf}}] \quad (2)$$

$v_i(t)$ is the present velocity of the i[th] particle, r_1 and r_2 are random numbers to introduce stochasticity, p_i is the best-known position of the particle and g is the global best position found by the swarm.

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (3)$$

Since PSO is a minimization technique, we guide the swarm toward optimal solutions by negating the F1-score.

$$Fitness\ (x_i) = -\frac{1}{k}\sum_{j=1}^{k} F_{weighted}^{j} \quad (4)$$

Where, k =5 denotes the number of cross-validation folds, and $F_{weighted}^{j}$ is weighted F1 score. The goal of the objective function was to minimize the negative weighted F1-score, which was determined using 5-fold cross-validation on the SMOTE-resampled training data. The hyperparameters optimized using PSO for EXT model are $n_{estimators}$ = 380, max_{depth}=50, $min_{samples_{split}}$ = 2, $min_{samples_{leaf}}$= 1.

3. System Implementation and Deployment

3.1 Web Server Setup and ESP32 Communication

Once the model is deployed on the ESP32, the web server configuration plays a vital role in enabling real-time interaction with the ASL gesture recognition system. By taking advantage of the ESP32's Wi-Fi capabilities, we establish a local network environment that allows users to access the gesture recognition system from any device with a web browser. This configuration increases accessibility and flexibility by eliminating the need for a physical connection and enabling wireless interaction. The ESP32 hosts a web server on port 80 (the standard port for HTTP communication) to handle incoming HTTP GET requests to the "/get Gesture" route over Wi-Fi. When a GET request is received, the server triggers the "handleGetGesture" function, which retrieves the latest predicted gesture label generated by the embedded machine learning model. Gesture prediction is performed using the optimized EXT model integrated into the ESP32 firmware through the eloquent library. This model processes data collected from hybrid sensors and uses feature vectors to make accurate predictions. The predictive capability of the model depends on the training process, during which it learns to identify patterns in sensor data corresponding to different ASL gestures. An HTML response in the "sendPredictedGesture" function was created to enable continuous communication between the ESP32 and the client device. The client-side interface is updated based on this response to display predicted gestures, ensuring that users receive real-time feedback. Users always have up-to-date gesture predictions because the interface is refreshed periodically using JavaScript. Essentially, the web server setup bridges the ASL gesture recognition system and external devices, enabling real-time visual interaction. The proposed system thereby improves usability and communication for people with hearing loss. Thus, the sensor-integrated ESP32 combined with the AI framework enables consistent and effective gesture recognition under different scenarios. The complete proposed workflow is shown in Figure 8.

4. Results and Performance Analysis

4.1 ML Model Evaluation

The results demonstrate the performance of the ML model trained to recognize ASL gestures and provide insights through a real-time demonstration on a locally hosted website.

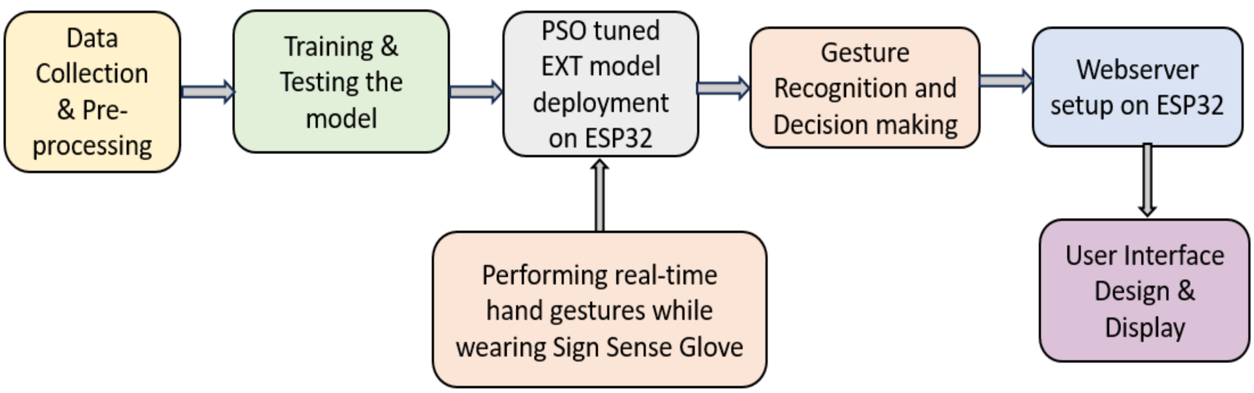


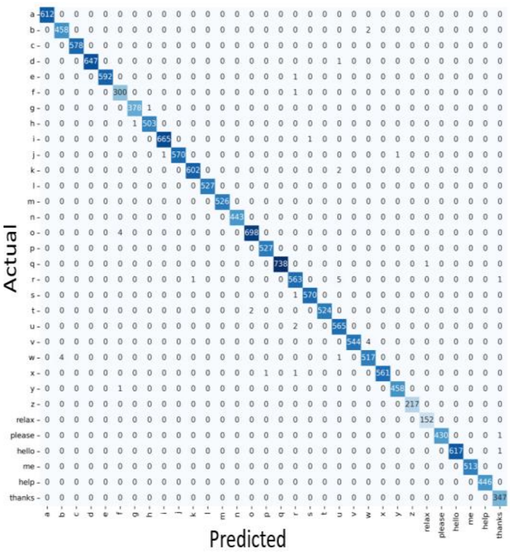
Figure 8. Block diagram representing the end-to-end workflow of the proposed ASL recognition system from sensing to text output.

Table 1. Performance metrics for all PSO-tuned ML models

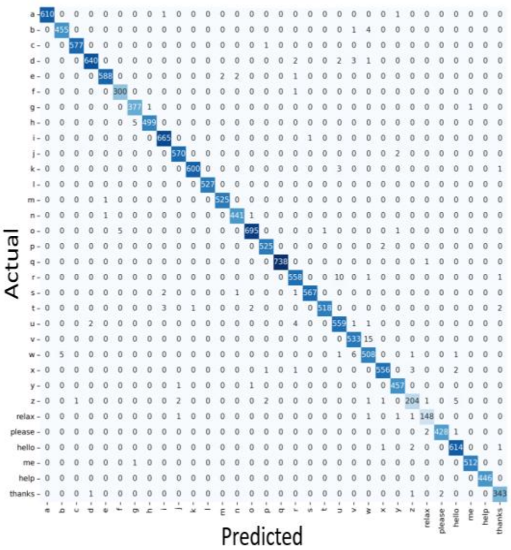
Models	Precision	Recall	F1 Score	Accuracy	AUC-ROC
RF	0.9974	0.9974	0.9974	0.9974	0.9999
DT	0.9910	0.9909	0.9910	0.9910	0.9954
LR	0.9874	0.9873	0.9871	0.9873	0.9999
CB	0.9976	0.9976	0.9976	0.9976	1.0000
KNN	0.9903	0.9901	0.9902	0.9901	0.9991
XGB	0.9972	0.9972	0.9972	0.9972	1.0000
SVC	0.9942	0.9942	0.9942	0.9942	1.0000
EXT	0.9987	0.9986	0.9987	0.9986	0.9999

Table 2. Ablation study showing the contribution of IMU, flex sensors, fused, and PSO-tuned EXT mode

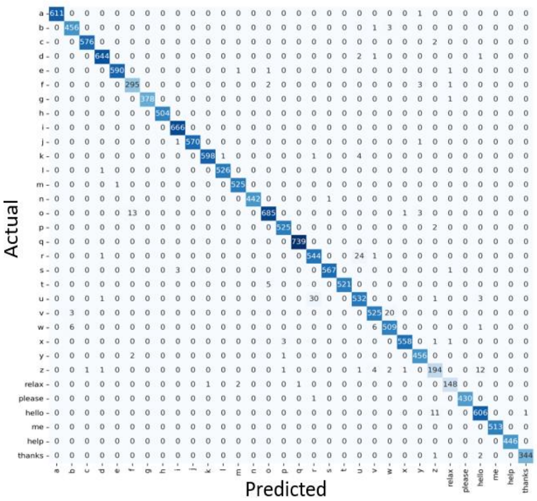
Type	Precision	Recall	F1 Score	Accuracy	AUC-ROC
IMU	0.9972	0.9971	0.9971	0.9971	0.9997
Flex	0.9301	0.9299	0.9297	0.9299	0.9978
Fused without PSO	0.9984	0.9984	0.9983	0.9984	0.9997
Fused with PSO	0.9987	0.9986	0.9987	0.9986	0.9999



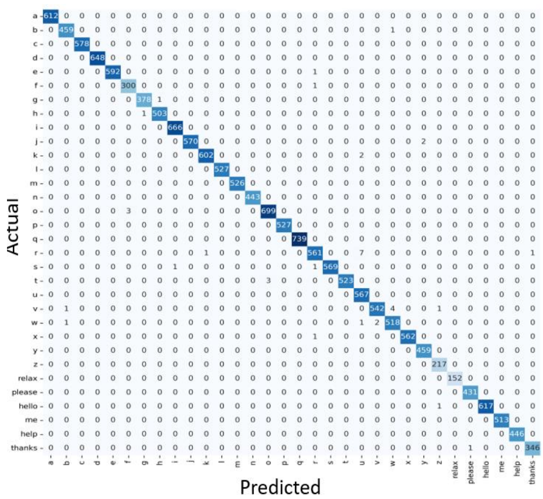
(a)



(b)



(c



(d

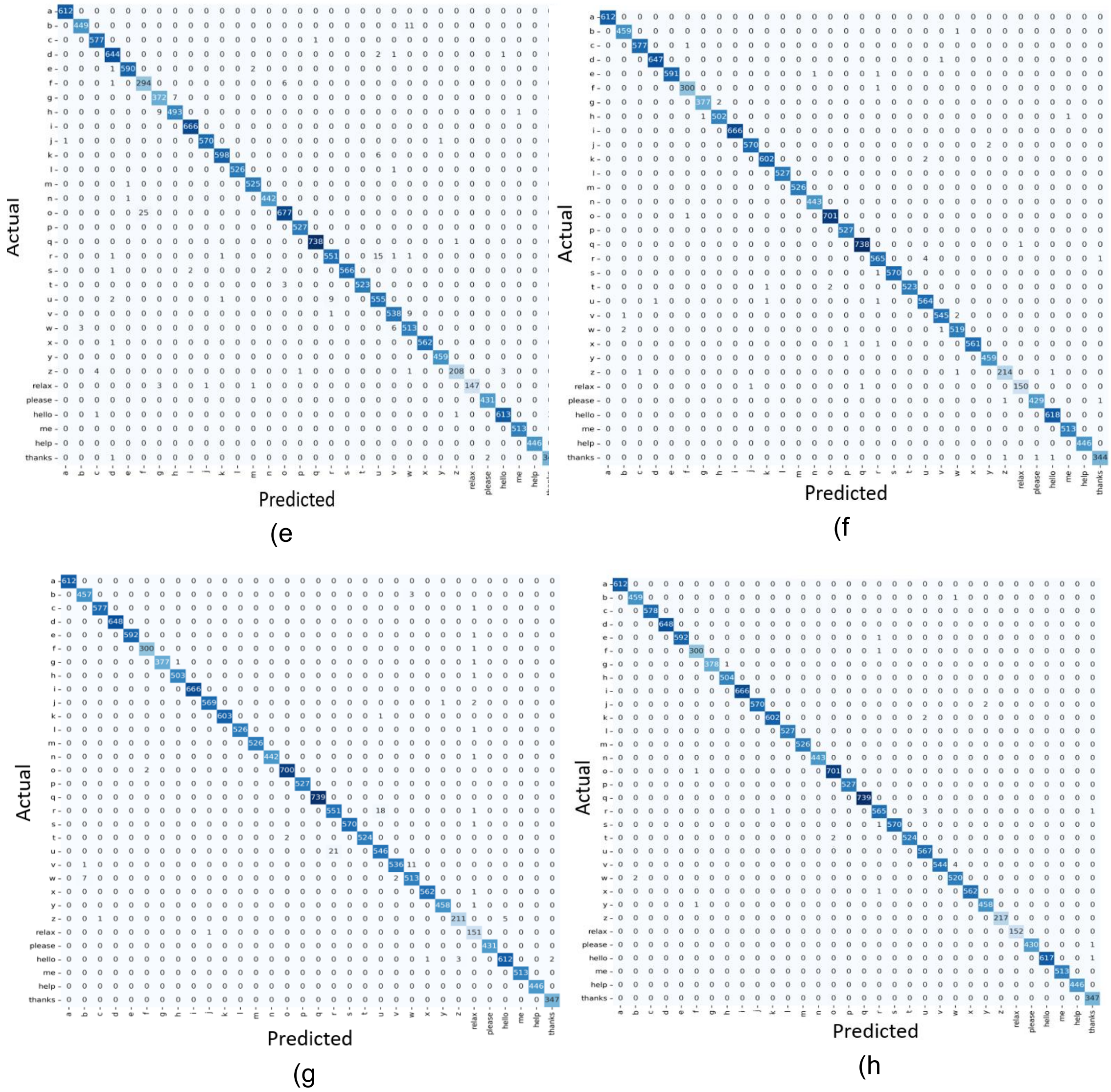


Figure 9. Confusion matrices of different PSO-tuned machine learning models: (**a**) RF, (**b**) DT, (**c**) LR, (**d**) CB, (**e**) KNN, (**f**) XGB, (**g**) SVC, (**h**) EXT.

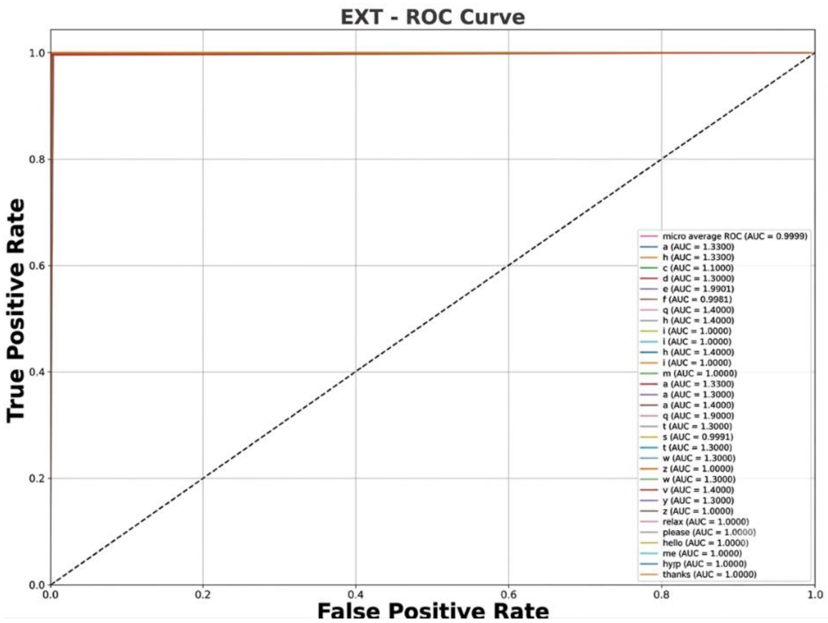


Figure 10. Receiver Operating Characteristic (ROC) and AUC curves of the PSO-tuned EXT model.

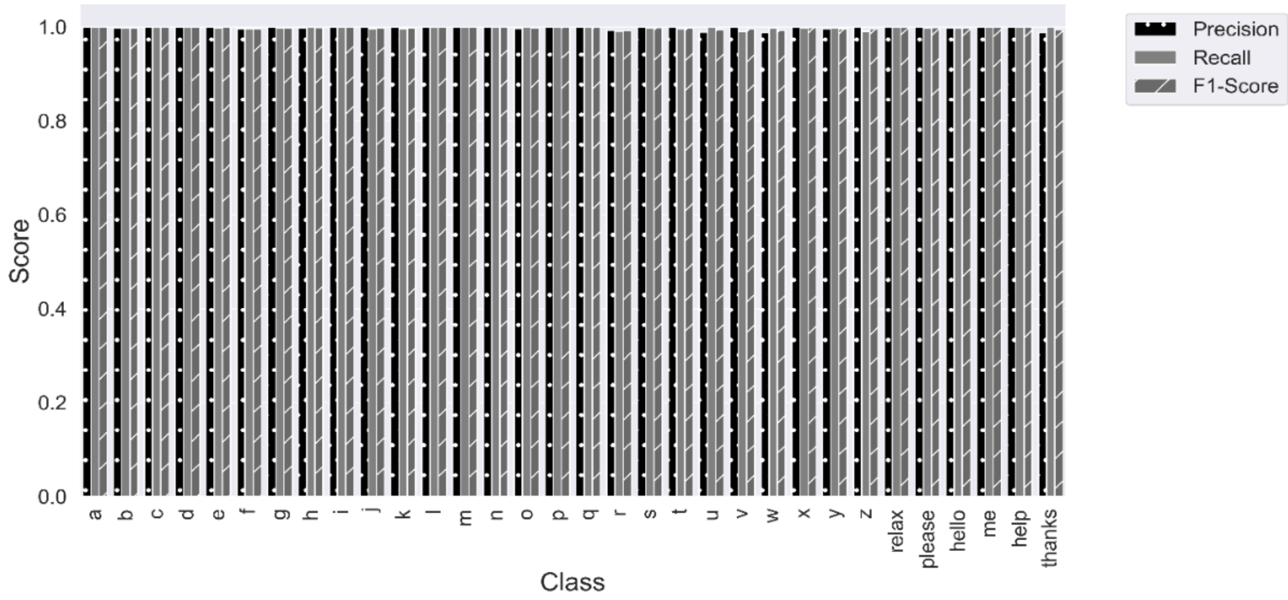


Figure 11. Class-wise performance metrics (precision, recall, and F1-score) of the PSO-tuned EXT across all ASL classes.

In addition, the challenges encountered during implementation and the strategies used to address them are presented. Table 1 displays the performance metrics for all models, with EXT showing the best performance. The confusion matrix provides a detailed analysis of the model's performance by showing the number of true positives, false positives, true negatives, and false negatives for each ASL gesture class. It visually demonstrates the model's ability to classify gestures and accurately identify patterns of misclassification.

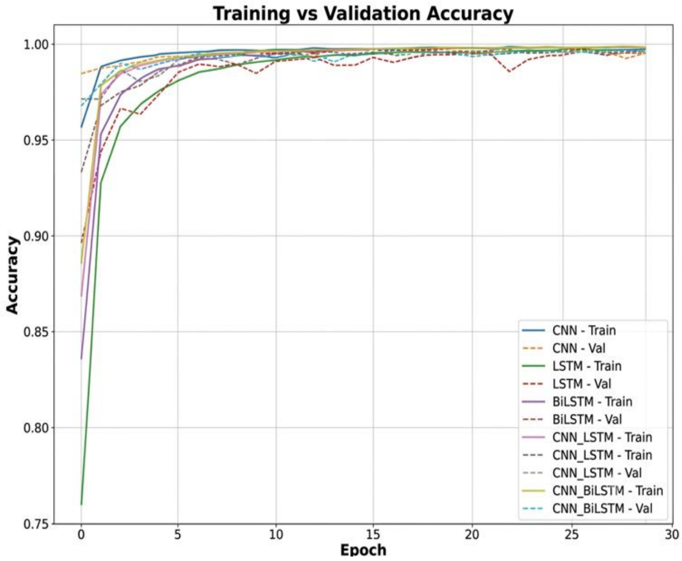
To assess the contribution of each sensing component in the proposed system, an ablation study was carried out, as shown in Table 2. The findings demonstrate that flex-only sensing yields lower accuracy, whereas IMU-only sensing achieves good performance because of its strong ability to capture motion and orientation. The improvement in overall performance achieved through the fusion of IMU and flex sensors demonstrates the advantage of multimodal sensing. Furthermore, PSO-based tuning provides a slight additional increase in classification accuracy by refining the model parameters.

These results help us understand whether certain gestures are systematically misclassified and which areas of the model need improvement. Figure 9 (a-h) displays the confusion matrices of all PSO-tuned ML models, with EXT performing better. Figures 10 and 11 present the ROC curve and class-wise metrics for the PSO-tuned EXT model, which show promising results compared with the other models.

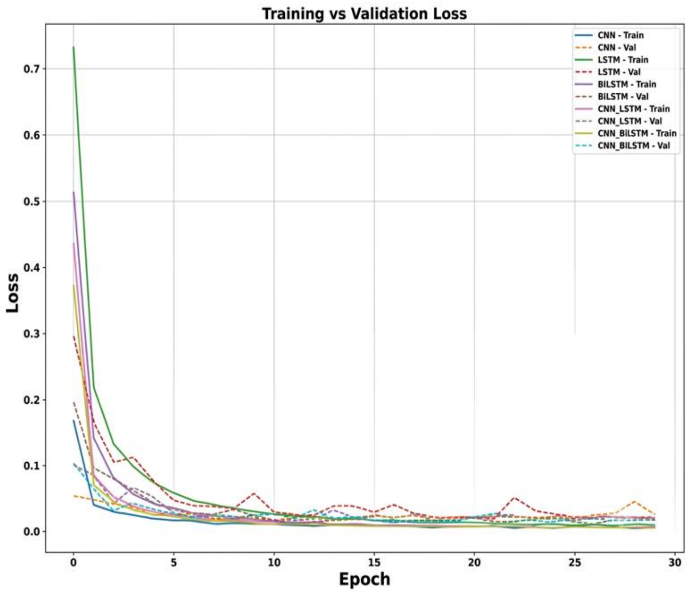
4.2 DL Model Evaluation

In addition, the Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), Bidirectional LSTM (BiLSTM), CNN-LSTM, and CNN-BiLSTM models were deployed to assess the classification performance of several DL architectures. Precision, recall, F1-score, training accuracy, testing accuracy, and the Area under the Receiver Operating Characteristic Curve (AUC-ROC) were among the performance indicators used to evaluate each model. Table 3 summarizes the findings. With AUC-ROC values of 1.000 across all architectures, Table 3 shows that all models achieved very high classification performance, demonstrating a strong ability to distinguish between classes. The CNN-BiLSTM architecture performed better than the other models in almost every metric, including precision (0.9961), recall (0.9960), F1-score (0.9960), and testing accuracy (0.9960). This illustrates how effectively convolutional feature extraction and BiLSTM's bidirectional temporal learning capabilities work together.

With F1-scores of 0.9951, 0.9952, and 0.9952, respectively, the standalone CNN, LSTM, and BiLSTM models likewise demonstrated exceptional performance, indicating their ability to capture spatial and sequential dependencies on their own. With an F1-score of 0.9955, the CNN-LSTM hybrid model outperformed the basic models. The training and validation accuracy and loss curves for the DL models are shown in Figure 12. The training and validation accuracies over epochs are plotted on the left, and the corresponding loss curves are shown on the right. The plot of accuracy against the number of training iterations illustrates how model accuracy changes over successive training iterations. These curves track the training process and indicate whether the model has converged or requires additional training iterations. Once the curve reaches maximum accuracy and becomes stable, it indicates that the model has learned effectively from the training data.



(a)



(b)

Figure 12. Training and validation (a) accuracy and (b) loss curves of DL models used for ASL gesture classification

Table 3. Performance metrics for all DL models

Models	Precision	Recall	F1 Score	Training Accuracy	Testing Accuracy	AUC-ROC
CNN	0.9952	0.9951	0.9951	0.9983	0.9951	1.0000
LSTM	0.9953	0.9953	0.9952	0.9974	0.9953	1.0000
BiLSTM	0.9953	0.9952	0.9952	0.9986	0.9952	1.0000
CNN-LSTM	0.9955	0.9955	0.9955	0.9977	0.9955	1.0000
CNN-BiLSTM	0.9961	0.9960	0.9960	0.9987	0.9960	1.0000

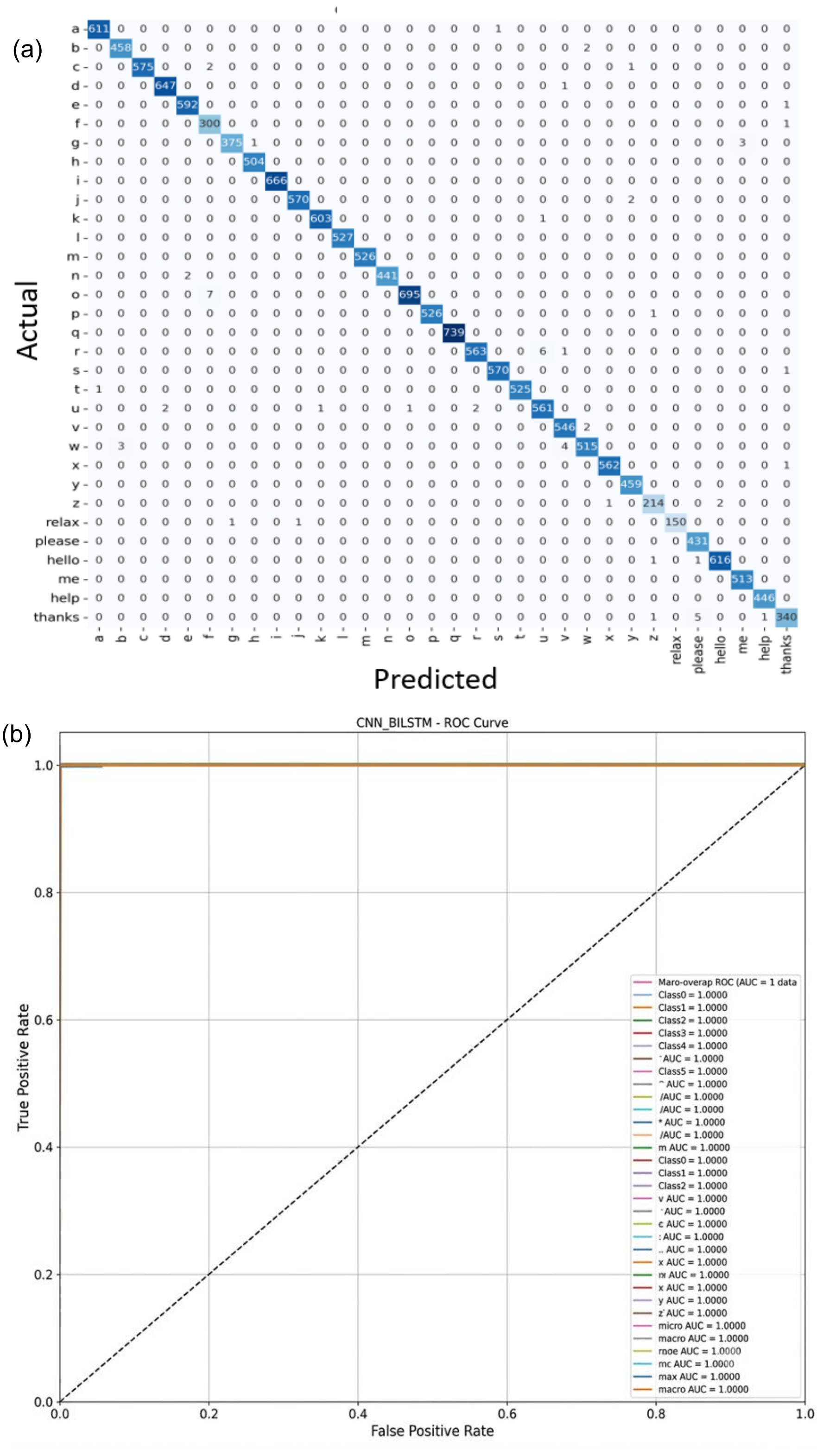


Figure 13. Performance evaluation of CNN-BiLSTM model: (**a**) confusion matrix, (**b**) ROC curve

Thus, a model typically exhibits increasing accuracy during the initial epochs, followed by a stabilization period that reflects learning and convergence. Remarkably, the CNN-BiLSTM model demonstrates superior generalization capability by attaining the maximum precision, F1-score, recall, and validation accuracy. The corresponding confusion matrix and AUC-ROC curve are shown in Figure 13, whereas the CNN and LSTM models take longer to converge.

A comparison of the performance metrics of both ML and DL models shows that the PSO-tuned EXT classifier achieves the highest testing accuracy of 99.86%. Therefore, the PSO-tuned EXT classifier was chosen for deployment on the ESP32 microcontroller. For ease of integration, we saved the trained model in .h format, which is compatible with the Arduino IDE. This allowed us to upload the model directly to the ESP32 microcontroller for real-time sensor data processing and gesture recognition on the device itself. By implementing the model on the ESP32, we achieved a streamlined and efficient system capable of ASL gesture recognition without relying on external processing resources.

4.3 Real-Time System Demonstration

The standard ASL letters from A to Z and the words "please," "hello," "thank you," "help," and "me" are shown in Figure 14 (a-f). The interactive user interface demonstrates the model's real-time ASL gesture recognition capabilities on a locally hosted website. Users can input gestures through the interface, and the AIEWG wearable device provides immediate feedback by recognizing hand motions and translating them into legible text. Figure 15 shows how well the AIEWG performs in real time in identifying static hand motions corresponding to different alphabetic letters and words.

It provides a realistic demonstration of the model's accuracy and responsiveness, allowing users to experience its performance in a real-world environment. In addition, this demonstration plays a vital role in validating the model's effectiveness and evaluating its usability and user experience.

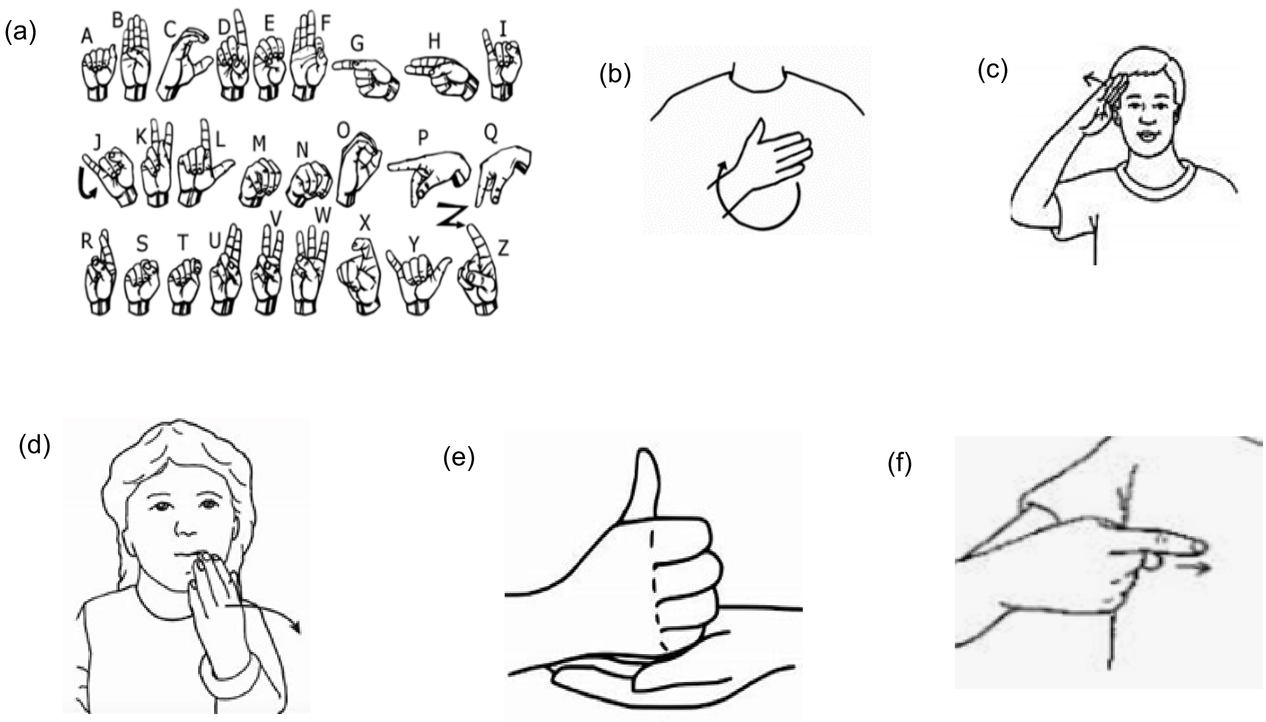


Figure 14. Standard ASL gesture samples used in the study: (**a**) alphabets, (**b**) "please", (**c**) "hello", (**d**) "thank you", (**e**) "help", (**f**) "me".

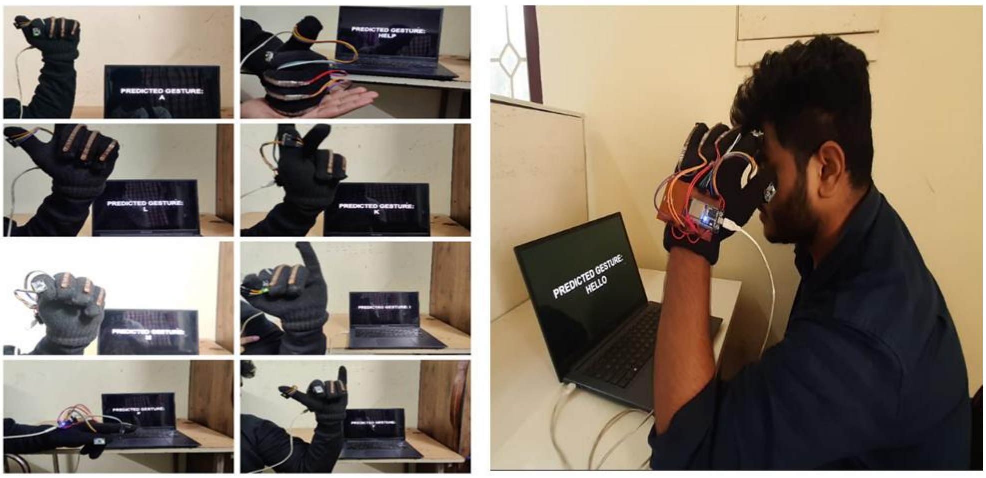


Figure 15. Real-time ASL gesture recognition results showing live detection and text output for multiple letters and words using the proposed AIEWG system.

5. Discussions

In the proposed AIEWG system, two IMU sensors are placed on the thumb and index finger of the glove, and three flex sensors are mounted on the ring finger, little finger, and middle finger. The MPU6050 IMU sensors capture wrist orientation and motion dynamics, while the flex sensors record fine-grained finger-bending data. The combination of multiple sensing modalities greatly improves feature separability, thereby enhancing the overall performance of the proposed system. The performance evaluation of the proposed AIEWG system demonstrates the effectiveness of the PSO-tuned EXT model in accurately identifying ASL movements from multimodal sensor inputs. The strong diagonal structure of the confusion matrix indicates that most gesture classes are accurately categorized, with little misclassification. This demonstrates that, for static ASL gesture identification, the AIEWG system effectively captures discriminative feature patterns from both flex sensors and MPU6050 inertial measurement units. Based on our literature review, various sensor-based gesture recognition studies are summarized in Table 4 for ease of comparison. According to the comparison, several studies report lower accuracy than the present work. Nevertheless, these studies differ in the number of volunteers, sign language gestures, hardware design, and model validation techniques.

It is evident that previous research has mostly used DL models such as CNNs and LSTMs in conjunction with IMU- and flex-sensor-based systems. For example, Faisal *et al.* [20] used a parallel-path CNN with IMU and flex sensors and reported 84.42% accuracy across 25 participants, while Wen *et al.* [21] combined triboelectric nanogenerator (TENG) sensors with a 1D CNN model and achieved 86.67% accuracy. Similarly, Maharjan *et al.* [22] used an LSTM model with MPU6050 and flex sensors but achieved only 80% accuracy, suggesting that sequential modeling for gesture recognition remains challenging. In contrast, Kanwal *et al.* [23] reported 95.6% accuracy using an LSTM-based approach with flex sensors and IMU data, while Liu *et al.* [18] achieved 95.85% accuracy using strain sensors and CNN-based classification for Chinese sign language. Nevertheless, many of these methods either rely on computationally intensive DL architectures or were tested with a small number of participants, which could affect their suitability for embedded deployment and real-world generalization. In contrast, the proposed solution achieves a much higher accuracy of 99.86% across 10 participants, even though it uses a simpler PSO-tuned EXT model rather than DL techniques. Depending on Wi-Fi usage and processing overhead, the system achieves an end-to-end latency of 25–60 ms at a 100 Hz sampling rate. About 10–30 ms of the overall latency is due to Wi-Fi communication delay. Because of the ensemble size (380 trees), on-device inference with the EXT classifier takes about 5–25 ms. The system shows stable real-time performance for wearable ASL gesture detection using ESP32 hardware, with a total model size of about 4.05 MB and a runtime memory requirement of about 800 KB.

This work still has limitations related to signer variability, which may affect feature consistency and classification accuracy across users, as well as variations in finger length, hand geometry, gesture articulation patterns, and sensor positioning. Recalibration may be required because of potential sensor drift, variation in flex-sensor sensitivity, and changes in IMU alignment over prolonged use. Integrating multiple flex sensors and IMU modules and routing the wires may also affect long-term wearing comfort. In addition, the sensor mapping and gesture orientation may not easily translate to right-hand use, as the proposed prototype uses a left-hand glove configuration. Another challenge of the proposed approach is distinguishing gestures with similar finger-bend and wrist-orientation patterns. These similarities may lead to misclassification, especially when user-related variations occur in real time. Furthermore, the existing system does not yet model the temporal transitions necessary for dynamic sign recognition and supports only static ASL movements.

Table 4. Comparison of sensor-based hand gesture recognition studies

Author and year	Sensors	Classifiers	Number of participants	Accuracy
Wen *et al.* [21]	Triboelectricnanogenerator (TENG) sensors	1D CNN	Not mentioned	86.67%
Faisal *et al.* [20]	1 IMU and 5 flex sensors	Parallel-path CNN	25	84.42%
Maharjhan *et al.* [21]	MPU-6050 and five flex sensors	LSTM	18	80%
Liu *et al.* [18]	Strain sensors, an IMU and FPCB for Chinese sign language	CNN	Not mentioned	95.85%
Kanwal *et al.* [23]	5 flex sensors and an IMU for Urdu sign language	LSTM	1	95.6%
Proposed	2 IMU and 3 flex sensors	PSO-tuned EXT model	10	99.86%

6. Conclusions

This project presents an AIEWG system for real-time ASL gesture recognition using a sensor-based hardware platform integrated with an ML framework. The proposed system captures real-time data on finger bends, hand motions, and wrist orientations and successfully translates ASL gestures into text. Although DL models were also evaluated, the PSO-tuned EXT model achieved superior performance in terms of accuracy, precision, recall, and F1-score. This demonstrates the effectiveness of the proposed model in accurately classifying ASL gestures using the developed AIEWG through web-based visualization. From a practical perspective, this device advances assistive communication technology for people with disabilities by reducing communication barriers and enabling more effective interaction with the digital world. However, the current prototype has certain limitations. The functionality of the proposed device should be improved by testing it with more subjects, and the current study focuses only on static gesture recognition. The most practical next step for future deployment is to scale the system to larger and more diverse user groups, incorporate reliable dynamic gesture recognition, and further optimize the model for extended wearable use.

References

[1] B. Jain, M. Chandna, A. Dasgupta, K. Bansal, Sign Language Recognition: Current State of Knowledge and Future Directions. In 2023 14th International Conference on Computing Communication and Networking Technologies (ICCCNT) IEEE, (2023) 1–7. https://doi.org/10.1109/ICCCNT56998.2023.10307879

[2] A. Filipowska, W. Filipowski, P. Raif, M. Pieniążek, J. Bodak, P. Ferst, K. Pilarski, S. Sieciński, R.J. Doniec, J. Mieszczanin, E. Skwarek, K. Bryzik, M. Henkel, M. Grzegorzek, Machine Learning-based Gesture Recognition Glove: Design and Implementation. Sensors, 24(18), (2024) 6157. https://doi.org/10.3390/s24186157

[3] R. Tchantchane, H. Zhou, S. Zhang, G. Alici, A Review of Hand Gesture Recognition Systems Based on Noninvasive Wearable Sensors. Advanced Intelligent Systems, 5(10), (2023) 2300207. https://doi.org/10.1002/aisy.202300207

[4] İ. Umut, Ü. C. Kumdereli, Novel Wearable System to Recognize Sign Language in Real Time. Sensors, 24(14), (2024) 4613. https://doi.org/10.3390/s24144613

[5] J. Kim, S. Mastnik, E. André, EMG-based hand gesture recognition for realtime biosignal interfacing. in Proceedings of the 13th International Conference on Intelligent User Interfaces, (2008) 30–39. https://doi.org/10.1145/1378773.1378778

[6] J. Wu, L. Sun, R. Jafari, A Wearable System for Recognizing American Sign Language in Real-Time Using IMU and Surface EMG Sensors. IEEE Journal of Biomedical and Health Informatics, 20(5), (2016) 1281-1290. https://doi.org/10.1109/JBHI.2016.2598302

[7] B.G. Lee, S.M. Lee, Smart Wearable Hand Device for Sign Language Interpretation System with Sensors Fusion. IEEE Sensors Journal, 18(3), (2018) 1224-1232. https://doi.org/10.1109/JSEN.2017.2779466

[8] P.N. Huu, T. Phung Ngoc, Hand Gesture Recognition Algorithm Using SVM and HOG Model for Control of Robotic System. Journal of Robotics, 2021(1), (2021) 3986497. https://doi.org/10.1155/2021/3986497

[9] A. Kasapbaşi, A.E.A. Elbushra, O. Al-Hardanee, A. Yilmaz, DeepASLR: A CNN based human computer interface for American Sign Language Recognition for Hearing-Impaired Individuals. Computer Methods and Programs in Biomedicine Update, 2, (2022) 100048. https://doi.org/10.1016/j.cmpbup.2021.100048

[10] H. Alsolai, L. Alsolai, F.N. Al-Wesabi, M. Othman, M. Rizwanullah, A.A. Abdelmageed, Automated Sign Language Detection and Classification using Reptile Search Algorithm with Hybrid Deep Learning. Heliyon, 10(1), (2024) e23252. https://doi.org/10.1016/j.heliyon.2023.e23252

[11] T.H. Noor, A. Noor, A.F. Alharbi, A. Faisal, R. Alrashidi, A.S. Alsaedi, G. Alharbi, T. Alsanoosy, A. Alsaeedi, Real-Time Arabic Sign Language Recognition Using a Hybrid Deep Learning Model. Sensors, 24(11), (2024) 3683. https://doi.org/10.3390/s24113683

[12] I.A. Adeyanju, O.O. Bello, M.A. Adegboye, Machine Learning Methods for Sign Language Recognition: A Critical Review and Analysis. Intelligent Systems with Applications, 12, (2021) 200056. https://doi.org/10.1016/j.iswa.2021.200056

[13] J. Qi, L. Ma, Z. Cui, Y. Yu, Computer Vision-based Hand Gesture Recognition for Human-Robot Interaction: A Review. Complex & Intelligent Systems, 10(1), (2024) 1581-1606. https://doi.org/10.1007/s40747-023-01173-6

[14] M.M. Rahman, A. Uzzaman, F. Khatun, M. Aktaruzzaman, N. Siddique, A Comparative Study of Advanced Technologies and Methods in Hand Gesture Analysis and Recognition

Systems. Expert Systems with Applications, 266, (2025) 125929. https://doi.org/10.1016/j.eswa.2024.125929

[15] M. Alaghband, H.R. Maghroor, I. Garibay, A Survey on Sign Language Literature. Machine Learning with Applications, 14, (2023) 100504. https://doi.org/10.1016/j.mlwa.2023.100504

[16] R. Kaur, G. Karmakar, F. Xia, M. Imran, Deep Learning: Survey of Environmental and Camera Impacts on Internet of Things Images. Artificial Intelligence Review, 56(9), (2023) 9605-9638. https://doi.org/10.1007/s10462-023-10405-7

[17] K.R. Pyun, K. Kwon, M.J. Yoo, K.K. Kim, D. Gong, W.H. Yeo, S. Han, H. Ko, Machine-Learned Wearable Sensors for Real-Time Hand-Motion Recognition: Toward Practical Applications. National Science Review, 11(2), (2024) nwad298. https://doi.org/10.1093/nsr/nwad298

[18] Y. Liu, X. Jiang, X. Yu, H. Ye, C. Ma, W. Wang, Y. Hu, A Wearable System for Sign Language Recognition Enabled by a Convolutional Neural Network. Nano Energy, 116, (2023) 108767. https://doi.org/10.1016/j.nanoen.2023.108767

[19] M. Daviran, A. Maghsoudi, R. Ghezelbash, Optimized AI-MPM: Application of PSO for Tuning the Hyperparameters of SVM and RF Algorithms. Computers & Geosciences, 195, (2025) 105785. https://doi.org/10.1016/j.cageo.2024.105785

[20] M.A.A. Faisal, F.F. Abir, M.U. Ahmed, M.A.R. Ahad, Exploiting Domain Transformation and Deep Learning for Hand Gesture Recognition using a Low-Cost Dataglove. Scientific Reports, 12(1), (2022) 21446. https://doi.org/10.1038/s41598-022-25108-2

[21] F. Wen, Z. Zhang, T. He, C. Lee, AI Enabled Sign Language Recognition and VR Space Bidirectional Communication Using Triboelectric Smart Glove. Nature communications, 12(1), (2021) 5378. https://doi.org/10.1038/s41467-021-25637-w

[22] S. Maharjan, S. Shrestha, S. Fernando, (2024) Sign Language Detection and Translation Using Smart Glove. In Innovative Computing and Communications, A.E. Hassanien, S. Anand, A. Jaiswal, P. Kumar, Eds., Singapore: Springer Nature Singapore, 1021, 535–554. https://doi.org/10.1007/978-981-97-3591-4_41

[23] T. Kanwal, S. Altaf, R.M. Yousaf, K. Sattar, A Smart Glove-Based System for Dynamic Sign Language Translation Using LSTM Networks. Engineering Proceedings, 118(1), (2025) 45. https://doi.org/10.3390/ECSA-12-26530

Authors Contribution Statement

R. Valarmathi: Writing - Review & Editing. K. Ghousiya Begum: Conceptualization, Methodology, Formal Analysis, Supervision, Investigation, Writing - Review & Editing. R.D. Manoj: Writing - Original Draft, Formal analysis, Investigation. All authors have read and approved the final manuscript.

Funding

The authors declare that no funds, grants, or any other support were received during the preparation of this manuscript.

Competing Interests

The authors declare that they have no conflicts of interest regarding the publication of this manuscript.

Data Availability

The data supporting the findings of this study are available from the corresponding author upon reasonable request.

Has this article screened for similarity?

Yes