SVU-International Journal of Engineering Sciences and Applications (2025) 6(2): 195-203
Print ISSN 2785-9967 | Online ISSN 2735-4571
DOI 10.21608/SVUSRC.2025.388195.1287

AI-Based Framework for Real-Time Recognition of Arabic and English Sign Languages

Asmaa G. Seliem[1,□], Shaimaa Mohamed Elembay[1], Mohamed Nasser Elshayeb[1]

Abstract— Over 300 sign languages are used worldwide, posing challenges for effective communication between deaf and hearing individuals. This study presents a bilingual sign language recognition (SLR) system using deep learning to enhance accessibility for the deaf and mute communities. The system processes real-time video input, leveraging MediaPipe for hand and body landmark extraction. For static gesture classification (e.g., alphabet recognition), a Support Vector Machine (SVM) with a linear kernel is employed. For dynamic gesture sequences (e.g., word-level recognition), a Long Short-Term Memory (LSTM) network is used to model temporal patterns. The models were trained on large-scale datasets of Arabic and English sign languages, achieving recognition accuracies exceeding 99% for English letters and over 93% for selected Arabic words. The training dataset consists of images from Kaggle and real-time videos, and the test dataset uses independent real-time videos not seen during training. The system supports sign-to-text translation as well as voice and text-to-sign conversion through avatars or image sequences, promoting inclusive, real-time communication across linguistic boundaries.

Keywords: Deaf and Hard of Hearing (DHH); Long Short-Term Memory; MediaPipe; Sign Language Recognition (SLR).

1 Introduction

Artificial Intelligence (AI) is increasingly applied across various fields [1,2]. Sign language is vital for communication within the deaf community [3–5]. Over time, it has evolved into complete languages across cultures. Sign Language Recognition (SLR) aims to translate gestures into text or speech, supporting communication between deaf and hearing individuals.

Received: 23 May 2025/ Accepted: 07 November 2025
□ Asmaa G. Seliem, asmaseliem90@gmail.com
Shaimaa Mohamed Elembay, shimaa.mohamed@eng.mti.edu.eg,
Mohamed Nasser Elshayeb, melshayeb508@gmail.com
1.Modern University for information and technology
2.Faculty of Engineering, Biomedical department, at Modern University for Technology & Information (MTI)

Hearing loss affects an estimated 466 million people globally, presenting significant challenges in communication, social integration, and access to essential services [6]. Sign language is not universal, with different standards existing across countries, such as the notable differences between Egyptian and Libyan Sign Language. Additionally, regional variations, akin to accents or slang, further complicate understanding.

Misunderstandings can have severe consequences, such as in legal situations or during medical consultations. Hearing-impaired individuals often face discrimination during job applications and interviews. Recruiters may find it cumbersome to accommodate their needs, leading to feelings of neglect. Telephone interviews are nearly impossible without an interpreter, and in-person interviews can be challenging if the interviewer is unprepared. Moreover, deaf individuals are twice as likely to suffer from psychological issues such as depression and anxiety, primarily due to feelings of isolation. Most deaf children are born to hearing parents, yet regular use of sign language within these families remains limited. As Tegan Howell et al. [7] reported, only a small percentage of Australian families with deaf or hard-of-hearing children use sign language at home, contributing to social and emotional isolation among these children.

Recent advancements in deep learning and computer vision have shown great promise in enhancing SLR systems. Dakhli and Bakari [8] demonstrated how integrating both manual and non-manual components—such as facial expressions and body posture—significantly improves recognition accuracy. Complementing this, Zhang and Jiang [9] provided a comprehensive overview of cutting-edge deep learning approaches, including CNNs, RNNs, and Transformers, that are advancing the capabilities of modern SLR systems. Studies by Padden and Humphries highlight the cultural significance of sign languages and the need for inclusive communication tools, while the World Health Organization emphasizes the global impact of hearing loss and the importance of accessible communication technologies [10].

As highlighted by Madhiarasan and Roy [11], understanding the variety of sign language modalities and datasets was essential for building accurate and inclusive

SLR frameworks. Najib [12] emphasized how machine learning techniques are being leveraged to interpret sign language in real time, reducing communication barriers for the deaf community. Bansal et al. [13] further explored the role of intelligent systems and nature-inspired algorithms in enhancing the performance and adaptability of modern SLR solutions. These developments validate the ongoing need for robust and socially responsive SLR technologies to support the inclusion and well-being of deaf individuals.

Leveraging the power of assistive technology can significantly enhance the quality of life for individuals with disabilities by opening up new opportunities and expanding their range of options.

Significant advancements in AI have led to the development of powerful tools and frameworks for image and sign language recognition. Deep learning libraries such as TensorFlow and PyTorch provide robust platforms for training and deploying convolutional and temporal neural models, while OpenCV remains a cornerstone for image preprocessing and real-time video analysis. Pretrained models like MediaPipe and OpenPose enable efficient hand and body pose estimation. Additionally, recent hybrid deep-learning approaches that combine spatial and temporal cues, such as those reviewed by Buttar et al. [14], had significantly improved static and dynamic sign recognition. A comprehensive, 25-year survey of Continuous Sign Language Recognition (CSLR) [15] emphasized the critical role of multimodal cues—particularly non-manual features such as facial expressions and body posture—in enhancing system performance. Notably, Hirooka et al. [16] introduced a Stack Spatial Temporal Transformer Network that captures hierarchical spatial and temporal dependencies across multiple sign languages, pushing the boundaries of cross-cultural recognition accuracy and efficiency.

This research introduces an AI-powered system designed to support real-time communication for deaf and mute communities, offering combined solutions that go beyond previous research, which primarily focused on sign language image recognition and provided only partial solutions to the communication challenges faced by deaf individuals. The system is implemented as a web-based application, offering a low-cost and accessible solution for translating Arabic and English sign languages into text—and vice versa. Unlike many existing systems that rely on specialized gloves or visual markers, the proposed approach uses only a standard webcam to capture hand and body movements, allowing for natural, markerless interaction using bare hands. Through the integration of computer vision, machine learning, and deep neural networks, the system processes video input in real time to recognize sign gestures and convert them into spoken or written language. Conversely, it translates voice or text input into sign language using either animated avatars or sequences of gesture images.

The novelty of this work lies in its real-time, bilingual design that supports both Arabic and English sign languages, enabling bidirectional translation between sign, text, and speech. Static signs (e.g., letters) are classified using SVM, while dynamic gesture sequences (e.g., full words) are handled using LSTM networks. This combination allows for robust recognition of both alphabetic signs and temporally dependent gestures.

To validate our model selection, we briefly experimented with alternative classifiers such as CNNs and GRUs; however, they introduced higher complexity without significant performance gains in our setup. Therefore, SVM and LSTM were retained due to their balance of accuracy and computational efficiency for static and sequential sign recognition, respectively.

Due to dataset limitations, Arabic translation currently supports 25 commonly used words and the full alphabet, while the English component includes the complete alphabet. By bridging the gap in bilingual SLR systems and eliminating the need for wearable sensors, this system offers a novel and practical solution for inclusive, multimodal communication—particularly in Arabic-speaking regions where such tools are scarce.

The remainder of this paper is organized as follows: Section 2 presents the material and method, starting with dataset and data structure. Then, describe the proposed system architecture in detail, including the image processing pipeline, recognition workflow, and the AI models used. Section 3 results and discusses evaluates the performance of the models. Finally, Section 4 concludes the study and outlines directions for future work.

2 Martial and Methods

2.1 The dataset used

The Arabic sign language dataset used in this study is the Arabic Alphabets Sign Language Dataset (ArASL), published by Ghazanfar Latif, Jaafar Alghazo, Nazeeruddin Mohammad, Roaa AlKhalaf, and Rawan AlKhalaf. This dataset, comprising 54,049 images of Arabic sign letters, was sourced from Kaggle and is available on Mendeley Data [17]. For English sign language, we gathered 46,032 images of English letters, primarily using a computer camera and OpenCV technology. This extensive dataset was crucial for training and developing our sign language learning system.

The dataset structure is organized such that each image is represented as a row in a CSV file. The first column contains the label corresponding to the sign (i.e., the letter or word), and the remaining columns store the extracted hand landmark coordinates (x, y, z) as a flattened list. If no landmarks are detected in a frame, all corresponding values are recorded as zeros to preserve structural consistency. This standardized format enables seamless input into machine learning models. The Arabic dataset

comprises 54,049 labeled images across all alphabetic signs, while the English dataset includes 46,032 labeled images captured via webcam. Each class is balanced to ensure fair training and evaluation.

2.2 Data Structure

Table 1 illustrates the data structure for Arabic letters (it is a part of the data). Each row in the dataset.csv file corresponds to an image, with the first column indicating the label (gesture category) and the subsequent columns containing the flattened list of hand landmarks. If no landmarks are detected, the columns contain zeros.

Table 1 Arabic letters Data structure for the proposed algorithm

ain	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ain	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ain	-0.101	0.655	0.373	-0.109	0.649	0.331	-0.121	0.641
ain	0.000	0.00	0.00	0.00	0.00	0.00	0.00	0.00
ain	-0.124	0.669	0.404	-0.131	0.662	0.360	-0.141	0.652
ain	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ain	-0.123	0.673	0.402	-0.131	0.662	0.362	-0.138	0.649
ain	-0.117	0.674	0.389	-0.124	0.663	0.355	-0.131	0.656
ain	-0.120	0.669	0.395	-0.128	0.658	0.352	-0.140	0.650
ain	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ain	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ain	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ain	-0.111	0.677	0.381	-0.119	0.667	0.339	-0.130	0.660
ain	-0.104	0.705	0.380	-0.114	0.697	0.334	-0.128	0.680
ain	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
ain	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

The system incorporates robust error handling to ensure the integrity and consistency of the dataset. Specifically, when no hand landmarks are detected in an image, the code automatically fills in zeros for the corresponding data points. This approach prevents the disruption of the dataset structure and ensures that each entry maintains a uniform format, regardless of detection success as illustrated in **Table 2**.

This method effectively manages potential errors and maintains the reliability of the dataset for subsequent processing and analysis. Several libraries were utilized for processing and managing the data. The `cv2` (OpenCV) library played a crucial role in image processing tasks, such as reading images (`cv2.imread`), converting color spaces (`cv2.cvtColor`), and flipping images (`cv2.flip`). The `mediapipe` library provided ready-to-use machine learning solutions, particularly for hand tracking (`mediapipe.solutions.hands`). The `os` module facilitated interaction with the operating system, enabling directory navigation and file path handling. Finally, the built-in `csv` module was employed for reading and writing CSV files, ensuring efficient data management and manipulation.

Table 2 Handles potential errors (part of data)

ain	-0.101	0.655	0.373	-0.109	0.649	0.331	-0.12	0.64
ain	-0.124	0.669	0.404	-0.131	0.662	0.360	-0.14	0.65
ain	-0.123	0.673	0.402	-0.131	0.662	0.362	-0.13	0.64
ain	-0.117	0.674	0.389	-0.124	0.663	0.355	-0.13	0.65
ain	-0.120	0.669	0.395	-0.128	0.658	0.352	-0.14	0.65
ain	-0.111	0.677	0.381	-0.119	0.667	0.339	-0.13	0.66
ain	-0.104	0.705	0.380	-0.114	0.697	0.334	-0.12	0.68

2.3 Proposed system

The proposed SLR system incorporates three distinct interaction modes designed to enhance communication for the deaf community. These include voice-to-sign translation, text-to-sign conversion through an avatar-based interface, and real-time sign recognition from video input. By integrating voice, text, and gesture-based inputs, the system facilitates multimodal communication and promotes greater accessibility for individuals who are deaf or hard of hearing. The overall framework is illustrated in **Fig. 1**.

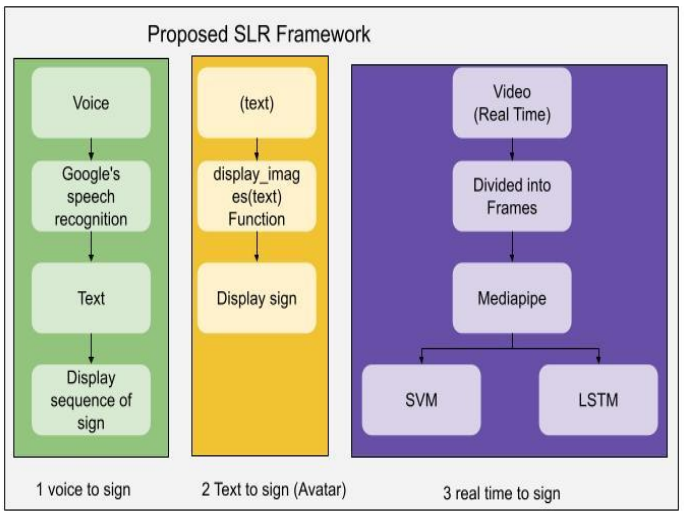


Fig. 1 Proposed framework

2.4 Image processing workflow

The image processing workflow involves several key functions to ensure accurate and consistent data extraction, as shown in **Fig. 2**. The process begins with reading the image from a specified file path using OpenCV. The image is then preprocessed by converting it from BGR to RGB color space and flipping it horizontally for uniform orientation. A MediaPipe Hands instance is initialized, configured to process static images with specific parameters. The preprocessed image is then processed using this MediaPipe Hands instance. During processing, the system checks for the detection of hand landmarks, extracting their x, y, and z coordinates into a list. If no landmarks are found, the list is filled with zeros to maintain data consistency. Finally, the MediaPipe Hands instance is closed, releasing the resources it used to ensure efficient resource management.

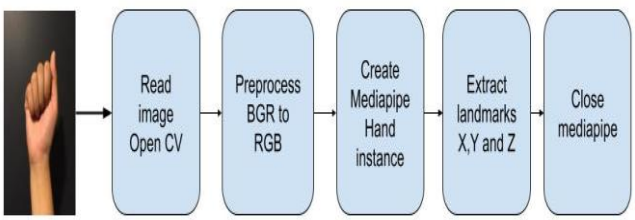


Fig. 2 Image processing workflow

2.5 Hand gesture recognition for alphabetic characters

In this study, the process of hand gesture recognition for alphabetic characters involved several key stages. Initially, data collection was performed using a computer camera and OpenCV to capture images of the English alphabet. These images were then subjected to preprocessing, wherein hand landmarks were extracted using the MediaPipe library. The extracted landmarks, along with their corresponding labels, were compiled into a CSV file for subsequent analysis.

Data cleaning followed, where null values were removed from the dataset and the data was represented as a data frame using the pandas library. For model training, an SVM classifier was employed. This classifier was trained on features derived from the spatial coordinates of hand landmarks (x, y) present in the CSV file. The trained model demonstrated a recognition accuracy of approximately 99% on both training and testing datasets, with additional performance metrics including recall, precision, and F1 score.

Real-time detection was implemented using a web camera that captured a sequence of frames. Each frame was processed using the MediaPipe framework, specifically its Palm Detection and Hands models, to detect hand positions and extract landmarks. If hands were detected, the landmarks were drawn on the frame using the “mp.drawing.draw_landmarks” function, and the processed frame was displayed using cv2.imshow. The detection loop continued until the 'Esc' key was pressed, at which point resources were released.

For Arabic Sign Language (ArASL) alphabets, the process was similar, with the data collected from the "ArASL_Database_54K_Final" available on Kaggle. The application offered three interactive options: 'S' to add the predicted letter to the current string, 'D' to delete the last letter, and 'Esc' to quit the application.

2.6 Hand gesture recognition for words

This study details the methodology employed for recognizing hand gestures corresponding to Arabic words through a series of systematic steps as presented in **Fig. 3**.

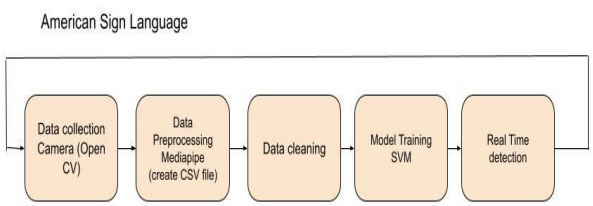


Fig. 3 Hand recognition workflow

1. Data Collection: The data collection process involves real-time video capture using a webcam. The camera streams a sequence of frames, which are processed by the MediaPipe library. Specifically, landmarks from the upper part of the body, indexed from 0 to 22 in MediaPipe, are used to train the Arabic word recognition system. The data collection process includes creating directories for each action and sequence to organize and prevent overwriting of existing data. The video capture loop iterates through each action and sequence, capturing frames, processing them with the mediapipe_detection function, and drawing landmarks using the draw_landmarks function. Key points for pose, face, and both hands are extracted from the results. If any landmarks are absent, zeros are appended to ensure data consistency. These key points are then flattened into a 1D array and saved as .npy files. The processed frames, with drawn landmarks, are displayed in real-time, and the loop terminates when the 'q' key is pressed.

2. Data Preprocessing: Data preprocessing involves several key steps using MediaPipe models. Initially, images are converted from BGR to RGB format, as MediaPipe models require RGB input. To optimize performance, the image is temporarily set as non-writable during model processing. The MediaPipe model processes the image to detect landmarks, and the results are returned. After processing, the image's writeability is restored and converted back to BGR format for further use with OpenCV. The extract_keypoints function then extracts key points from various body parts, including pose landmarks, face landmarks, left hand landmarks, and right-hand landmarks. If no landmarks for the left or right hand are detected, an array of zeros with a shape of 21x3 is returned. All extracted key points are concatenated into a single array.

3. Model Training: The model training phase utilizes Long LSTM networks, which are particularly effective for processing and understanding sequences due to their ability to retain long-term dependencies and capture temporal dynamics. LSTM models are well-suited for sign language recognition, where the sequence and timing of gestures are crucial. The LSTM model is trained on a dataset of labeled sign language gestures, learning to map input sequences to their corresponding labels. After training, the model is saved to preserve the learned weights.

4. Model Evaluation: The performance of the trained model is evaluated by computing training and testing accuracies. A confusion matrix is also plotted to visually assess the model's performance across different classes.

5. Real-Time Recognition: Real-time recognition involves several post-processing steps to ensure accurate and efficient gesture detection. Initially, images are converted from BGR to RGB format as MediaPipe models expect RGB input. To enhance performance, images are set as non-writable during model processing. After processing, images are converted back to BGR format, for consistent display with OpenCV. Drawing utilities from MediaPipe are initialized to annotate landmarks on the images.

The real-time inference process consists of:

I. **Initialization:** Opening the webcam and initializing the MediaPipe holistic model.

II. **Frame Capture:** Capturing a frame from the webcam and processing it to detect landmarks.

III. **Landmark Detection:** Detecting and drawing face, pose, and hand landmarks on the frame.

IV. **Prediction:** Extracting key points from the detected landmarks and using them to make predictions with the pre-trained model. Consistent predictions are added to the sentence.

V. **Visualization:** Displaying the predicted action on the video feed and showing the frame in a window.

VI. **Termination:** The loop continues until the 'q' key is pressed, after which resources are released and windows are closed.

2.7 Using AI for the proposed system

2.7.1 Support vector machine

Support Vector Machines are a supervised learning technique used for classification by finding a hyperplane that maximizes the margin between two classes. When applied to multiclass classification, SVMs utilize strategies such as One-vs-One (OvO) and One-vs-All (OvA) to handle multiple classes effectively, as shown in **Fig. 4**.

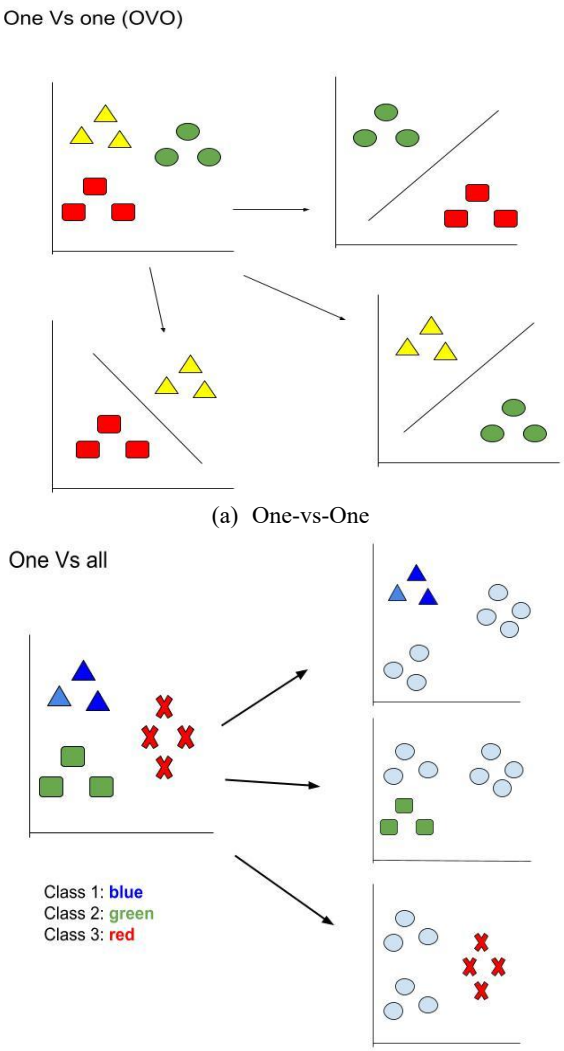
SVMs employ different kernels to suit various types of data. The Linear Kernel is computationally efficient and performs well when the number of features is large compared to the number of samples. However, it has a limited ability to capture complex, non-linear relationships within the data. In contrast, the Radial Basis Function (RBF) Kernel is suitable for non-linearly separable data, as it can transform the input space into a higher-dimensional space where classes might become separable by a hyperplane.

The choice between kernels depends on the nature of the data. Linear kernels are advantageous when the data appears linearly separable or when there are many features relative to the number of samples, making them faster to train and evaluate. On the other hand, RBF kernels are better suited for data with complex, non-linear relationships, though they require more computational resources and are slower due to the calculation of pairwise distances in high-dimensional spaces.

2.7.2 LSTM (working Idea)

Long Short-Term Memory (LSTM) networks are a type of RNN designed to address the vanishing gradient problem and capture long-term dependencies in sequential data, as shown in **Figure 7**. The LSTM architecture includes a chain structure with four neural networks and various memory blocks called cells.

LSTMs use backpropagation through time to adjust parameters based on the error between predicted and actual outputs. This approach allows gradients to flow through multiple time steps, enabling the network to learn from experiences and refine its predictions accordingly.



(a) One-vs-One

(b) One-vs-All

Fig. 4 Multiclass classification tasks using different strategies

3 Results and Discussion

The real-time performance of the proposed SLR system is influenced by several practical factors, including latency, ambient lighting, and camera quality. Latency primarily depends on the processing time required for image acquisition, landmark extraction, and model inference. In our implementation, we utilized a standard laptop webcam (720p resolution, 30 fps) under typical indoor lighting conditions. While the system maintained stable performance with minimal delay, variable lighting—especially low-light or high-glare environments,

was observed to affect the accuracy of hand and body landmark detection. Additionally, lower-resolution or low-frame-rate cameras may hinder precise tracking of dynamic gestures. To mitigate these issues, basic preprocessing techniques such as brightness normalization and landmark smoothing were applied. For future deployments, incorporating adaptive exposure control, lightweight model quantization, and GPU acceleration can further reduce latency and improve system responsiveness in diverse real-time scenarios.

3.1 Voice and text to sign language

The microphone can be activated to capture spoken language, which is then translated into sign language sequences represented by a series of images illustrating individual letters, as shown in **Fig. 5**. Additionally, written text can be converted into sign language through the use of an avatar, as depicted in **Fig. 6**.

Fig. 5 Web page used in translating spoken language into sign language as sequence of several images represents English letters

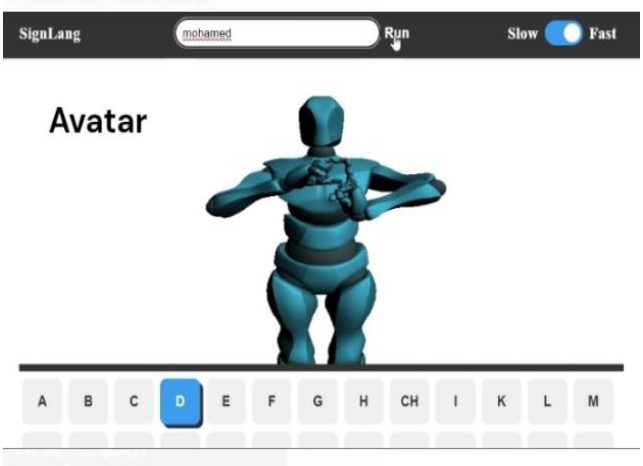


Fig. 6 Web page for Text-to-Sign Language translation through animated avatars or video sequences

3.2 Sign language to text in real time

a) English letters result

In this subsection, we present the outcomes of our English letter detection algorithm. Various samples of detected English letters are illustrated in **Fig. 7**. These examples demonstrate the accuracy and effectiveness of our system in recognizing and translating spoken and written inputs into corresponding sign language letters. Each sample in **Fig. 7** highlights the system's capability to accurately capture and represent the nuances of different letters, showcasing the robustness of our detection methodology.

Fig. 7 Detection of English letters in some samples.

In **Fig. 8**, a screenshot of a webpage is displayed. This webpage is designed to detect English sign language for the phrase “Hello world” and convert it into written English words and sentences, as well as into an audio format.

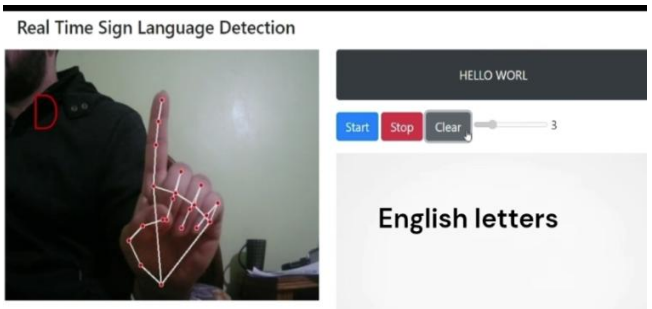


Fig. 8 Screen shoot of a webpage during detection of a sentence (“Hello World”), sample for entering Letter D in world

The model's accuracy using a Support Vector Machine (SVM) with a linear kernel achieved 99.95% during training and 98.26% during testing. **Figure 9** presents the confusion matrix for the testing phase, detailing the performance of the model in classifying English letters. The matrix provides a comprehensive overview of the model's predictive capabilities, illustrating the number of correct and incorrect predictions for each letter, thus allowing for a detailed assessment of its accuracy and areas for potential improvement.

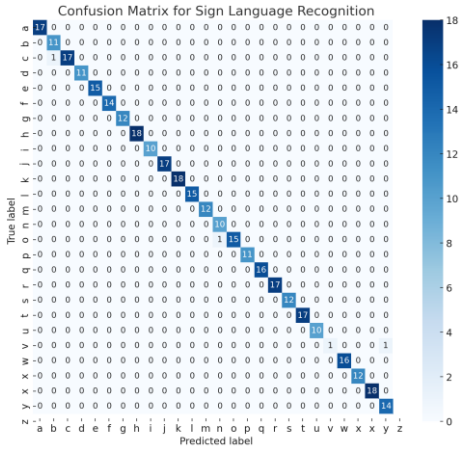


Fig. 9 Confusion matrix for testing the detection of English letters

b) Arabic letters results

Some samples of Arabic alphabet detection are shown in **Fig. 10**.

Fig. 10 Detection of the Arabic alphabet in some samples

The SVM model with a linear kernel achieved an accuracy of 97.75% during training and 92.24% during testing. Similarly, the SVM model with a Radial Basis Function (RBF) kernel achieved 96.65% training accuracy and 91.54% testing accuracy. The confusion matrix obtained during the testing phase for Arabic letter detection is presented in **Fig. 11**.

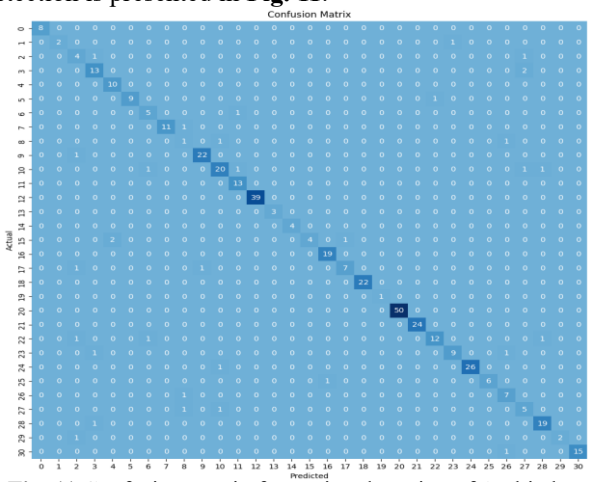
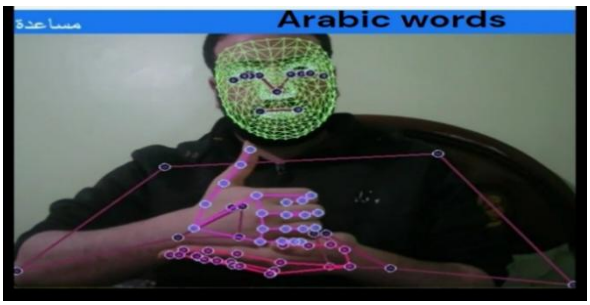


Fig. 11 Confusion matrix for testing detection of Arabic letters

c) Arabic words result

In the context of word detection, a significant challenge lies in whether a single frame or image of the upper body can accurately represent one or more words. To address this, we utilized MediaPipe to prepare upper-body images as depicted in **Fig. 12**. Initially, the LSTM model exhibited suboptimal accuracy when trained on a dataset comprising 25 words. To enhance performance, the dataset was reduced to include only 8 words. This adjustment resulted in an improved accuracy of 95.95% during training and 93.75% during testing. **Figure 13** illustrates the confusion matrix for the testing phase, providing a detailed analysis of the model's performance in detecting some of these 8 Arabic words. This matrix offers insights into the classification accuracy and highlights the specific areas where the model performs well and where it requires further refinement.



(a) Sample for Arabic words "مساعدة"

(b) Sample for Arabic words "أهلا"

Fig. 12 Detection of Arabic words

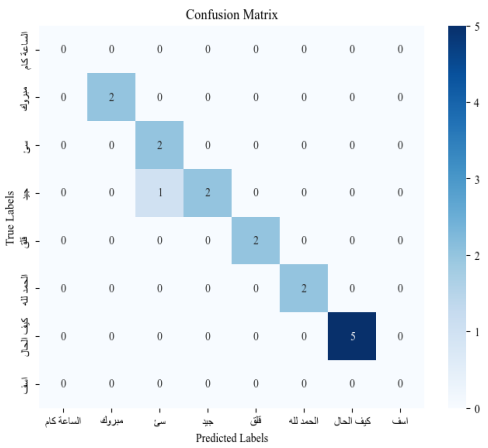
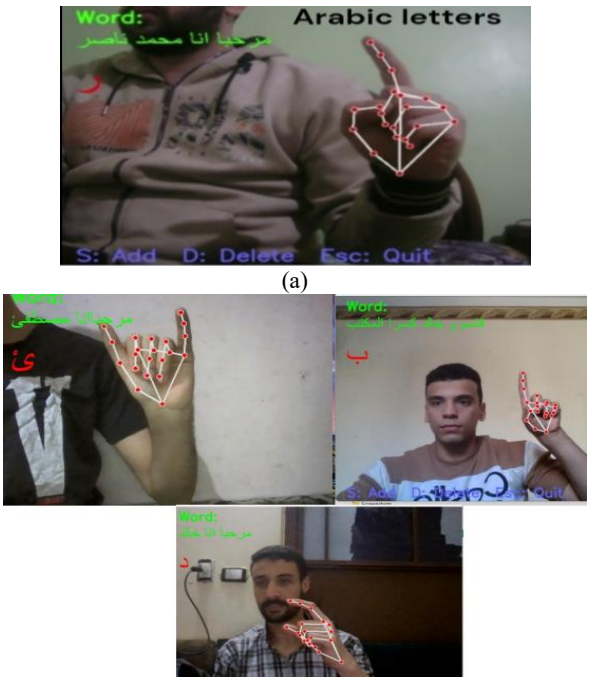


Fig. 13 Detection Confusion matrix of 8 Arabic words

d) Sign to fully Arabic sentences

The proposed system translates sign language into complete Arabic sentences. This feature allows users to convert individual signs into fully formed, grammatically correct sentences in Arabic, enhancing communication accuracy and efficiency. **Figure 14** illustrates this process in detail, showcasing how the system processes and integrates each sign into coherent and contextually appropriate sentences.



(a)

(b)

Fig. 14 Detection of Arabic sentences.

Despite its strong performance within the defined scope, the current system has limitations in scalability and gesture complexity. The Arabic vocabulary is limited to 25 predefined words, and the system focuses on isolated rather than continuous signs. Additionally, more nuanced gestures involving facial expressions, finger spelling, or overlapping hand movements are not yet supported. To address these challenges, future work will explore the use of continuous sign language datasets, the incorporation of non-manual features (e.g., facial and head movements), and the implementation of advanced deep learning models such as Transformers and Graph Neural Networks (GNNs) to improve the modeling of spatial-temporal dependencies and broaden vocabulary coverage.

To better evaluate the effectiveness and novelty of our proposed bilingual sign language recognition system, we conducted a comparative analysis with recent studies in the field. **Table 3** summarizes key aspects of these studies, including the methods used, datasets, application types, and achieved accuracy. This comparison highlights the strengths of our system in terms of real-time performance, bilingual support, and the ability to handle both static and dynamic gestures using lightweight architecture. The proposed MediaPipe + SVM/LSTM approach demonstrates competitive performance compared to prior studies. Our system achieved the highest English letter recognition accuracy (98.26%) among the compared works, outperforming previous CNN-based and CNN-LSTM models. For Arabic letters (92.24%) and Arabic words (93.75%), the results remain robust despite the increased complexity and diversity of the dataset, as our model addresses a broader range of classes in a real-time, web-based environment with avatar integration. While prior research such as Rouabhi et al. (2024) reported 95.5% accuracy on ArSL2018 and Alani et al. (2021) achieved ~94% using MobileNetV2-A, our results indicate that integrating MediaPipe-based feature extraction with hybrid classifiers can yield superior or comparable accuracy. Slightly lower accuracy in Arabic letter recognition is attributed to dataset variability and will be targeted in future optimization efforts.

Table 3 Comparison of Recent Arabic Sign Language Recognition Systems with the Proposed Approach

Study	Model	Dataset	Accuracy	Application
Rouabhi et al. (2024) [18]	Pre-trained CNN	ArSL2018	95.5%	Mobile app (real-time)
Alani et al. (2021)[19]	CNN-LSTM	ArSL2018	~ 94%	MobileNetV 2-A
This Study	MediaPipe + SVM/LSTM	ArASL_5 4K & custom English dataset	98.26% (EN), 92.24% (AR letters), 93.75% (AR words)	Web-based, real-time & Avatar

4 Conclusion

This research has demonstrated the effectiveness of integrating advanced machine learning models and MediaPipe for real-time SLR and translation. The system efficiently translates spoken language into sign language sequences and written text into sign language using animated avatars. The detection of English letters achieved remarkable accuracy, using a linear kernel that reached 99.95% during training and 98.26% during testing. Similarly, the Arabic letters detection model exhibited high accuracy, with the SVM using a linear kernel achieving 97.75% in training and 92.24% in testing, and the Radial Basis Function kernel model achieving 96.65% in training and 91.54% in testing. Despite these successes, challenges such as the inability to consolidate the research into a single webpage and reduced accuracy with larger Arabic word datasets were noted. To address these issues, future work will focus on enhancing model performance through fine-tuning, data augmentation, and exploring advanced techniques such as Transformers, Temporal Convolutional Networks (TCNs), and Graph Neural Networks (GNNs).

Additionally, this system holds significant potential for real-world deployment in various domains such as education, healthcare, and public services, where real-time sign language interpretation can improve communication accessibility for the Deaf and Hard of Hearing (DHH) community. By promoting inclusive communication, the proposed system contributes to social equity and the broader adoption of assistive technologies.

References

[1] M. Abdelsattar, A. A. A. Rasslan, and A. Emad-Eldeen, "Detecting dusty and clean photovoltaic surfaces using MobileNet variants for image classification," SVU-International Journal of Engineering Sciences and Applications, vol. 6, no. 1, pp. 9–18, Jun. 2025.

[2] M. Abdelsattar, A. AbdelMoety, and A. Emad-Eldeen, "Comparative analysis of machine learning techniques for fault detection in solar panel systems," SVU-International Journal of Engineering Sciences and Applications, vol. 5, no. 2, pp. 140–152, 2024.

[3] M. Papatsimouli, P. Sarigiannidis, and G. F. Fragulis, "A survey of advancements in real-time sign language translators: integration with IoT technology," Technologies, vol. 11, p. 83, 2023.

[4] L. Al Khuzayem, S. Shafi, S. Aljahdali, R. Alkhamesie, and O. Alzamzami, "Efhamni: A Deep Learning-Based Saudi Sign Language Recognition Application," Sensors, vol. 24, p. 3112, 2024.

[5] I. Papastratis, C. Chatzikonstantinou, D. Konstantinidis, K. Dimitropoulos, and P. Daras, "Artificial intelligence technologies for sign language," Sensors, vol. 21, p. 5843, 2021.

[6] K. A. Alshehri, W. M. Alqulayti, B. E. Yaghmoor, and H. Alem, "Public awareness of ear health and hearing loss in Jeddah, Saudi Arabia," South African journal of communication disorders, vol. 66, pp. 1-6, 2019.

[7] *Tegan Howell, Valerie Sung, Libby Smith , Shani Dettman . Australian families of deaf and hard-of-hearing children: Are they using sign?2024 International Journal of Pediatric Otorhinolaryngology*, 385, 104.

[8] Dakhli, A., & Bakari, W. *Deep learning-based sign language recognition system using both manual and non manual components fusion*. 2023 *AIMS Mathematics*, 9(1), 2105–2122.

[9] Zhang, Y., & Jiang, X. *Recent advances on deep learning for sign language recognition*. 2024 *Computer Modeling in Engineering & Sciences*, 139(3), 2399–2450.

[10] WHO Regional Office for the Western Pacific, *WHO Regional Office for the Western Pacific*. World Health Organization, 2023. [Online]. Available: https://www.who.int/westernpacific. [Accessed: Dec. 12, 2023].

[11] Madhiarasan M, Roy PP. A comprehensive review of sign language recognition: Different types, modalities, and datasets. arXiv preprint arXiv:2204.03328. 2022 Apr 7.

[12] F. M. Najib, "Sign language interpretation using machine learning and artificial intelligence," *Neural Computing and Applications*, vol. 37, pp. 841–857, 2025.

[13] S. Bansal, A. Tyagi, and R. K. Goel, "A Deep Survey of Intelligent Systems for Sign Language Recognition System," in *Role of Nature-Inspired Algorithms in Real World Applications*, Springer, 2025.

[14] A. M. Buttar, U. Ahmad, A. H. Gumaei, A. Assiri, M. A. Akbar, and B. F. Alkhamees, "Deep Learning in Sign Language Recognition: A Hybrid Approach for the Recognition of Static and Dynamic Signs," *Mathematics*, vol. 11, no. 17, p. 3729, Aug. 2023.

[15] Sarah N. Alyami, H. Luqman, M. Hammoudeh, "Reviewing 25 years of continuous sign language recognition research: Advances, challenges, and prospects," *Information Processing & Management*, vol. 61, no. 5, p. 103774, Sep. 2024.

[16] K. Hirooka, A. S. M. Miah, T. Murakami, Y. Akiba, Y. S. Hwang, and J. Shin, "Stack Transformer Based Spatial-Temporal Attention Model for Dynamic Multi-Culture Sign Language Recognition," *arXiv*, Mar. 2025.

[17] G. Latif, J. Alghazo, N. Mohammad, R. AlKhalaf, and R. AlKhalaf, *Arabic Alphabets Sign Language Dataset (ArASL)*, Mendeley Data, V1, 2019. DOI: 10.17632/y7pckrw6z2.1

[18] Rouabhi, S., Aissani, D., & Hadjilef, A. (2024). Real-time Mobile Application for Arabic Sign Alphabet Recognition Using Pre-trained CNN. Soft Computing.

[19] Houssem Lahiani, Mondher Frikha. A Comparative Study of Two Deep learning Architectures for Gesture Recognition on ArSL2018 Dataset, 25 June 2024, PREPRINT (Version 1) available at Research Square