CommIT Journal 20(2), 363–377, 2026

Dynamic Sign Language Recognition: A Hybrid Approach Combining MediaPipe and LSTM

Youssef Farhan[1*], Zineb Haimer[2], Oumaima Lamaamar[3], and Abdessalam Ait Madi[4]
[1−4]Advanced Systems Engineering Laboratory, National School of Applied Sciences, Ibn Tofail University
Kenitra, Morocco 14000
Email: [1]youssef.farhan@uit.ac.ma, [2]zineb.haimer@uit.ac.ma, [3]oumaima.lamaamar@uit.ac.ma, [4]abdessalam.aitmadi@uit.ac.ma

***Abstract*—The inability of the general population to understand sign language creates serious communication barriers for hearing-impaired individuals in critical situations such as healthcare emergencies, educational settings, and workplace environments. Addressing this gap requires effective and accurate automated recognition systems. This proposed research presents a real-time American Sign Language (ASL) recognition system for 24 dynamic signs that integrates the MediaPipe frame work with Long Short-Term Memory (LSTM) network. To enhance performance while reducing computational complexity, only the most relevant features are extracted from self-recorded dynamic sign videos: coordinates of 65 key hand and body landmarks, complemented by 41 engineered angle features between joint connections and 26 engineered distance features between specific landmark pairs, yielding a compact 285-feature representation per frame. As a result, LSTM network handles spatiotemporal sequence modeling across 25-frame sequences, effectively capturing the dynamic nature of sign language gestures. The proposed system achieves 98% test accuracy, with precision, recall, and F1-score of 98% across 720 test samples. It also successfully interprets all 24 dynamic signs in real-time testing scenarios using a standard webcam, including visually similar sign pairs such as "Good"/"Bad" and "Mother"/"Not", demonstrating its practical applicability for assistive communication technologies. The research contribution lies in the systematic integration of MediaPipe Holistic with strategic feature engineering (combining landmark coordinates with computed angles and distances) and LSTM modeling to achieve efficient real-time dynamic sign recognition with reduced computational and data requirements.**

***Index Terms*—American Sign Language, Dynamic Signs, MediaPipe, Holistic Model, Long Short-Term Memory (LSTM)**

Received: Aug. 09, 2025; received in revised form: Dec. 12, 2025; accepted: Dec. 22, 2025; available online: Sep. 03, 2026.
*Corresponding Author

I. Introduction

COMMUNICATION is necessary in everyday life because it enables people to express opinions, share ideas, and participate in society [1]. However, communicating clearly can be more difficult for hearing-impaired individuals. Hence, Sign Language (SL) is extremely important in improving these people's lives by allowing them to communicate with others and exchange knowledge and feelings. But, a communication gap exists between hearing-impaired individuals and the general population due to the general population's lack of familiarity with SL [2]. To overcome this gap, solutions such as a rapid sign-to-text recognition system, similar to the previous work [3], should be developed to assist this population in engaging and integrating into society. This research expands upon the previous research [4]. In this present research, the subject is explored, and additional explanations and analyses are presented.

SL recognition plays an essential role in bridging communication gaps and fostering inclusivity for individuals with hearing impairments [5, 6]. Its significance extends across various facets of life, including equal access to information, education, healthcare, and employment, and it is necessary to ensure safety during emergencies [5, 7]. By achieving high accuracy in real-time recognition of SL, automated systems can break down communication barriers and empower hearing-impaired individuals to participate fully in society.

More than 300 SLs are in use worldwide [8], including British SL, Japanese SL, Arabic SL, Brazilian SL, and American SL (ASL), and each presents regional and/or national differences [9]. The development of a universal SL recognition system is complicated by this linguistic diversity. Although progress has been made in developing automated translation systems for certain

SLs [10, 11], significant challenges persist due to the complexity of SL grammar and the wide variety of signing styles that different individuals use. Nevertheless, advances in computer vision and machine learning have the potential to enhance communication by improving the efficiency and accuracy of SL recognition systems.

Since ASL is the most widely used SL [12], it is reasonable to focus on it. Static and dynamic signs are the two forms of ASL [13]. Static signs are simpler to locate and recognize because they do not involve movement. In contrast, dynamic signs require detection of movement and are much more difficult to recognize. Their identification necessitates tracking all sign movements over a sequence of frames and analyzing each of these frames before interpreting the sign. Despite the difficulty in dynamic signs, recognizing them is necessary because they represent a major part of ASL vocabulary. The research aims to address the real-time recognition of dynamic signs by developing a system capable of accurately identifying a set of commonly used dynamic signs in ASL.

Many researchers have been investigating various methods to overcome the challenges associated with ASL recognition since the 1980s [14]. Early approaches relied on sensor gloves for tracking hand gestures [15], which suffer from hardware dependency, discomfort, and high costs. Traditional methods using Hidden Markov Models (HMMs) for dynamic sign language recognition through multi-sensor data fusion [16]. Recent studies have combined Convolutional Neural Network (CNN) and Long Short-Term Memory (LSTM) networks for dynamic sign recognition [17–19]. They have used CNN for feature extraction and LSTM for classification. More recently, the MediaPipe framework has been applied to extract landmark coordinates, which are then processed using various neural networks including One-Dimensional (1D) CNN [20], Recurrent Neural Network (RNN) [21, 22], and LSTM [23]. Additionally, distances and angles are calculated using the landmark coordinates acquired via the MediaPipe framework [24], but this is only done for static signs. Nevertheless, existing approaches face limitations including dependency on large datasets, complex processing requirements, sensitivity to environmental factors, and ineffective feature generation. This research addresses these limitations by using MediaPipe to extract relevant landmark coordinates and developing two additional engineered features, distances and angles, to enhance recognition performance while reducing dataset requirements and environmental sensitivity.

Dynamic sign recognition requires the acquisition and processing of sequential information and features to effectively identify a sign [25, 26]. However, the presence of irrelevant information, such as background, skin tone, and colors can significantly decrease model accuracy and require considerable time and effort for training [26]. As a solution, only the features relevant to dynamic sign recognition are extracted, discarding all other irrelevant information. This method not only enhances the model's accuracy but also reduces training time and computational effort, making it an efficient approach for dynamic sign recognition. Furthermore, this solution allows for the construction of more lightweight and portable models that can be used in real-time systems, providing greater access to dynamic sign recognition technologies.

Effective implementation of the dynamic sign identification process requires integration of various tools and methods. A critical component is the use of a suitable framework capable of accurately determining the coordinates of key landmarks on the human body. For this purpose, the MediaPipe framework, specifically designed for such tasks [27], is selected. After obtaining these coordinates, they are used as input to construct and train an LSTM model. The LSTM network is a type of RNN that excels at processing sequential data [28, 29], such as the sequential features extracted from dynamic signs. The model is trained on a large dataset of labeled dynamic signs. The dataset is created by recording all sign gestures performed by the first author using a standard webcam, after which the videos are annotated and organized for model training. This process enables the model to learn the temporal patterns and discriminative features required for accurate sign identification.

The main contributions of the research include the following:

1) Systematic feature engineering for dynamic signs: Development of a structured feature engineering pipeline tailored for dynamic sign recognition, combining an optimized subset of 65 landmarks with 41 angle features and 26 distance features to construct a compact and discriminative 285-feature representation suitable for temporal sequence modeling.
2) Efficiency optimization: Achievement of 98% recognition accuracy using only 150 videos per sign, demonstrating substantially lower data requirements compared to deep learning methods that typically rely on large-scale datasets [18, 19].
3) Real-time validation and deployability: Development of a complete inference pipeline capable of performing feature extraction and sign interpretation in real-time on a standard consumer-grade webcam, confirming practical suitability for

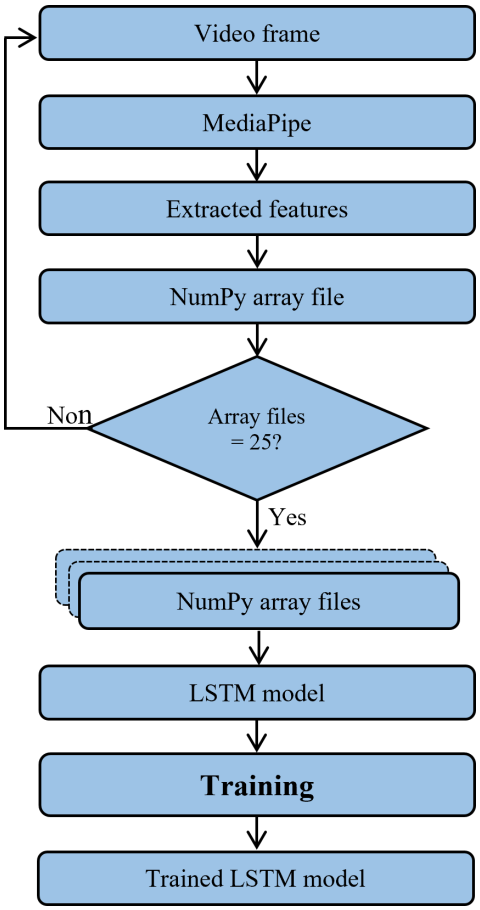


Fig. 1. Training process. Note: Long Short-Term Memory (LSTM).

assistive communication applications.

4) Comprehensive dynamic sign recognition: Accurate classification of 24 frequently used ASL dynamic signs, validated across both controlled test data and real-world evaluation scenarios.

II. Research Method

A. Proposed Training and Testing Process

Figure 1 presents a summary of the model's development and training approach. The model is built in stages, including recording video of dynamic signs, extracting the coordinates of custom landmark features using MediaPipe, calculating angles and distances, and saving all these data in a NumPy array. Then, these data are used to train the LSTM network and obtain the trained model.

Following the training phase, the model goes through the testing phase. The first step evaluates the model's performance metrics using test data and determines the number of correct and incorrect predictions. In the second step, a standard computer camera is used to evaluate the model's response and accuracy in real-world situations. These steps are presented in Fig. 2.

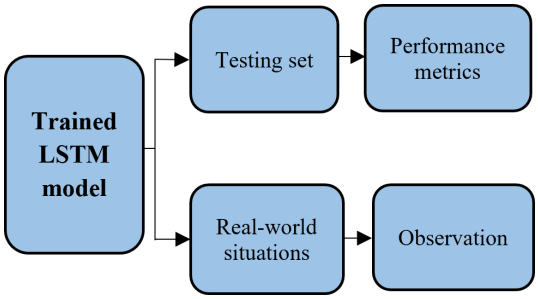


Fig. 2. Testing process. Note: Long Short-Term Memory (LSTM).

To provide a systematic and reproducible description of the methodology, Algorithm 1 formalizes the complete pipeline in pseudocode format. The algorithm is organized into four phases: data preparation (feature extraction from 150 videos per sign and dataset splitting), training (LSTM network training with validation data), testing (metrics evaluation on reserved test data), and inference (real-time recognition with webcam). This algorithmic representation complements Figs. 1 and 2, providing a detailed step-by-step description that enhances reproducibility.

B. Proposed Model Architecture

To recognize a sign and determine its corresponding name, it is necessary to track multiple frame features. It requires sequential analysis of grouped data, with each group containing relevant features. Due to the sequential nature of the data processing task, the LSTM network is selected as the optimal approach for execution. The network is built with 4 LSTM layers, 4 dense layers, and an output layer containing 24 neurons, corresponding to the total number of dynamic signs to be recognized. Table I presents a summary of the network layers and the number of parameters to be trained.

In addition to the training parameters, the model requires additional hyperparameter adjustments before starting the training process. The hyperparameters are factors that control the learning process, and their values can have a significant impact on training performances and so must be appropriately adjusted. In the research the dataset is divided into two portions: 80% of the data is used for training, comprising 120 videos for each sign and ensuring an adequate sample for effective training, with the remaining 30 videos for testing.

The input layer neurons are set to match the number of frames, specifically 25, as elaborated in the subsequent section. Likewise, the output layer neurons

Algorithm 1. Dynamic Sign Language Recognition Pipeline

INPUT: Video frames for training/testing
OUTPUT: Recognized sign label

Phase 1: Data Preparation
For each sign s in vocabulary (24 signs):
  For each video v in dataset (150 videos per sign):
    Record 25 frames
    For each frame f:
      Extract 65 landmark coordinates using Holistic model
      Compute 41 angles (Equations (1)–(2), Table III)
      Compute 26 distances (Equation (3), Table IV)
      Concatenate into 285-feature vector
      Save 285-feature vector as a separate .npy file
    End For
  End For
End For
Split dataset: training, validation and testing (as specified in Table II)

Phase 2: Training
Split training + validation data (as specified in Table II)
Train LSTM network with:
  Training data: 102 videos per sign
  Validation data: 18 videos per sign
  Input: sequences of 25 frames × 285 features
  Architecture: 4 LSTM layers + 5 Dense layers
  Output: 24 classes (one per sign)
  Hyperparameters: as specified in Table II

Phase 3: Testing (Metrics Evaluation)
For each video v in test set (30 videos per sign):
  Load 25 .npy files (one per frame)
  For each frame f:
    Extract 285-feature vector from .npy file
  End For
  Feed sequence to trained LSTM model
  Apply Softmax to get probability distribution
  Compare predicted label with ground truth
  Update performance metrics (accuracy, precision, recall)
End For
Report overall test accuracy

Phase 4: Inference (Real-Time Recognition)
Capture 25 frames from webcam in real-time
For each frame f:
  Extract 65 landmark coordinates using Holistic model
  Compute 41 angles and 26 distances
  Concatenate into 285-feature vector
End For
Feed sequence to trained LSTM model
Apply Softmax to get probability distribution
Return sign label with highest probability
Display recognized sign on screen

TABLE I
MODEL SUMMARY.

Layer (type)	Output Shape	Parameters
LSTM 1 (LSTM)	(None, 25, 64)	89,600
BN 1 (BatchNormalization)	(None, 25, 64)	256
LSTM 2 (LSTM)	(None, 25, 128)	98,816
Dropout 1 (Dropout)	(None, 25, 128)	0
LSTM 3 (LSTM)	(None, 25, 128)	131,584
BN 2 (BatchNormalization)	(None, 25, 128)	512
LSTM 4 (LSTM)	(None, 64)	49,408
BN 3 (BatchNormalization)	(None, 64)	256
Dense 1 (Dense)	(None, 256)	16,640
Dropout 2 (Dropout)	(None, 256)	0
Dense 2 (Dense)	(None, 128)	32,896
BN 4 (BatchNormalization)	(None, 128)	512
Dense 3 (Dense)	(None, 64)	8,256
BN 5 (BatchNormalization)	(None, 64)	256
Dense 4 (Dense)	(None, 128)	8,320
Dropout 3 (Dropout)	(None, 128)	0
Output (Dense)	(None, 24)	3,096
Total params		440,408 (1.68 MB)
Trainable params		439,512 (1.68 MB)
Non-trainable params		896 (3.50 KB)

Note: Batch Normalization (BN) and Long Short-Term Memory (LSTM).

TABLE II
THE FINAL HYPERPARAMETERS.

Hyperparameter	Value
Testing data	20% (30 videos)
Training and validation data	80% (120 videos)
Training data	85% (102 videos)
Validation data	15% (18 videos)
Sequence length – input layer	25
Hidden layers + neurons per layer	LSTM – 128 neurons
	LSTM – 128 neurons
	LSTM – 64 neurons
	Dense – 256 neurons
	Dense – 128 neurons
	Dense – 64 neurons
	Dense – 128 neurons
Target length – output layer	24
Activation function (input/hidden layers)	ReLU
Activation function (output layer)	Softmax
Dropout	Yes
Batch normalization	Yes
Optimizer	Adam
Batch size	32 (default value)
Epoch	400

Note: Rectified Linear Unit (ReLU) and Long Short-Term Memory (LSTM).

correspond to the number of dynamic signs to be detected, totaling 24. Because of their significance, the Dropout and Batch Normalization techniques are used to improve model performance. The epoch count is set to 400 after performing some initial training runs, striking a balance to avoid both overfitting and underfitting. Table II summarizes all the hyperparameters used.

C. Dataset

The collection and utilization of training and testing datasets are critical factors in computer vision applications. Effective dataset construction requires careful consideration of data quality, quantity, and representativeness to ensure robust model performance. This section explains the dataset collection process, which involves three main stages: recording primary video data of dynamic signs, engineering relevant features through landmark extraction and mathematical computations, and consolidating these features into efficient array structures for model training.

For primary dataset, to capture the entire movement of each dynamic sign, a short video consisting of 25 frames is filmed. Following a series of experiments, it has been determined that 25 frames are the best number to ensure that the entire sign gesture is correctly captured. If the number of frames is increased, the

recording may continue even after the sign movement is complete, resulting in unnecessary additional information. In contrast, reducing the number of frames may cause the recording to stop before the sign movement is completed, resulting in incomplete data. As a result, 25 frames are selected as an appropriate number to ensure that the recorded videos capture the full range of motion for each sign.

The goal of this research is to interpret 24 different dynamic signs commonly used in everyday conversation by developing a computer vision model. These signs include "Hello," "Thanks," "Yes," "No," "Father," "Mother," "Fine," "Please," "Forget," "Busy," "Happy," "Sad," "Finish," "Go," "Help," "More," "Not," "Correct," "Wrong," "Bus," "Bad," "Good," "School," and "Book." A total of 150 videos are recorded for each sign, producing a total of 3,600 videos.

Next, feature engineering, the process of selecting and transforming raw data into meaningful inputs for machine learning models, forms a critical component of this SL recognition system. This phase focuses on extracting only those features from dynamic sign videos that are essential for accurate sign identification. While recorded SL videos contain abundant information, only specific elements, such as hand movements, finger positions, and spatial relationships between hands and face, contribute meaningfully to sign recognition. Conversely, features like skin color, background elements, and clothing characteristics, despite their visual prominence, provide negligible value for dynamic sign identification. By systematically filtering out these non-contributory elements and preserving only recognition-relevant features, two significant advantages are achieved: enhanced model accuracy through focused learning on relevant information and substantially improved training efficiency by eliminating computational overhead from processing irrelevant data.

The MediaPipe framework is used for this feature extraction process as it is specifically designed to detect and track precise coordinate points on the human body. Figures 3 and 4 represent the landmarks topology for the hand model and pose model, respectively. Recognizing that SL encompasses more than isolated hand movements, the Holistic model is implemented, which integrates three specialized models to simultaneously capture and track landmark coordinates across face, hands, and body posture [30].

A strategic reduction in the landmark set is implemented after careful analysis of feature contribution to sign differentiation. Facial landmarks from the face model (468 points in total) are excluded because facial features demonstrate minimal variation between different signs. Similarly, landmarks below the waist

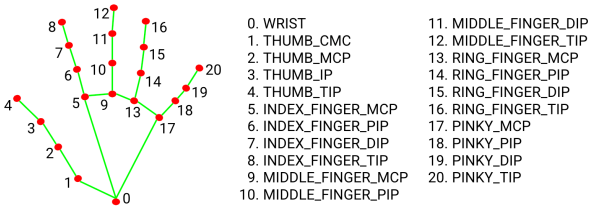


Fig. 3. Landmarks of hand model [31].

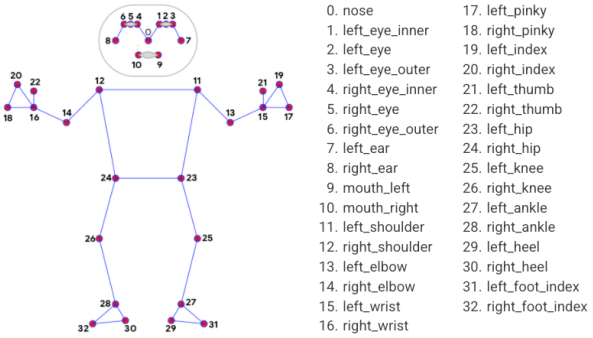


Fig. 4. Landmarks of pose model [32].

are eliminated as SL movements are rarely influenced by lower body positioning. The most informative body points are retained in the optimized landmark selection. Specifically, all hand landmarks are preserved along with selected pose model landmarks corresponding to arms, hands, shoulders, and face. This refinement results in a focused set of 65 landmarks: 23 from the pose model and 42 from the combined right- and left-hand models.

The spatial coordinates of these landmarks are extracted using MediaPipe alongside custom-developed functions. For pose landmarks, 4 values (x, y, z, and visibility) are captured, while 3 values (x, y, z) are required for hand landmarks, yielding a total of 218 coordinate values. However, landmark coordinates alone are insufficient for robust dynamic sign recognition. Additional descriptive derived features are necessary to effectively capture the temporal and relational aspects of SL movements.

When making various dynamic signs, certain body parts stretch or bend, causing variations in the angles formed by the body's landmarks. This dynamic feature becomes a valuable asset for sign recognition because it provides a clearer understanding of the sign's meaning. Distances between landmarks are also important in sign recognition. These distances vary between signs, making them a useful feature in sign identification. The system's ability to accurately interpret a wide range of dynamic signs is improved by extracting and analyzing both angle changes and distance variations.

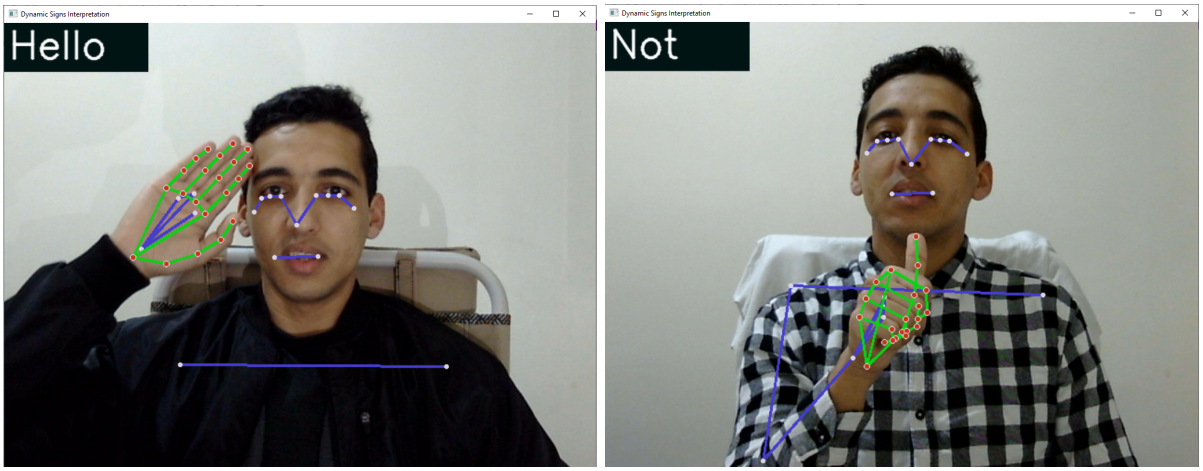


Fig. 5. Angles and distances' comparison between two sign frame.

To highlight these distinctions, an example is presented in Fig. 5 which compares a frame with the "Hello" sign to one with the "Not" sign. As shown, the hand structure and finger arrangement differ, resulting in differences in the angles and distances between hand landmarks. Similarly, the elbow of the hand changes shape with each sign, resulting in distinct angle and distance measurements. For this reason, angles and distances have been added as new features derived from the extracted coordinates.

Using the (x, y) coordinates of three selected landmarks, angles are computed based on the mathematical formulas presented in Eqs. (1) and (2). The specific sets of the three landmarks used for this calculation are listed in Table III, resulting in 41 angle values in total. The angles range from -180° to 180° values, significantly larger than the normalized coordinate values extracted by MediaPipe, which range from 0 and 1 [30]. The angles are also normalized to align more closely with these coordinate scales. To achieve this, each angle value is divided by 180, restricting the normalized angles to a range between -1 and 1. Regarding distances, the distance between two landmarks is computed based on their (x, y) coordinates according to the mathematical formula presented in Eq. (3). The pairs of landmarks used for these distance calculations are listed in Table IV and result in 26 distance values in total. The equations are:

$$\widehat{ABC}_{\text{rad}} = \text{atan2}(y_C - y_B,\, x_C - x_B) - \text{atan2}(y_A - y_B,\, x_A - x_B), \quad (1)$$

$$\widehat{ABC}_{\text{deg}} = \widehat{ABC}_{\text{rad}} \times \frac{180}{\pi}, \quad (2)$$

$$d_{AB} = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}, \quad (3)$$

where (x_A, y_A), (x_B, y_B), and (x_C, y_C) denote the 2D coordinates of three selected landmarks A, B, and C, with B representing the vertex of the angle in Eqs. (1) and (2). The same coordinate notation for landmarks A and B is used in the distance formula in Eq. (3).

TABLE III
LANDMARKS USED FOR CALCULATION OF ANGLES.

Pose		Hands	
No.	Landmarks	No.	Landmarks
1	[1, 0, 4]	1	[2, 3, 4]
2	[1, 3, 7]	2	[5, 6, 7]
3	[4, 6, 8]	3	[6, 7, 8]
4	[11, 12, 14]	4	[9, 10, 11]
5	[12, 11, 13]	5	[10, 11, 12]
6	[12, 14, 16]	6	[13, 14, 15]
7	[11, 13, 15]	7	[14, 15, 16]
8	[22, 16, 18]	8	[17, 18, 19]
9	[21, 15, 17]	9	[18, 19, 20]
10	[19, 11, 7]	10	[4, 0, 8]
11	[20, 12, 8]	11	[8, 0, 20]
		12	[16, 17, 20]
		13	[8, 5, 12]
		14	[4, 5, 20]
		15	[8, 13, 20]

TABLE IV
LANDMARKS USED FOR CALCULATION OF DISTANCES.

Pose		Hands	
No.	Landmarks	No.	Landmarks
1	[0, 7]	1	[0, 4]
2	[0, 8]	2	[0, 8]
3	[0, 20]	3	[0, 12]
4	[0, 19]	4	[0, 16]
5	[7, 8]	5	[0, 20]
6	[12, 20]	6	[4, 8]
7	[11, 19]	7	[4, 20]
8	[11, 12]	8	[8, 20]
9	[15, 16]		
10	[19, 20]		

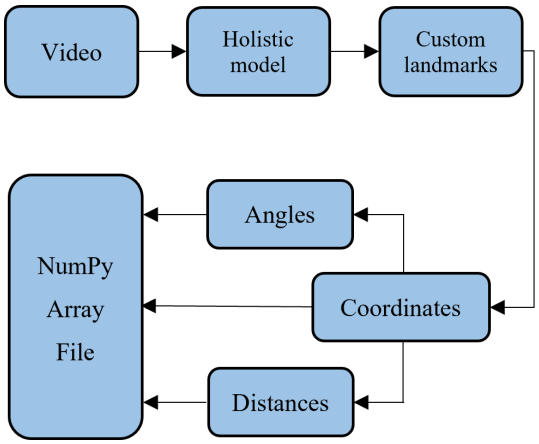


Fig. 6. Dataset collection process.

D. Dataset Collection

All extracted features, including the coordinates of custom landmarks, angles, and distances, are consolidated into a single array file using the NumPy library. This array contains only the relevant features for dynamic sign recognition, excluding redundant or extraneous information. Consequently, these NumPy arrays offer a more efficient and compact representation compared to storing the entire video as they retain only the most informative data. Each array file stores a total of 285 features. The overall process is illustrated in Fig. 6.

A custom Python script automates the entire process: recording dynamic signs, identifying and extracting custom landmark coordinates, calculating angles and distances based on extracted coordinates, and compiling and saving all features into a single file. This automation ensures seamless execution in real time. This process is accomplished while recording a video for each dynamic sign. It means that the previous frame's features are already saved in the NumPy array before the next frame is captured.

III. Results and Discussion

The research objective is to interpret 24 commonly used dynamic signs in ASL in real-time, using the MediaPipe framework and an LSTM network. However, the MediaPipe framework encounters challenges in detecting hands in horizontal or vertical positions, which commonly occur in signs such as "Help," "Father," or "Mother," where the hands are tilted slightly. Despite these difficulties, the project achieves significant progress although distinction between the signs "Good" and "Bad" as well as "Mother" and "Not" remains challenging due to their similarities. Additionally, the dynamic sign "Bus" is generated by combining the signs for "B," "U," and "S" to form the complete sign for "Bus." This adds complexity to the project, as it requires the model to recognize the sequence of signs correctly.

The training process of the LSTM network is the key to achieving precise and reliable results for dynamic sign recognition. A network architecture with a large number of trainable parameters (440,408) is built with the objective of identifying 24 dynamic signs. This network is optimized using the Adam optimizer and normalized using batch normalization throughout the training phase to ensure that the model can understand and generalize well to new data. The network is able to successfully capture and absorb the pertinent features of the dynamic signs through this process, which results in high accuracy during testing.

To ensure optimal performance, the model is trained over 400 iterations (epochs). The loss value and categorical accuracy for both the training and validation stages are recorded at each epoch to monitor model performance and prevent overfitting. The training process shows stable convergence with minimal fluctuation in later epochs. Following training, the model is evaluated with testing data as well as with real-time videos captured with a standard webcam.

A. Simulation Results

The model's training and validation results are shown in Figures 7 and 8. Figure 7 depicts the variation in loss over the epochs, indicating that the loss value began at about 0.3 and gradually decreased with fluctuations. The validation loss fluctuates significantly and eventually reaches a value of 0.1376, whereas the training loss decreases significantly and stabilizes at 0.056 by the end of the training phase. Figure 8 illustrates the variation in categorical accuracy during the training and validation processes. Categorical accuracy begins at around 20% and gradually increased with fluctuations, similar to the loss variation. At the end of the training phase, training categorical accuracy reaches 99.84%, while validation categorical accuracy ultimately achieves a value of 97.69%. These results demonstrate successful model convergence and effective learning of the dynamic sign patterns.

Despite having 440,408 parameters, the model's training time is relatively short, taking less than 27 minutes, which is an impressive achievement when compared to the training times of similar complex neural networks. This efficient process is made possible by the use of a feature reduction technique that filtered out irrelevant information, leaving only the features needed for accurate dynamic sign recognition. With such a short training time, the training process can be

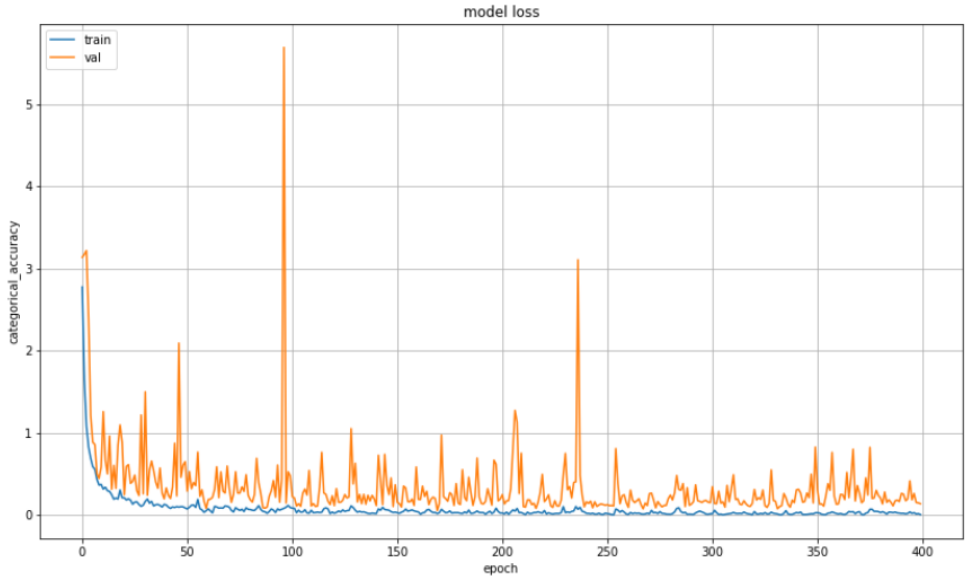


Fig. 7. Loss value during training.

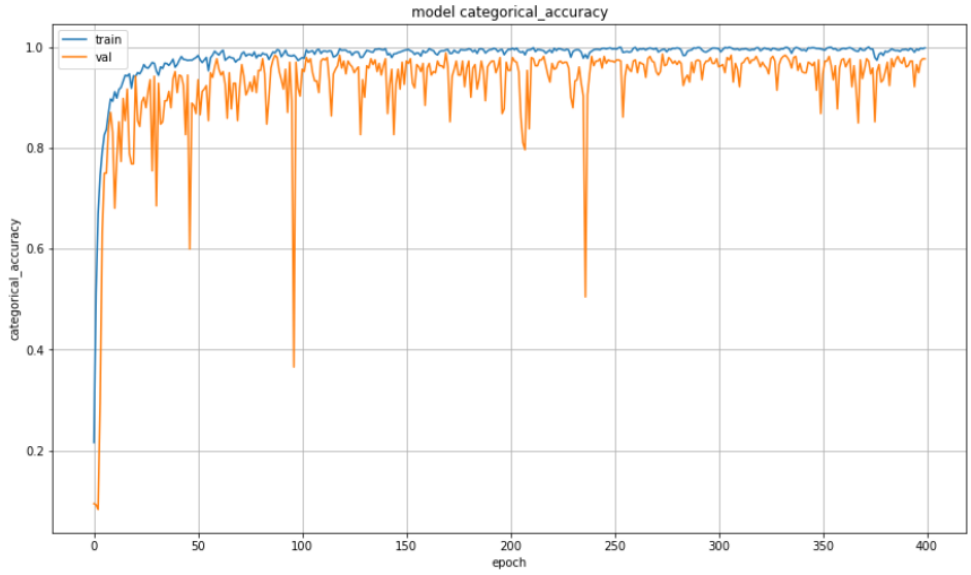


Fig. 8. Accuracy value during training.

repeated with different hyperparameters to achieve the desired results if necessary. Furthermore, if the training is carried out on a computer with greater performance than the i7-6700 CPU @ 3.40GHz and 16.0 GB of RAM used for this training, the training time can be reduced even further.

Once the model finishes training, its effectiveness is assessed using the test dataset. Each dynamic sign’s precision, recall, and F1-score values are calculated, with the results given in Table V. The support column shows the number of test samples randomly sampled for each sign, with sample sizes ranging from 22 to 38 samples per sign. The results demonstrate excellent performance across all metrics, with overall accuracy reaching 98% on 720 test samples. Analysis of individual sign performance reveals that several signs achieve perfect precision scores of 1.00, including “Hello,” “Thanks,” “Yes,” “No,” “Please,” “Happy,” “Bus,” “Good,” and “Book.” The recall values also present a strong performance, with most signs achieving scores above 0.96. The lowest performing sign is “Book” with a recall of 0.89, while “Wrong” shows relatively

TABLE V
RESULTS OF PERFORMANCE EVALUATION.

Sign	Precision	Recall	F1-Score	Support
Hello	1.00	1.00	1.00	33
Thanks	1.00	1.00	1.00	30
Yes	1.00	1.00	1.00	35
No	1.00	1.00	1.00	34
Father	1.00	0.96	0.98	26
Mother	0.96	0.96	0.96	28
Fine	0.97	1.00	0.99	38
Please	1.00	1.00	1.00	38
Forget	1.00	0.96	0.98	25
Busy	1.00	1.00	1.00	26
Happy	1.00	1.00	1.00	30
Sad	1.00	1.00	1.00	24
Finish	1.00	0.97	0.99	34
Go	1.00	1.00	1.00	29
Help	0.92	1.00	0.96	34
More	0.97	1.00	0.98	29
Not	0.93	0.97	0.95	29
Correct	1.00	0.96	0.98	24
Wrong	0.90	0.86	0.88	22
Bus	1.00	1.00	1.00	31
Bad	0.97	0.97	0.97	29
Good	1.00	1.00	1.00	34
School	0.91	0.97	0.94	30
Book	1.00	0.89	0.94	28
Accuracy			0.98	720
Macro avg.	0.98	0.98	0.98	720
Weighted avg.	0.98	0.98	0.98	720

lower precision at 0.90 and recall at 0.86, consistent with the confusion patterns observed in the confusion matrix. The macro and weighted average values for precision, recall, and F1-score are all 98%, confirming the model's robust and balanced performance across all 24 dynamic signs.

A confusion matrix, which shows the number of correctly and incorrectly classified examples, is created to provide an effective visual representation of the test results. Figure 9 presents this matrix and gives a more comprehensive look at how well the model performs for each dynamic sign. The matrix demonstrates that most signs achieve perfect or near-perfect classification, with the majority of predictions falling along the diagonal. However, minor misclassifications are observed, primarily with the "Wrong" sign, which is occasionally confused with "Correct" (8 instances) and "Not" (2 instances), likely due to similar hand movements. Other notable confusions include "Father" being misclassified once as "Mother" and "Book" showing occasional confusion with "Good." These results suggest these sign pairs share overlapping gestural characteristics.

Considering that it is trained to recognize 24 dynamic signs, the model achieves excellent test accuracy of 98.0%. This remarkable accuracy and the results in the confusion matrix are achieved by removing non-essential features such as background and colors and focusing solely on the essential features, notably hand and upper body pose landmark coordinates obtained with MediaPipe. Furthermore, two key features are extracted from these landmarks and used in sign recognition: distances and angles. This method is extremely effective in allowing the model to accurately classify the signs. By extracting only 285 relevant features from each video frame, the model not only achieves high accuracy but also significantly reduces training time. To evaluate the significance of this performance, a comprehensive comparison with recent state-of-the-art systems is presented in the following section.

B. Performance Comparison with State-of-the-Art Systems

To contextualize the results and demonstrate the significance of the 98% test accuracy achieved, the proposed system's performance is compared with recent state-of-the-art SL recognition approaches that employ similar methodologies. Table VI presents a comprehensive comparison of the proposed system with recent works published between 2023 and 2025. The proposed system achieves competitive performance relative to recent MediaPipe-based LSTM models across multiple SLs. Guruprakash et al. [33] have reported 96.75% accuracy on 44 isolated Indian SL signs, while Ridwang et al. [34] have achieved 97.1% on 36 alphanumeric signs from Indonesian SL. Uddin et al. [35] have obtained 95% accuracy on 11 numerical Norwegian SL signs. Khartheesvar et al. [36] have achieved 87.4% accuracy on the large-scale INCLUDE dataset comprising 263 word-level signs and 4,292 videos, reflecting the challenges associated with larger vocabularies. When compared to hybrid deep learning approaches, the proposed system maintains superior performance while using a significantly simpler architecture. Buttar et al. [37] have reported 96.5% accuracy using a YOLOv6 + LSTM model for continuous ASL signs, and Baihan et al. [38] have achieved 93.5% accuracy using a CNN-LSTM-Attention model on 26 isolated signs. Hassan et al. [39] have focused on static ASL recognition using a DenseNet-based architecture trained on more than 64,000 images.

The consistently high performance of the MediaPipe + LSTM approaches across these previous studies validates the robustness of this architectural framework for SL recognition. The proposed system's 98% accuracy on 24 dynamic ASL signs represents a strong balance between vocabulary size and recognition accuracy. An important trade-off observed across these systems is that larger vocabularies (such as Khartheesvar's 263 signs) typically result in lower accuracy (87.4%), while smaller vocabularies achieve higher performance. The proposed system strikes a practical balance with 24

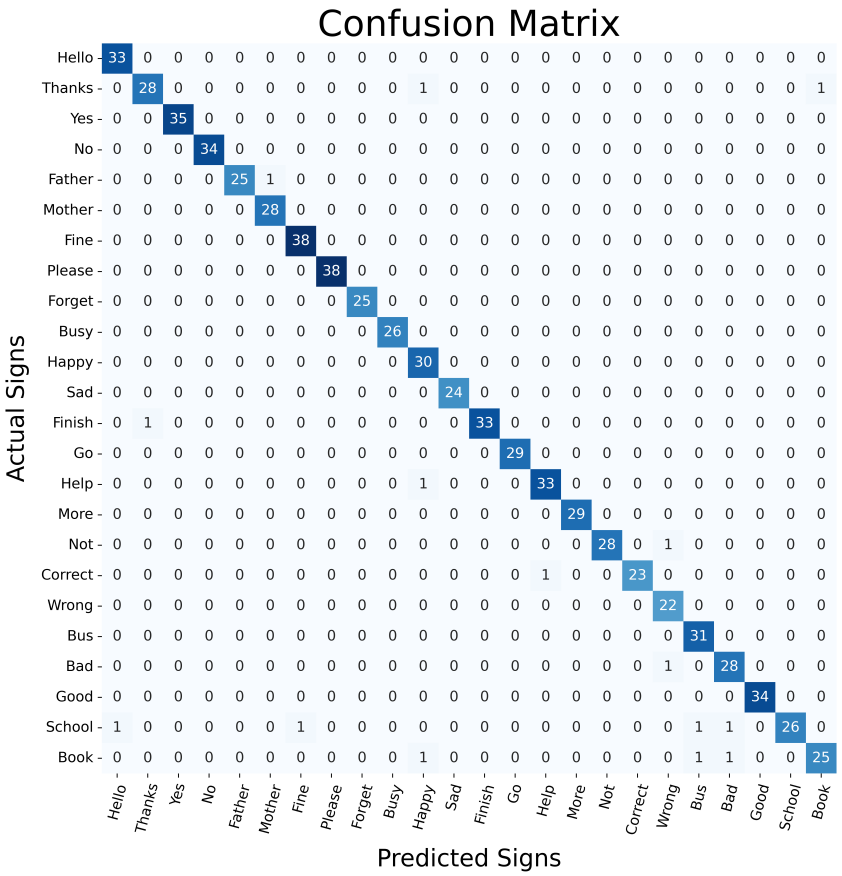


Fig. 9. Confusion matrix.

TABLE VI
PERFORMANCE COMPARISON WITH RECENT SIGN LANGUAGE RECOGNITION SYSTEMS.

Study	Method	Sign Language	Dataset Size	Signs	Accuracy
The proposed system	MediaPipe + LSTM	ASL	3,600 videos	24 (dynamic)	98.00%
[33]	MediaPipe + LSTM	Indian SL	N/A	44 (isolated)	96.75%
[36]	MediaPipe + LSTM	Indian SL	INCLUDE dataset (4,292 videos)	263 (word-level)	87.40%
[34]	MediaPipe + Modified LSTM	Indonesian SL	Custom dataset	36 (alphanumeric)	97.10%
[35]	MediaPipe + LSTM	Norwegian SL	Educational videos	11 (numbers 0–10)	95.00%
[37]	YOLOv6 + LSTM	ASL	Custom dataset	Continuous signs	96.50%
[38]	CNN-LSTM-Attention	ASL	Public datasets	26 (isolated)	93.50%
[39]	SNDA (DenseNet)	ASL	64,000+ images	24 (static)	N/A

Note: Long Short-Term Memory (LSTM), American Sign Language (ASL), Convolutional Neural Network (CNN), Sign Language (SL), and Sign Nevestro Densenet Attention (SNDA).

commonly used dynamic signs being identified with 98% accuracy.

Regarding dataset considerations, the proposed system uses 3,600 videos, which translates to approximately 90,000 frames when accounting for the 25-frame sequence length per video. This represents a moderate dataset size that achieves strong performance without requiring extensive data collection efforts. The strategic feature engineering approach, extracting only 285 relevant features per frame (65 landmarks + 41 angles + 26 distances), enables the system to learn discriminative patterns efficiently while maintaining real-time processing capability. While direct comparison across systems is limited by differences in datasets, sign types, and evaluation protocols, the proposed system demonstrates competitive results, supported by high precision, recall, and F1-score and validated through real-time operation on standard hardware.

C. Real-Time Results

The accuracy of the proposed model in real-world situations is tested using a standard webcam. In real-



Fig. 10. Real-time testing

time testing, the name of the recognized sign is displayed on the screen in a box placed in the upper left corner. Figure 10 depicts various examples of real-time testing, with the detected sign name appearing at the top left corner of the screen. In one case, the beginning of the “Thanks” movement is accurately identified by the model, which displays its name in the black box at the top left. In addition to correctly interpreting it, the detection threshold is set to 99.9%. It means that the system detects and translates only dynamic signs with detection scores close to 100%, demonstrating the model’s high accuracy.

The model performs well in real-world testing, correctly identifying all 24 dynamic signs without any false detections and displaying their names in the designated box on the screen. Furthermore, despite the difficulty of identifying 24 dynamic signs, some of which are highly similar, such as “Good” and “Bad” or “Mother” and “Not,” the model’s recognition of the signs is correct, demonstrating its effectiveness in real-world situations. This challenge is addressed using the Batch Normalization technique and by adding more data to the training process, resulting in reduced confusion between similar signs and improved model performance. It is important to note that this high performance is also attributed to the extracted relevant features, such as distances and angles.

Moreover, an additional challenge for the model is created by including the dynamic sign “Bus.” This sign is formed by three individual signs, “B,” “U,” and “S,” which are signed sequentially to form the complete sign for “Bus.” As a result, the model needs to correctly recognize the sequence of signs to detect this dynamic sign. This increases the model’s complexity and makes it a more challenging issue to solve. However, the model demonstrates a remarkable degree of precision in identifying the “Bus” sign during testing. Overall, the model’s successful recognition of the “Bus” sign demonstrates its ability to tackle even the most complex dynamic sign recognition tasks.

D. Limitations and Scalability of the Model

The proposed model, which combines the MediaPipe Holistic model and an LSTM network, has demonstrated impressive results in the real-time recognition of 24 dynamic ASL signs. The key advantages

of this approach are its ability to extract only the relevant features (coordinates, angles, and distances) while filtering out irrelevant information like background, skin tone, and clothing, leading to improved accuracy and efficiency. The use of MediaPipe for feature extraction is particularly advantageous as it is a language-agnostic method that relies on universal body landmarks rather than language-specific features. This foundational aspect of the model suggests that it can be adapted to recognize a wider variety of signs across different SLs with minimal modifications to the core architecture.

To achieve this scalability, several strategies can be used. A hierarchical classification approach can be implemented whereby signs are grouped into categories, allowing a two-stage classification process that first identifies the category before recognizing the specific sign. Additionally, transfer learning techniques can be leveraged to expand the system's vocabulary by fine-tuning the model with new sign data without requiring complete retraining. This is particularly useful when incorporating signs from different SLs, where grammar and sign execution vary significantly.

Furthermore, the modular nature of the model supports the development of specialized models for different categories of signs, such as numbers, letters, and common phrases, which can then be integrated to form a comprehensive recognition system. Scaling the model to accommodate other SLs like British SL will involve recalibrating the feature extraction process and retraining the network with language-specific datasets. Additionally, exploring advanced machine learning paradigms like few-shot learning, which allows a model to learn from a very small number of examples, or meta-learning, where the model improves its learning ability over multiple learning episodes, can enable rapid adaptation to new signs or languages with limited training data. However, successfully scaling the model will likely necessitate a larger and more diverse dataset, along with potential adjustments to the model architecture to maintain high accuracy across an expanded range of signs and languages. By incorporating these strategies, the proposed model can evolve into a robust, scalable solution capable of interpreting a wide variety of signs, making it a valuable tool for a broader spectrum of users.

The ability of the MediaPipe framework to recognize body landmarks plays a crucial role in the accuracy of dynamic sign recognition. Though it generally performs well in this regard, identifying hand landmarks in challenging positions, such as horizontal orientation, can be difficult. Nevertheless, this is a rare occurrence that does not significantly impact the model's overall performance. One challenge observed during the system's development is detecting hands in horizontal or vertical positions, particularly in signs such as "Help," "Father," or "Mother." To address this, the hands are slightly tilted to ensure that the model can accurately recognize these signs.

During real-time testing, it is observed that the model occasionally exhibits minor limitations when interpreting successive dynamic signs. Although this issue does not significantly impact overall performance, it results in slight delays when transitioning between certain signs. This latency stems from the model's processing time in recognizing certain dynamic signs, which can be a disadvantage in applications requiring rapid detection of continuous signing. The potential cause of this delay is the computational cost of processing video data in real time. To address this limitation, the main approach is to explore the use of more efficient hardware accelerators, such as Graphics Processing Units (GPUs), to accelerate the processing of video data. Additionally, model compression techniques such as model pruning, which involves reducing the complexity of a model by removing less important parameters or knowledge distillation and training a smaller model to replicate the performance of a larger model, can be explored to reduce the model's computational requirements without sacrificing accuracy. By addressing this limitation, the aim is to create a system for SL recognition that is not only accurate but also responsive and capable of interpreting entire sentences rather than individual dynamic signs.

IV. Conclusion

The communication barrier that hearing-impaired individuals confront as a result of the majority of people's lack of knowledge of SL highlights the need for a system that facilitates interaction between these groups. In the research, a dynamic sign recognition system based on the MediaPipe framework is described to extract custom landmark coordinates. This method efficiently eliminates background noise, dress style, and skin tone from the sign recognition procedure. The model is trained on 24 commonly used ASL terms and demonstrates strong performance during evaluation, achieving 98% accuracy. The LSTM model trained on the relevant features performs successfully during the testing process.

Despite these promising results, the proposed system has certain limitations. The current system recognizes isolated dynamic signs rather than continuous, sentence-level signing, and occasional delays may occur when interpreting successive dynamic signs in real-time conditions. These limitations are related to the computational cost of real-time video processing and

the challenges of accurately detecting hand landmarks in certain orientations. These limitations, discussed in more detail in Section III-D, define clear directions for future work.

Based on the results, future research in this domain should concentrate on incorporating dynamic signs to interpret sentences. Ultimately, the proposed system has the potential for significant practical implications for real-world communication scenarios with hearing-impaired individuals. It aims to enable seamless and accurate SL recognition in a variety of environments, from educational institutions and healthcare facilities to workplaces, thereby fostering greater societal integration.

Acknowledgement

The authors received no financial support for the research.

Author Contribution

Conceived the MediaPipe-LSTM approach and designed the feature engineering pipeline, Y. F.; Recorded and collected the dataset of 3,600 dynamic sign videos, Y. F.; Developed the landmark extraction scripts and the LSTM model implementation, Y. F.; Performed model training, testing, and real-time system evaluation, Y. F.; Wrote the original manuscript draft, Y. F.; Assisted in refining the feature extraction methodology, Z. H.; Reviewed and revised the manuscript, Z. H.; Contributed to result verification and interpretation, O. L.; Contributed to State-of-the-Art discussion and revision, O. L.; Provided guidance on the overall research direction, A. A. M.; and Supervised the research and provided critical review of the manuscript, A. A. M.

Data Availability

The data that support the findings of the research are openly available in Zenodo at https://doi.org/10.5281/zenodo.21462002, reference number [40].

References

[1] I. A. Adeyanju, O. O. Bello, and M. A. Adegboye, "Machine learning methods for sign language recognition: A critical review and analysis," *Intelligent Systems with Applications*, vol. 12, pp. 1–36, 2021.

[2] S. A. E. El-Din and M. A. Abd El-Ghany, "Sign language interpreter system: An alternative system for machine learning," in *2020 2nd Novel Intelligent and Leading Emerging Sciences Conference (NILES)*. Giza, Egypt: IEEE, Oct. 24–26, 2020, pp. 332–337.

[3] Y. Farhan, A. Ait Madi, A. Ryahi, and F. Derwich, "American sign language: Detection and automatic text generation," in *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*. Meknes, Morocco: IEEE, March 3–4, 2022, pp. 1–6.

[4] Y. Farhan and A. Ait Madi, "Real-time dynamic sign recognition using MediaPipe," in *2022 IEEE 3rd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS)*. Fez, Morocco: IEEE, Dec. 1–2, 2022, pp. 1–7.

[5] B. Singh and A. K. Dubey, "Deaf education and sign language: Strategies, challenges, and benefits," *Naveen International Journal of Multidisciplinary Sciences (NIJMS)*, vol. 1, no. 3, pp. 75–82, 2025.

[6] S. Stone, "FSU expert highlights sign language's role for the deaf and hard-of-hearing community," 2025. [Online]. Available: https://bit.ly/3UbifaF

[7] N. A. A. Boi-Dsane, "Being understood: How to expand sign language access for the deaf community," 2024. [Online]. Available: https://bit.ly/3Ue3iEJ

[8] A. A. Hosain, P. S. Santhalingam, P. Pathak, H. Rangwala, and J. Kosecka, "Hand pose guided 3D pooling for word-level sign language recognition," in *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. Computer Vision Foundation, 2021, pp. 3429–3439.

[9] A. S. Al-Shamayleh, R. Ahmad, N. Jomhari, and M. A. M. Abushariah, "Automatic arabic sign language recognition: A review, taxonomy, open challenges, research roadmap and future directions," *Malaysian Journal of Computer Science*, vol. 33, no. 4, pp. 306–343, 2020.

[10] S. Srivastava, S. Singh, Pooja, and S. Prakash, "Continuous sign language recognition system using deep learning with MediaPipe holistic," *Wireless Personal Communications*, vol. 137, pp. 1455–1468, 2024.

[11] S. Alyami, H. Luqman, and M. Hammoudeh, "Isolated Arabic sign language recognition using a transformer-based model and landmark keypoints," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 23, no. 1, pp. 1–19, 2024.

[12] A. Sultan, W. Makram, M. Kayed, and A. A. Ali, "Sign language identification and recognition: A comparative study," *Open Computer Science*, vol. 12, no. 1, pp. 191–210, 2022.

[13] R. A. Kadhim and M. Khamees, "A real-time American sign language recognition system using convolutional neural network for real datasets," *TEM Journal*, vol. 9, no. 3, pp. 937–943, 2020.

[14] A. Costa, "ASLScribe: Real-time American sign language alphabet image classification using MediaPipe hands and artificial neural networks," 2019, final project in University of Georgia.

[15] J. Galka, M. Masior, M. Zaborski, and K. Barczewska, "Inertial motion sensing glove for sign language gesture acquisition and recognition," *IEEE Sensors Journal*, vol. 16, no. 16, pp. 6310–6316, 2016.

[16] P. Kumar, H. Gauba, P. P. Roy, and D. P. Dogra, "Coupled HMM-based multi-sensor data fusion for sign language recognition," *Pattern Recognition Letters*, vol. 86, pp. 1–8, 2017.

[17] F. Obaid, A. Babadi, and A. Yoosofan, "Hand gesture recognition in video sequences using deep convolutional and recurrent neural networks," *Applied Computer Systems*, vol. 25, no. 1, pp. 57–61, 2020.

[18] M. U. Rehman *et al.*, "Dynamic hand gesture recognition using 3D-CNN and LSTM networks," *Computers, Materials & Continua*, vol. 70, no. 3, pp. 4675–4690, 2022.

[19] W. Zhang, J. Wang, and F. Lan, "Dynamic hand gesture recognition based on short-term sampling neural networks," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 1, pp. 110–120, 2020.

[20] A. Kanodia, P. Singh, D. Rajesh, and G. Malathi, "Indian sign language using holistic pose detection," *Turkish Journal of Physiotherapy and Rehabilitation*, vol. 32, no. 3, pp. 19 746–19 752, 2021.

[21] B. Duy Khuat, D. Thai Phung, H. Thi Thu Pham, A. Ngoc Bui, and S. Tung Ngo, "Vietnamese sign language detection using MediaPipe," in *Proceedings of the 2021 10th International Conference on Software and Computer Applications*. Kuala Lumpur, Malaysia: Association for Computing Machinery, Feb. 23–26, 2021, pp. 162–165.

[22] A. Chaikaew, K. Somkuan, and T. Yuyen, "Thai sign language recognition: an application of deep neural network," in *2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering*. Cha-am, Thailand: IEEE, March 3–6, 2021, pp. 128–131.

[23] M. G. Grif and Y. K. Kondratenko, "Development of a software module for recognizing the fingerspelling of the Russian sign language based on LSTM," *Journal of Physics: Conference Series*, vol. 2032, no. 1, pp. 1–7, 2021.

[24] J. Shin, A. Matsuoka, M. A. M. Hasan, and A. Y. Srizon, "American sign language alphabet recognition by extracting feature from hand pose estimation," *Sensors*, vol. 21, no. 17, pp. 1–19, 2021.

[25] L. Zheng, B. Liang, and A. Jiang, "Recent advances of deep learning for sign language recognition," in *2017 International Conference on Digital Image Computing: Techniques and Applications (DICTA)*. Sydney, NSW, Australia: IEEE, Nov. 29–Dec. 01, 2017, pp. 1–7.

[26] N. B. Ibrahim, H. H. Zayed, and M. M. Selim, "Advances, challenges and opportunities in continuous sign language recognition," *Journal of Engineering and Applied Sciences*, vol. 15, no. 5, pp. 1205–1227, 2020.

[27] C. Lugaresi *et al.*, "MediaPipe: A framework for building perception pipelines," 2019. [Online]. Available: https://arxiv.org/abs/1906.08172

[28] S. Bouktif, A. Fiaz, A. Ouni, and M. A. Serhani, "Multi-sequence LSTM-RNN deep learning and metaheuristics for electric load forecasting," *Energies*, vol. 13, no. 2, pp. 1–21, 2020.

[29] A. Sherstinsky, "Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network," *Physica D: Nonlinear Phenomena*, vol. 404, 2020.

[30] GitHub, "MediaPipe holistic," 2024. [Online]. Available: https://github.com/google-ai-edge/mediapipe/blob/master/docs/solutions/holistic.md

[31] ——, "MediaPipe hands," 2024. [Online]. Available: https://github.com/google-ai-edge/mediapipe/blob/master/docs/solutions/hands.md

[32] ——, "MediaPipe pose," 2024. [Online]. Available: https://github.com/google-ai-edge/mediapipe/blob/master/docs/solutions/pose.md

[33] B. Guruprakash, N. Gurusamy, M. Ramnath, S. Sumathi, E. Mariappan, and T. Saravanan, "ISL sign language recognition using LSTM-driven deep learning model," *Journal of Electrical Systems*, vol. 20, no. 3, pp. 1–9, 2024.

[34] Ridwang, A. A. Ilham, I. Nurtanio, and Syafaruddin, "Dynamic sign language recognition using mediapipe library and modified LSTM method," *International Journal on Advanced Science, Engineering & Information Technology*, vol. 13, no. 6, pp. 2171–2180, 2023.

[35] M. Z. Uddin, C. Boletsis, and P. Rudshavn, "Real-time Norwegian sign language recognition using MediaPipe and LSTM," *Multimodal Technologies and Interaction*, vol. 9, no. 3, pp. 1–15, 2025.

[36] G. Khartheesvar, M. Kumar, A. K. Yadav, and D. Yadav, "Automatic Indian sign language recognition using MediaPipe holistic and LSTM network," *Multimedia Tools and Applications*, vol. 83, pp. 58 329–58 348, 2024.

[37] A. M. Buttar *et al.*, "Deep learning in sign language recognition: A hybrid approach for the recognition of static and dynamic signs," *Mathematics*, vol. 11, no. 17, pp. 1–20, 2023.

[38] A. Baihan, A. I. Alutaibi, M. Alshehri, and S. K. Sharma, "Sign language recognition using modified deep learning network and hybrid optimization: A Hybrid Optimizer (HO) based optimized CNNSa-LSTM approach," *Scientific Reports*, vol. 14, pp. 1–22, 2024.

[39] E. Hassan, M. Y. Shams, T. Abd El-Hafeez, and M. Elseddik, "A novel model for expanding horizons in sign language recognition," *Scientific Reports*, vol. 15, pp. 1–21, 2025.

[40] Y. Farhan and A. Ait Madi, "DynASL-24: Dynamic American Sign Language (ASL) recognition dataset," 2026. [Online]. Available: https://zenodo.org/records/21462002