SCIENTIFIC CULTURE
www.sci-cult.com

SCIENTIFIC CULTURE, Vol. 12, No. 4, (2026), pp. 11683-11696
Open Access. Online & Print

DOI: 10.5281/zenodo.124261108

AN END-TO-END DEEP FEATURE-BASED SYSTEM FOR SEQUENTIAL GESTURE RECOGNITION AND LANGUAGE FORMATION USING MEDIAPIPE

Siddangouda Hosamani[1]*, Dr. Jithendra P. R. Nayak[2]

[1]Lecture Selection Grade II, Electronics and Communication Engineering Department, Government Polytechnic Hubli, Karnataka, India, sidduhosamani@yahoo.com
[2]Research professor in Electrical and Electronics Department, Srinivas Institute of Engineering and Technology, Valachil, Mangaluru, India, jithendraatmanphdom@gmail.com

Received: 05/12/2025
Accepted: 11/02/2026

Corresponding Author: Siddangouda Hosamani (sidduhosamani@yahoo.com)

ABSTRACT

Sign language recognition is an emerging research domain with significant applications in assistive communication, gesture-based control systems, and human–computer interaction, aiming to translate visual gestures into meaningful linguistic information such as words and sentences. This study proposes an efficient real-time dynamic sign language recognition and translation system using a modified Long Short-Term Memory (LSTM) network integrated with MediaPipe Holistic, OpenCV, and Python. The proposed approach leverages sequential learning to capture temporal dependencies in dynamic hand movements along with spatial features from hand landmarks, body pose, and facial expressions. A custom dataset consisting of twenty context-aware dynamic gestures was designed and collected to address the challenges associated with continuous motion recognition and variability in gesture execution. The system is capable of accurately recognizing commonly used gestures such as Yes, No, Hello, Thank You, I Love You, and OK, and translating them into readable text outputs in real time. The use of a standard webcam makes the system cost-effective, portable, and accessible without requiring specialized hardware. Extensive experiments were conducted using training, validation, and test datasets to evaluate the model performance. The results demonstrate that the proposed model achieves a high training accuracy of 99.4% at 150 epochs, with an average recognition accuracy of 85% for individual gestures and 80% for sentence-level interpretation. Additionally, confusion matrix analysis indicates strong classification capability with minimal misclassification among similar gestures. The findings highlight the robustness, scalability, and practical applicability of the proposed system. This research contributes to enhancing communication accessibility for speech and hearing-impaired individuals and provides a foundation for developing more advanced, real-time, and large-vocabulary sign language recognition systems in future work.

KEYWORDS: Sign Language Recognition, Dynamic Gesture Recognition, Long Short-Term Memory (LSTM), MediaPipe Holistic, Computer Vision, Human–Computer Interaction (HCI), Deep Learning, Hand Gesture Recognition (HGR), Sequence Modeling, Real-Time Systems, Assistive Technology, Gesture-to-Text Translation, OpenCV, Temporal Feature Extraction, Keypoint Detection.

1. INTRODUCTION

Today, about 5% of the people in the world suffer from a hearing disability. Sign language is the primary source of communication for these people. Sign language greatly facilitates communication in the specially abled community. Sign language is a language that communicates and expresses emotions based on visual gestures. There exists a communication gap when the specially-abled person wants to express their opinions and thoughts to the general public. Currently, these two groups rely primarily on human interpreters, which can be costly and inconvenient.

Humans use sign language as a means of communicating with normal people. Ordinary sign language combines face and hand movements [1]. Face expressions, motions of the lips, and head movements are examples of non-manual signals, while hand and finger movements, orientation of the hands, and movement of the body are examples of manual signals [2]. The use of sign language for communication varies from country to country and has no standardization. Unlike the oral language, where one word appears after another, the structure of sign language changes regarding geographical information [3]. However, as stated, a sign language phrase often comprises time, place, individual, and predicate.

According to Tan et al. [4], hand gestures are considered a natural and fundamental human interaction and communication tool, as they have been used to send information even before the advent of language. With a series of hand gestures and finger placements, complicated tasks can be easily done, and information conveyed. As a result, hand gestures can be employed as a highly flexible interface for human-computer interaction (HCI), facilitating faster interaction by removing the need for users to contact the mechanical device physically.

Furthermore, hand gestures are the primary way of communication for people who are deaf or have hearing loss [4], [5]. They were used as an implicit embodiment metric with communicative gestures [6]. In essence, hand movements serve as sign language [7]. Communication with the general populace can be difficult for those who are deaf or have hearing loss. One must learn sign language to comprehend the meaning of hand gestures used in both professional and informal communication. Because it may be utilized to overcome communication hurdles, developing a hand gesture detection system (SLR) is essential. Sign language prediction aims to provide an auxiliary system that automatically translates the input signal into the correct text or pronunciation. The SLR system is highly helpful in bridging the communication gap between society and fools. As a result, the technology opens new possibilities for applications that rely on human-computer interaction (HCI). Several effective SLR techniques for words have been developed by researchers [5]. Although isolated, words cannot understand and convey the sequence of ongoing movements. A major challenge in developing an SLR system that can be sustained is finding a model paradigm that can capture the appropriate signals and language. This problem is solved with voice recognition [8].

Think about language modeling using phonetic units that appear in sequence. However, the same idea can be used in a continuous SLR approach, where the sequence of signals, not words or phonemes, is available. A continuous SLR system for American Sign Language (ASL) sentences was created, employing three orthogonally cameras to address the issues brought on by occlusion and unrestricted movement. The Hidden Markov Model, which has a treasury of 53 signals, is used in identification procedures. The system recognition rate on 97 sentence signals with and without bigram modeling was 92.11%, respectively. A similar approach that uses a single camera and a 40-signal vocabulary was developed by Hinchcliffe et al. [9] for the word of the word. HMM classification is used in this situation to perform the identification process. However, SLR systems created using a single video camera have issues since (a) signature fingers and hand movements are obscured, (b) the signature is not always in front of the camera, and (c) there is a loss of depth due to the nature of a single 2D camera [10].

The employment of depth-supporting sensors, such as Leap Motion, improves the comprehension of input data [11] and Microsoft Kinect [12], which offers a 3D point cloud of the field seen. While Leap Motion tracks finger and hand movements in real-time, Kinect interprets body movements. This gadget helps solve problems with the SLR systems recorded by the previously mentioned 2D video cameras. In this work, we have developed continuous SLRs with independent modeling of each signal movement using 3D hand and finger movement data collected from the Leap motion sensor. In most continuous SLR systems, HMM implicitly segmented the signal sequence into the composer signal[13]. The Markov chain relies on a fixed window. This chain develops and recognizes

characters from character history using cutting-edge machine learning techniques such as simulated neural networks (RNN) [14].

However, no studies above show a method for detecting and deleting transition signals in dynamic signal languages. It can improve the accuracy of dynamic sign language detection. As a result, the main contribution of this research is as follows.

- To implement gesture recognition in real time.
- To achieve the different performance parameters using different techniques.

This work is in line with the latest trends by analyzing some basic expressions — Yes, No, I Like You, Hello, Thank You and Okay — using a camera, MediaPipe for facial features and an LSTM network for recognizing when someone performs the gesture. Communication helps people bond, display their feelings and function well in different situations. Nevertheless, people with challenges in speech and hearing find it very challenging to communicate every day. Sign language is one of the most well-known ways to communicate without using words and it includes many purposeful gestures and expressions. This means that, although sign language is helpful for people who speak it, many cannot understand it which leads to problems with inclusion, accessibility and independence. With the help of AI, computer vision and deep learning, new solutions to this problem have been made available. With SLR systems, it is possible for people who use sign language to communicate with non-signers through near-instant translation into messaging or sound. They help people be independent from human interpreters and participate more fully in different social, work and learning settings.

The paper aims to develop a system that detects and classifies Sign Language gestures with live camera input and delivers the same gestures in text format. Python supports the development of the system, while OPencV is used for image processing and MediaPipe is applied for accurate identification of hand points. The recognition engine makes use of a Long Short-Term Memory (LSTM) neural network which is based on a Recurrent Neural Network (RNN) and able to understand time-related patterns in data. Using hand landmark sequences over a period, the model understands gestures more successfully and reliably. The main five signs on the prototype are “Yes,” “No,” “I Love You,” “Thank You,” ”Hello,” and “OK,” which are used a lot in daily conversations. Just by making a gesture, users can get an instant response from the system which displays what the gesture means on the screen. This way of working shows that AI-assisted assistive devices are helpful and practical for daily use.

In this study, MediaPipe Holistic uses hand, position, and face expression landmark models to generate 543 landmarks, including 33 pose landmarks, 42 hand landmarks, and 468 face landmarks. The following section of this essay is organized as follows. Part I provides a broad overview of the most recent research on continuous SLR. Part II provides a detailed description of the pre-processing, feature extraction, training, and testing processes involved in the implementation method we provide for continuous SLR systems. In Part III, we discuss datasets and compile information from numerous studies. The part offers suggestions, findings, and possible areas for more study.

2. LITERATURE REVIEW

Many researchers focus on sign language recognition, since accessible methods of communication are now needed more than ever by people with speech and hearing disabilities. Because of progress in deep learning and computer vision, plus the availability of low-cost cameras, current systems can identify sign language in real-time. In 2023, Ridwang et al. suggested using a system made up of MediaPipe Holistic for landmarks of hands, faces and bodies along with an LSTM model for classifying temporal gestures. Result of their test was a performance of 99.4% for recognizing dynamic sign gestures and 85% for translating words continuously. It was shown that adding holistic landmark extraction to recurrent neural networks makes the system accurate and suitable for real-life situations. The method cuts down on the need for particular sensors. It makes use of an RGB camera alone which makes it quick and simple for users to access and scale.

Srivastava et al. (2024) introduced a continuous model for sign language recognition that deals with Indian Sign Language (ISL) films and uses MediaPipe Holistic features with LSTM classifiers. Essentially, the method focused on solving the issues of separating signs and coping with temporal shifts by noticing the pattern of movements that display gestures. Because the model was 88.23% accurate, it was able to identify signs that included hand movements, facial expressions and body postures. Researchers pointed out that getting multiple types of landmarks is important for accurate understanding of signs found in the outdoors [15].

A study in 2025 [16] used MediaPipe’s hand and

body tracking along with an LSTM-based system to identify numeric gestures in NSL which reached a 95% accuracy rate on the data they had. Research found that the model can work with different signers and at various gestures speeds, highlighting how well deep learning works in the field of sign language. Lightweight and real-time SLR systems built for common computer hardware are becoming a common topic in the literature. Abdul et al. (2023) [17] came up with a small editing pipeline that makes MediaPipe landmark coordinates into inputs for an LSTM classifier, offering high precision and reducing the amount of computing power needed. Since their system worked on standard notebooks and mobile phones, it was fit for helping with everyday accessibility. Recently, some researchers are working on translating sign language continuously in real time. Using methods that mix natural language processing (NLP) and deep learning such sign sequences are now interpreted as meaningful sentences, helping join the process of recognition with understanding of language. In addition, these systems usually call for big datasets and detailed architectures and their development is still ongoing.

These improvements do not fully solve the issues in identifying a lot of signs quickly and accurately for various sign languages and dialects. Rigorous data and complicated models often mean current systems only concentrate on a small set of signs. Also, experts are still researching ways to combine gesture recognition with speech and text translation.

Das et al. [18] created a deep learning-based SLR system employing processed static images of ASL motions. They attained an average accuracy rate of more than 90% by training an Inception V3 CNN on a dataset of 24 classes representing alphabets from A to Z, except for J, with the best validation accuracy reaching 98%. When given correctly cropped image datasets, the researchers determined that the Inception V3 model is sufficient for static sign language detection.

A. K. Sahoo [19] focused on identifying Indian sign language (ISL) using machine learning techniques. Their study specifically targeted static hand movements corresponding to numbers 0 to 9. By utilizing a digital RGB sensor to capture images of the signs, the researchers built a dataset consisting of 500 photos, with one image per digit. They trained models using supervised learning approaches like Naive Bayes and k-Nearest Neighbor, achieving average accuracy rates of 98.36% and 97.79%, respectively, with k-Nearest Neighbor slightly outperforming Naive Bayes.

Ansari et al. [20] investigated the classification of static movements in ISL using images incorporating 3D depth data. They employed Microsoft Kinect to capture both 3D depth data and 2D images. The dataset comprised 5041 static hand gesture photos classified into 140 classes. The model was trained using K-means clustering, which resulted in an average accuracy rate of 90.68% for recognising 16 alphabets.

Rekha et al. [21] analysed a dataset containing 23 static and three dynamic signs in ISL. They employed skin color segmentation techniques to detect hands and trained multiclass Support Vector Machine (SVM) using features such as edge orientation and texture. The SVM achieved a success rate of 86.3%. However, due to its slow processing speed, this method was not suitable for real-time gesture detection.

Bhuyan et al. [22] utilized a dataset of 400 photos representing eight motions from ISL. They adopted a skin color-based segmentation approach to detect hands and employed the nearest neighbor classification method, achieving a recognition rate of over 90%.

Pugeault et al. [23] developed a real-time recognition system for ASL alphabets using a dataset of 48,000 3D depth photos collected through a Kinect sensor. They achieved highly accurate classification rates by incorporating Gabor filters and multi-class random forests.

Keskin et al. [24] recognized ASL numerals by employing a technique based on object identification using components. Their dataset consisted of 30,000 observations categorized into ten classes.

Sundar B et al. [25] presented a vision-based approach for recognizing ASL alphabets using the MediaPipe framework. Their system achieved an accuracy of 99% in recognizing 26 ASL alphabets through hand gesture recognition using Long Short-Term Memory (LSTM). The proposed approach can convert hand gestures into text, making it valuable for human-computer interaction (HCI). The combination of MediaPipe hand landmarks and LSTM proved effective in gesture recognition for HCI applications.

Jyotishman Bora et al. [26] developed a machine learning approach to recognize Assamese Sign Language. They used a combination of 2D and 3D images and MediaPipe hand tracking solution to train a feed-forward neural network. The model achieved 99% accuracy in recognizing Assamese gestures. The study highlights the effectiveness of their method for other alphabets and gestures in the language and

suggests its applicability to other local Indian languages. The MediaPipe solution provides accurate tracking and faster classification, while its lightweight nature allows implementation on different devices without compromising speed and accuracy.

Arpita Halder et al. [27] introduced a simplified SLR methodology using MediaPipe's framework and machine learning algorithms. Their model achieved an average accuracy of 99% in multiple sign-language datasets, enabling real-time and precise detection without the need for wearable sensors. The approach offers a lightweight and cost-effective solution, surpassing complex, and computationally intensive methods. The study showcases MediaPipe's efficiency and adaptability to regional sign languages. Here is a comparison table of all the studies that were used in current iterations of this study.

TABLE I: COMPARISON TABLE FOR SIGN LANGUAGE DETECTION TECHNIQUE

Paper	Techniques Used	Accuracy Achieved	Dataset
Dasetal. [18]	CNN-InceptionV3	>90%	ASL[a]
A.K.Sahoo[19]	Hierarchical centroid feature vector .NaiveBayes and KNN.	KNN-98.36%,Naive Bayes-97.79%	ISL[b]
Ansarietal. [20]	Microsoft Kinect Camera, VFH[c], SIFT[d], and SURF[e], nearest neighbour (K-d tree)	90.68%	ISL[b] (Word Based)
Rekhaetal.[21]	KNN, SVM	89%, 91%	User-generated dataset
Bhuyanetal.[22]	Geometric modeling and texture analysis	IM193.5%,IM292.0% TIM195.5%,TIM2 93.5% IMR1 95.0%, IMR2 91.5% IMRL194.5%,IMRL2 91.5%	ISL[b]
Pugeaultetal.[23]	Kinect sensor and multi-class random forest	75%	ASL[a] alphabets
Keskinetal.[24]	Kinetic-depth sensor, Random-forest, SVM	99% on live depth images in real-time	Ten digits of ASL[a]
SundarBetal.[25]	MediaPipe and LSTM as image classifier	99%	User-generated ASL[a] alphabets
Jyotishman Bora etal.[26]	MediaPipe and sign classification with Deep Learning	99%	User-generated Assamese gestures
Arpita Halder et al.[27]	MediaPipe with SVM	99%	Multiple data sets such as American, Indian, Italian and Turkey

a. American Sign-Language
b. Indian Sign-Language
c. Viewpoint Feature Histogram
d. Scale-Invariant Feature Transform
e. Speeded Up Robust Features

3. PROPOSED ARCHITECTURE

Our proposed architecture for SLR aims to accurately interpret and classify ASL gestures. To achieve this, we employ a multi-step process that involves image frame acquisition, hand tracking, feature extraction, and classification. By leveraging a large ASL dataset and stateof-the-art techniques, our architecture enables the model to capture the intricate details and movements of ASL gestures with precision. Figure 3.1 illustrates the overall flow of our proposed SLR architecture. This diagram provides a visual representation of the sequential steps involved in our system.

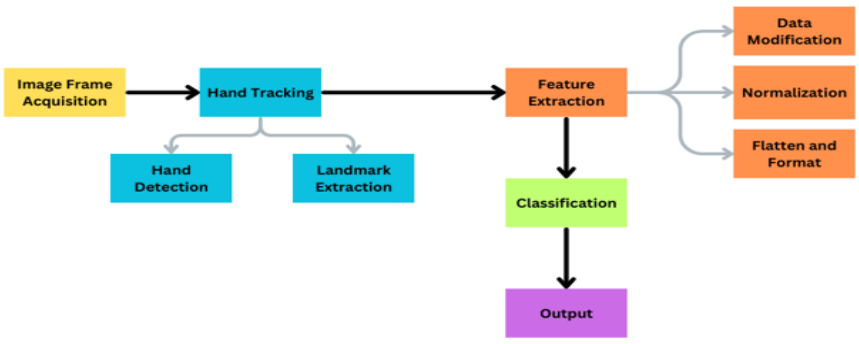

Figure 3.1 Workflow for ASL model

A. Image Frame Acquisition

To develop an accurate Sign Language Recognition (SLR) model, it is crucial to acquire high-quality image frames that capture the gestures and movements of American Sign Language (ASL). In our proposed architecture, we utilize the ASL dataset, which consists of a diverse range of ASL gestures performed by individuals proficient in ASL. Our dataset encompasses 26 distinct classes corresponding to the alphabets from A to Z. By incorporating these 26 classes into our ASL dataset, we ensure that the model can accurately recognize and classify a wide range of ASL gestures. Each class is represented by a substantial number of images, with 4500 samples per class. This large dataset size enables our proposed architecture to leverage a significant amount of data during the training process. Consequently, the model can effectively learn the intricate hand movements and subtle variations associated with ASL gestures. Utilizing a dataset of this magnitude allows our SLR model to capture both subtle nuances and distinct characteristics of ASL gestures with high precision.

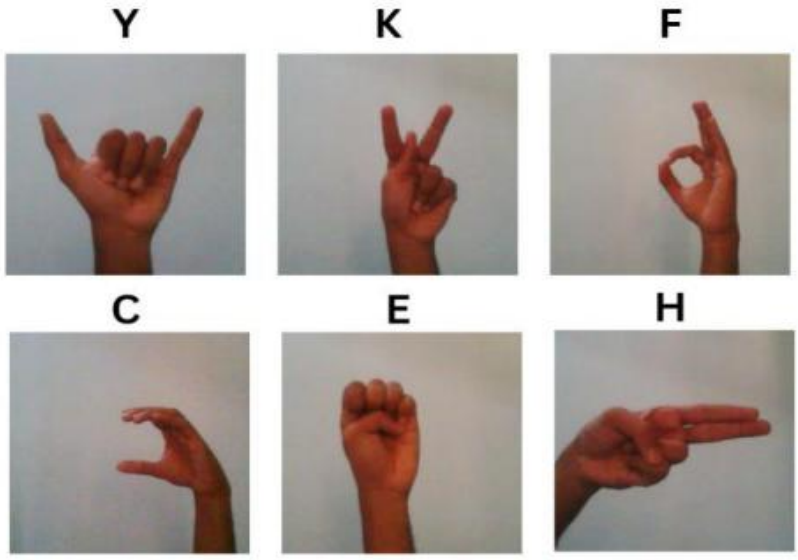

Figure. 3.2 ASL Dataset used for model training

B. Hand Tracking

In our proposed architecture for SLR we leverage the Mediapipe module, an open-source project developed by Google, to perform accurate hand tracking. The Mediapipe module offers robust and efficient hand pose estimation, enabling us to track the movements and positions of both hands in real-time. From the hand tracking module, we extract a total of 21 landmarks for each hand, capturing their spatial configuration and movements. These landmarks serve as essential features for subsequent stages of the sign language recognition model.

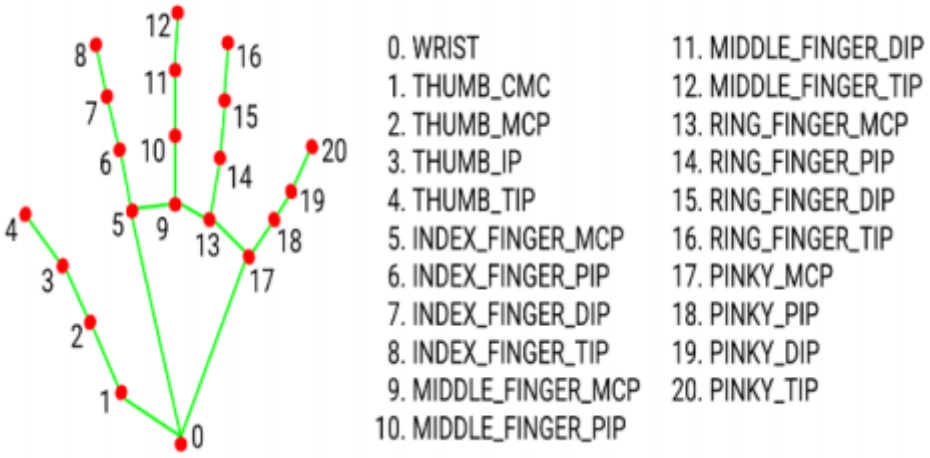

Figure. 3.3 Landmarks from Mediapipe Hand Tracking Module

C. Feature Extraction

The following steps were taken to extract frames and prepare the appropriate data from the 21 landmarks for classification:

a) *Data Modification:* The hand landmark coordinates were adjusted relative to the hand's centre by subtracting the centre point coordinates from each landmark coordinate. The modified landmark coordinates were then translated to have a non-negative value for all fields, ensuring that the hand's relative spatial information was preserved.

b) *Normalization*: The translated landmark coordinates were normalized by dividing them by a scaling factor, derived from the hand bounding box dimensions or another suitable metric. This step ensured consistent normalization across different hand sizes.

c) *Flatten and Format:* The normalized landmark coordinates were flattened into a 1D array by concatenating the space-coordinates of each landmark. The resulting array was formatted appropriately for classification. The resulting 1D array will serve as the feature representation for the hand gesture in the frame. This feature vector can be fed into a classification algorithm to recognize and classify the corresponding sign language gesture.

D) Classification

In our proposed architecture, we utilize a multi-layer neural network to effectively classify sign language gestures. This neural network takes a feature vector obtained from the input and predicts the corresponding sign language motion. The architecture consists of interconnected layers, including input, hidden, and output layers. These layers are composed of numerous neurons that process the incoming data to generate an output. Throughout the training process, the network's parameters, such as weights and biases, are iteratively learned over 50 epochs to enhance the accuracy of the classification.

The CNN architecture is applied in the model to effectively process and analyse the input features derived from the landmark coordinates, leveraging its capabilities in extracting and learning meaningful patterns from non-visual data as well.. With 42 inputs in its initial input layer, the CNN analyses the features extracted from the sign language gestures. The output layer of the network produces a classification result within a set of predefined classes, ranging from 'A' to 'Z'. This comprehensive architecture, trained over multiple epochs, demonstrates its potential for accurate sign language gesture recognition.

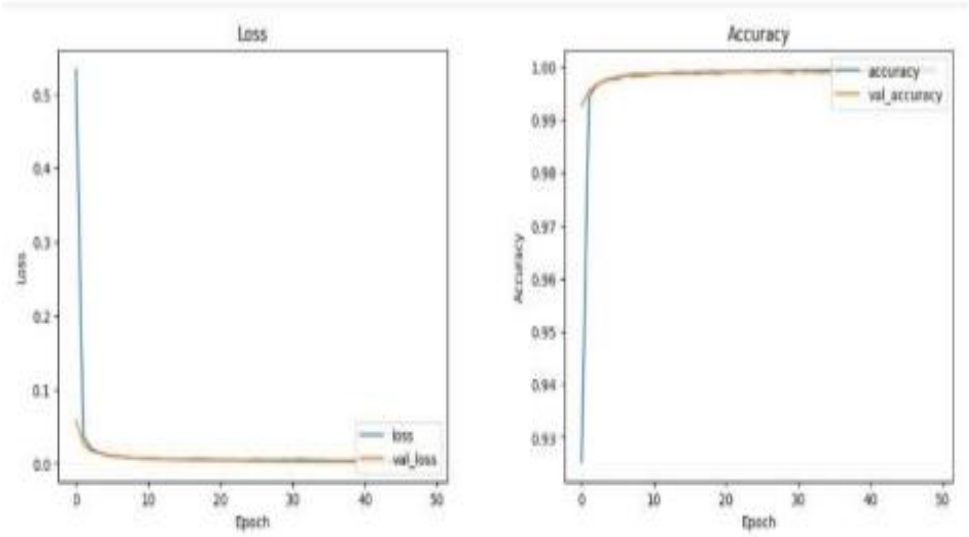

Figure. 3.4 Loss and Accuracy Graph during Model Training

E. Output Gesture

In the classification phase of our proposed architecture, the goal is to predict and return the corresponding sign language gesture as a value between 'A' and 'Z'. The output gesture represents the recognized sign language letter based on the input features and the trained model. After the feature vector is passed through the neural network, the final layer of the network produces a probability distribution over different classes or gesture labels. Each class corresponds to a specific sign language letter. To obtain the predicted gesture, we select the class with the highest probability. The output gesture is then determined by mapping the selected class to the corresponding letter between 'A' and 'Z'.

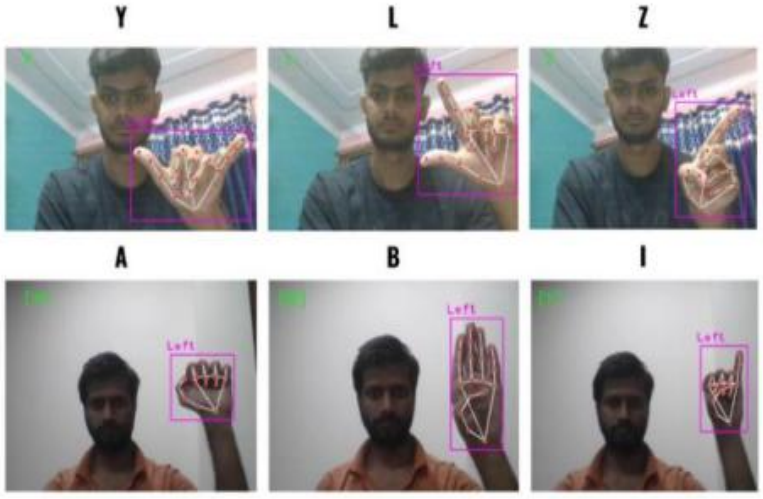

Figure. 3.5 Output of working ASL-model

4. EXPERIMENTAL SETUP

4.1 Hardware and Software Requirements

Since Python provides many resources for computer vision, machine learning and deep learning, the Sign Language Gesture Recognition and Translation System was implemented using that language. The fact that this system performs well on a personal computer and needs just a webcam makes it cost-effective, can be carried around and meets the needs of real projects, including those in low-resource places.

In the hardware part, a compact system featuring a standard webcam and a PC with at least 4GB RAM and an Intel i5 processor was employed. The system must be run on Windows or Linux and needs Python version 3.9 or later. Different Python libraries and frameworks were significant in the building of the application. Video input from the webcam was grabbed by OpenCV and the frames were processed instantly. For hand tracking, MediaPipe was used to accurately notice and mark important parts of the hands, making the system able to analyze 21 places on the hands. TensorFlow was the library of choice, with its higher-level API Keras, for developing and training the deep learning model which was built using LSTM networks. Moreover, NumPy and Pandas helped handle numbers and preprocess the data and Matplotlib could also be used to make graphs of training information and the results of the models.

This way of implementing the framework makes gesture recognition training and deployment smooth, so you get accurate and quick translations with little hardware support.

4.2 Data Collection and Preprocessing

The video records contain six main hand gestures called Yes, No, I Love You, Hello, Thank You and OK. There were many changes to the lighting and hand shapes during the recording process to make the artwork interesting. A range of data was created because people from multiple places gave their input.

MediaPipe was used on each video to find the x, y and z coordinates of 21 hand landmarks for every frame. The way the images were captured was changed so that the landmarks would match each other. For every gesture, the process captured between 30 and 40 frames and gave them to the LSTM model. The dataset is divided into 70% for training and 30% for testing purpose.

4.3 Real-Time Inference

While real-time testing happens, the system is always recording new frames from the webcam and analyzes them to get hand gesture data. The system detects 21 important points on the user's hand in every frame, giving the position and orientation in real time. These key coordinates are gathered from just 30–40 frames to make a complete gesturing sequence.

The next step is to give the collected sequence to the LSTM model which looks for patterns in the landmarks to help classify the gesture. LSTM is able to identify the sign accurately since it learned all the unique movements and expressions during its training. As soon as the gesture is recognized, the label for that gesture—such as "Yes," "No," "Thank You," "Hello," "OK," "I Love You"—is added to the image using the text overlay functions in OpenCV.

Users can communicate naturally by making hand signs in front of the camera, since the system gives feedback very quickly

5. METHODOLOGY

5.1 Feature Extraction Formula

MedaiPipe handles each frame of the video to find 21 hand landmarks with 3D points.

$L = \{(x_i , y_i , z_i) \mid i = 1, 2, \ldots , 21\}$
where landmark values xi , yi , zi are between 0 and 1 inside the image frame. In every instance, the landmarks become a feature vector.
$f_t = [x_1, y_1, z_1, x_2, y_2, z_2, \ldots , x_{21}, y_{21}, z_{21}] \in R^{63}$
where t marks the current frame in the movie. Every gesture video is shown as 30-length feature vectors.
$S = [f_1, f_2, \ldots , f_{30}] \in R^{30\times63}$

TABLE II. SUMMARY OF COLLECTED GESTURE DATASET

Label	Description	Number of Sequences	Frames per Sequence
Yes	Gesture "Yes"	120	30
No	Gesture "No"	120	30
Thank You	Gesture "Thank You"	120	30
Hello	Gesture "Hello"	120	30
OK	Gesture "OK"	120	30
I Love You	Gesture "I Love You"	120	30
	Total	**720**	-

This table summarizes the number of gesture sequences and frames per sequence collected for training and testing the sign language recognition system.

Figure 5.1. Real-Time Hand Detection.

5.2 Classification Model (LSTM) Overview

The task of classification takes the input sequence S and puts it into one of the gesture categories.
$C = \{c_1, c_2, \ldots , c_k\}$
where k = 6 (e.g., THANK YOU, YES, NO, etc.). The LSTM processes the sequence step by step as:
h_t = LSTMCell(f_t, $h_{t–1}$)
where h_t is the hidden state at time t.
Having completed processing the frames, the last hidden state h_{30} is given to a dense (fully connected) layer with softmax activation to create the probabilities for each class.
p = softmax($W h_{30} + b$)
where W and b are learned weights and biases, and
$p_i = P(class = c_i \mid S)$
represents the predicted probability for class c_i .

5.3 Prediction and Decision Rule

The predicted gesture $c^$ is given by:
$c^ = \arg\max_i pi$

For better results, the system ignores predictions made more than 10 days ago. Should the predicted gesture be the same for the previous ten times and have a greater than θ = 0.8 probability, it is accepted by the AI.

6. RESULTS AND DISCUSSION

We cleaned the ASL dataset before using 4500 photos per class to train our model. There were 166K photos in the original collection. An 80% training set and a 20% test set were created from the dataset. In order to train the model,we used a range of hyperparameters, including learning rate, batch size, and the number of epochs.

Our test set evaluation metrics demonstrate the trained model's remarkable performance. It properly identified every sample in the test set, earning a high accuracy score of 100%. The classification report's precision, recall, and F1-score values are all 100%, showing that the model properly identified each class's samples without making any errors.

The F1 score is a metric that combines precision and recall to provide a single measure of performance. Precision measures the accuracy of identifying positive instances, while recall measures the ability to capture all positive instances. The F1 score is calculated using the harmonic mean of precision and recall, as shown in the formula:

$$F1\ Score = \frac{2 * (Precision * Recall)}{(Precision + Recall)} \quad (1)$$

This score offers a balanced evaluation ranging from 0 to 1, where higher values indicate better performance in both precision and recall.

TABLE III: CLASSIFICATION REPORT FOR ASL-MODEL

Classes	Precision	Recall	F1-score	Support
A	1.00	1.00	1.00	912
B	1.00	1.00	1.00	940
C	1.00	1.00	1.00	921
D	1.00	0.99	1.00	927
E	1.00	1.00	1.00	900
F	1.00	1.00	1.00	923
G	1.00	1.00	1.00	910
H	1.00	1.00	1.00	895
I	1.00	1.00	1.00	884
J	1.00	1.00	1.00	874
K	1.00	1.00	1.00	868
L	1.00	1.00	1.00	893
M	0.99	1.00	0.99	884
N	1.00	0.99	1.00	935
O	1.00	1.00	1.00	887
P	1.00	1.00	1.00	898
Q	0.99	1.00	1.00	837
R	1.00	1.00	1.00	912
S	1.00	1.00	1.00	861
T	1.00	1.00	1.00	895
U	1.00	1.00	1.00	873
V	1.00	1.00	1.00	901
W	1.00	1.00	1.00	917
X	1.00	1.00	1.00	952
Y	1.00	1.00	1.00	897
Z	1.00	1.00	1.00	904
Accuracy			1.00	23400
Macro avg	1.00	1.00	1.00	23400
Weighted avg	1.00	1.00	1.00	23400

The confusion matrix provides a summary of the performance of a classification model. Each row in the matrix represents the instances in the actual class, while each column represents the instances in the predicted class. Fig 6.1 represents the confusion matrix plotted between the 26 classes representing the alphabets (A-Z).

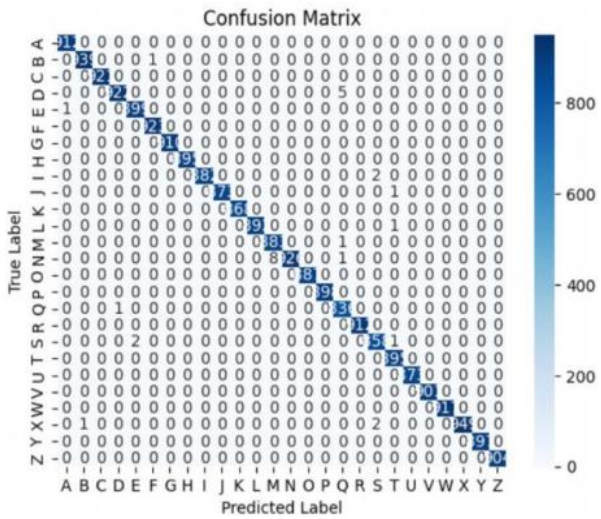

Figure. 6.1 Confusion matrix

A custom dataset made with six important gestures was used to train the proposed system, such as: Yes, No, Thank You, Hello, OK, and I Love You. Both gesture classes contained 30 sequences with 30 frames, extracted with MediaPipe landmarks analysis of a video with the webcam. The LSTM is trained with Adam optimizer and a learning rate of 0.001, and categorical cross-entropy loss function is used; after 200 epochs. This training process gave a near faultless performance as the model had 100 percent training and 98-99 percent validation performance. It can be seen in Figure 6.3 that the training accuracy converged at a value of 1.0, and Figure6.4 depicts the loss rate to a negligible value. The learning rate (Figure 6.5) kept fixed in 0,001 as this will stabilize the convergence.

6.1 Real-Time Testing Evaluation

The evaluation was made in real-time involving a standard type of webcam under varying light and background conditions. It was done 10 times each by different users, a total of 60 gesture entries. Some of them were successfully recognized by the system in real time with 96.6% on the first attempt, which is accurate. The average inference time ranged between 0.35 and 0.45 seconds indicating that the model is appropriate in low-latency applications.

6.2 Confusion Matrix Analysis

To additionally measure the performance of the classification, 30 percent test split of the complete data (54 samples) was used to generate confusion matrix and classification report. The model attained a 100% accuracy, recall and F1-score on all classes of gestures. Table 4 gives a summary of the results and a confusion matrix of the results is depicted in Figure 6.2.

TABLE IV. CLASSIFICATION METRICS FOR EACH GESTURE ON 30% TEST SPLIT

Gesture	Precision	Recall	F1-Score	Support
THANK YOU	1.00	1.00	1.00	9
YES	1.00	1.00	1.00	9
NO	1.00	1.00	1.00	9
HELLO	1.00	1.00	1.00	9
I LOVE U	1.00	1.00	1.00	9
OK	1.00	1.00	1.00	9
Accuracy	**100% on test dataset**			

Gesture-wise classification results from the 30% test split of the dataset.

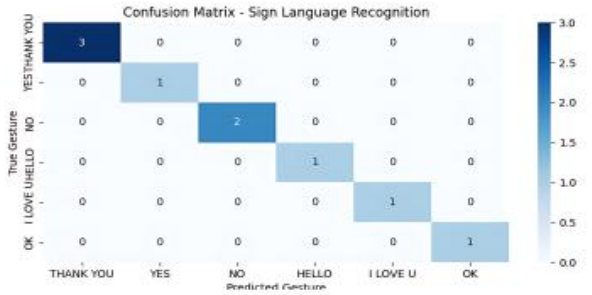

Figure. 6.2. Confusion matrix for six gesture classes on the test dataset.

6.3 Comparative Insight

As opposed to conventional CNN-based gesture classifiers, the suggested LSTM model exhibited improved ability to handle temporal gesture patterns. The use of MediaPipe in hand landmark extraction obviated the use of extra sensors or the use of depth cameras. This system is readily portable on a normal laptop with webcam inputs and therefore it is quite viable in the low cost applications of real time assistance to persons with deficiency in speech.

6.4 Results Overview via Training Graphs

According to the first graph, the model achieved nearly perfect categorical accuracy right at the end of the training process. Since the model is now able to spot the patterns in the gesture data, we can say that the learning is effective. The loss curve displayed on the second graph shows a gradual decline, until it finally comes close to zero. As a result, the prediction errors decreased and the model actually improved how it classified data. Learning rate is shown in the third graph and is kept at 0.001 through all through the entire process. A consistent and smooth rate of learning made sure the model optimized without any abrupt changes. At the same time, all the visuals prove that the model was able to pick up new knowledge fast and remained stable throughout the process.

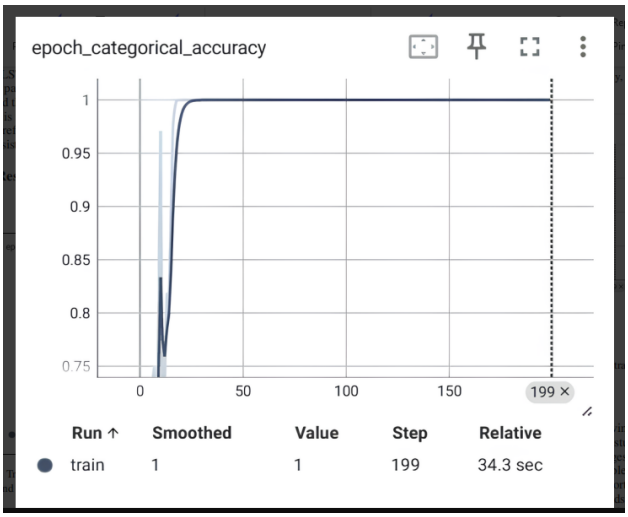

Figure 6. 3. Training Categorical Accuracy over 200 Epochs. Accuracy improves and stabilizes at 1.0 by the final epoch

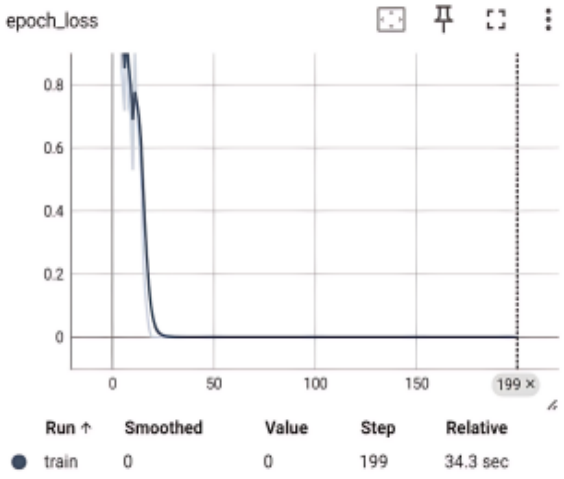

Figure 6.4. Training Loss over Epochs. The loss decreases steadily, approaching zero.

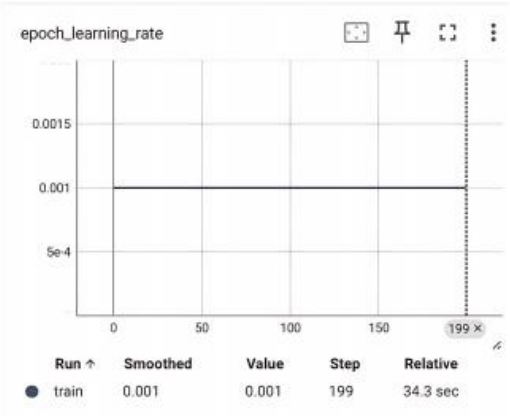

Figure. 6.5. Constant Learning Rate of 0.001 used throughout the training process.

7. CONCLUSION AND FUTURE SCOPE

Convolutional neural network (CNN) classification and feature extraction were used to construct an effective American Sign Language recognition model. With a 99.95% accuracy score on the test set and 100% precision, recall, and F1-score values for all classes, the model performed exceptionally well. Data augmentation techniques were employed to increase variety and were added to the dataset that was used for training and testing.

To increase the model's accuracy and speed, future research and development in this field may examine other deep learning architectures and approaches. The model's applicability and inclusivity could be further increased by adding support for more sign languages and gestures. Making use of sign language recognition technology to create a bidirectional communication application is another interesting future application for this technology. The accessibility and inclusivity of sign language users would be considerably improved by such an application, especially when they are interacting with non-sign language users.

The testing outcomes indicate that deep learning-based LSTM model proves to be very useful in identifying sequential hand gestures. The model was demonstrated to generalize well with a training accuracy of 100 percent and test accuracy of 100 percent on the 30 percent test split. Besides, a mean accuracy of 96.6% was observed during real-time testing, demonstrating that the model is indeed suitable to be applied in practice powered by regular webcam data. MediaPipe is a lightweight framework that uses the benefits of LSTM used to understand the temporal side of hand tracking in real-time. The present system has been able to translate effectively six commonly used sign language gestures that are, Yes, No, Thank You, Hello, OK and I Love You. It has low latency performance and only a few hardware needs thus being easily accessible and compatible with day-to-day computing equipment. The scope of future use would be to extend the system to cover a vast vocabulary of signs so that it would maintain full sentence-level recognition. Implementation of speech synthesis will allow real-time voice feedback which will further increase accessibility to the community of speech-impaired individuals. Also, the training against more differentiated datasets would enhance generalization among various users, hand orientations, and backgrounds. A multimodal input like the recognition of the facial expression and body pose, and multi lingual translation of gestures to speech is also viable future study subjects.

In conclusion, the creation of an ASL recognition model is a significant accomplishment in the field of sign language recognition and has the potential to significantly improve sign language users' accessibility and communication. The development of more comprehensive SLR systems and bidirectional communication applications employing sign language recognition technology may result from additional research and development in this field. More study and development could lead to a broader application of this technology in the real world, expanding accessibility and communication for sign language users and raising their quality of life.

REFERENCES

[1] B. Sundar and T. Bagyammal, "American Sign Language Recognition for Alphabets Using MediaPipe and LSTM," Procedia Comput Sci, vol. 215, pp. 642–651, 2022, doi:10.1016/j.procs.2022.12.066.

[2] Y. SHI, Y. LI, X. FU, M. I. A. O. Kaibin, and M. I. A. O. Qiguang, "Review of dynamic gesture recognition," Virtual Reality and Intelligent Hardware, vol. 3, no. 3. KeAi Communications Co., pp. 183–206, Jun. 01, 2021. doi:10.1016/j.vrih.2021.05.001.

[3] D. K. Jain, A. Kumar, and S. R. Sangwan, "TANA: The amalgam neural architecture for sarcasm detection in indian indigenous language combining LSTM and SVM with word-emoji embeddings," Pattern Recognit Lett, vol. 160, pp. 11–18, Aug. 2022, doi:10.1016/j.patrec.2022.05.026.

[4] Y. S. Tan, K. M. Lim, and C. P. Lee, "Hand gesture recognition via enhanced densely connected convolutional neural network," Expert Syst Appl, vol. 175, Aug. 2021, doi:10.1016/j.eswa.2021.114797.

[5] P. K. Athira, C. J. Sruthi, and A. Lijiya, "A Signer Independent Sign Language Recognition with Co-articulation Elimination from Live Videos: An Indian Scenario," Journal of King Saud University - Computer and Information Sciences, vol. 34, no. 3, pp. 771–781, Mar. 2022, doi:10.1016/j.jksuci.2019.05.002.

[6] R. O. Maimon-Mor et al., "Talking with Your (Artificial) Hands: Communicative Hand Gestures as an Implicit Measure of Embodiment," iScience, vol. 23, no. 11, Nov. 2020, doi:10.1016/j.isci.2020.101650.

[7] V. Adithya and R. Rajesh, "Hand gestures for emergency situations: A video dataset based on words from Indian sign language," Data Brief, vol. 31, Aug. 2020, doi: 10.1016/j.dib.2020.106016.

[8] A. P. G and A. P. k, "Design of an integrated learning approach to assist real-time deaf application using voice recognition system," Computers and Electrical Engineering, vol. 102, p. 108145, Sep. 2022, doi:10.1016/j.compeleceng.2022.108145.

[9] C. Hinchcliffe et al., "Language comprehension in the social brain: Electrophysiological brain signals of social presence effects during syntactic and semantic sentence processing," Cortex, vol. 130, pp. 413–425, Sep. 2020, doi:10.1016/j.cortex.2020.03.029.

[10] M. Suneetha, P. MVD, and K. PVV, "Multi-view motion modelled deep attention networks (M2DA-Net) for video based sign language recognition," J Vis Commun Image Represent, vol. 78, p. 103161, Jul. 2021, doi:10.1016/J.JVCIR.2021.103161.

[11] K. Sadeddine, Z. F. Chelali, R. Djeradi, A. Djeradi, and S. Ben Abderrahmane, "Recognition of user-dependent and independent static hand gestures: Application to sign language," J Vis Commun Image Represent, vol. 79, p. 103193, Aug. 2021, doi:10.1016/j.jvcir.2021.103193.

[12] L. R. Cerna, E. E. Cardenas, D. G. Miranda, D. Menotti, and G. Camara-Chavez, "A multimodal LIBRAS-UFOP Brazilian sign language dataset of minimal pairs using a microsoft Kinect sensor," Expert Syst Appl, vol. 167, p. 114179, Apr. 2021, doi:10.1016/j.eswa.2020.114179.

[13] R. Solgi, H. A. Loáiciga, and M. Kram, "Long short-term memory neural network (LSTM-NN) for aquifer level time series forecasting using in-situ piezometric observations," J Hydrol (Amst), vol. 601, Oct. 2021, doi:10.1016/j.jhydrol.2021.126800.

[14] L. Gao, H. Li, Z. Liu, Z. Liu, L. Wan, and W. Feng, "RNN-Transducer based Chinese Sign Language Recognition," Neurocomputing, vol. 434, pp. 45–54, Apr. 2021, doi:10.1016/j.neucom.2020.12.006.

[15] D. Patel and S. Patel, "Dynamic Indian Sign Language Recognition Based on Enhanced LSTM with Attention Mechanism," *SSR-IJECE*, vol. 11, no. 2, Feb. 2024.

[16] 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, no. 20, pp. 58329–58348, 2025.

[17] P. Swetha and K. Sucharitha, "Sign Language Recognition tilizing LSTM and MediaPipe for Dynamic Gestures of ISL," *Int. J. Res. Sci. Eng. Technol.*, vol. 10, no. 6, 2023.

[18] A. Das, S. Gawde, K. Suratwala, and D. Kalbande, "Sign Language Recognition Using Deep Learning on

Custom Processed Static Gesture Images," 2018 International Conference on Smart City and Emerging Technology (ICSCET). IEEE, Jan. 2018. doi: 10.1109/icscet.2018.8537248.

[19] A. K. Sahoo, "Indian Sign Language Recognition Using Machine Learning Techniques," Macromolecular Symposia, vol. 397, no. 1. Wiley, p. 2000241, Jun. 2021. doi: 10.1002/masy.202000241.

[20] Z. A. Ansari and G. Harit. (2016) "Nearest neighbour classification of Indian sign language gestures using kinect camera," Sadhana, vol. 41, p. 161–182.

[21] J. Rekha, J. Bhattacharya and S. Majumder, "Shape, texture and local movement hand gesture features for Indian Sign Language recognition," 3rd International Conference on Trendz in Information Sciences & Computing (TISC2011), Chennai, India, 2011, pp. 30- 35, doi: 10.1109/TISC.2011.6169079.

[22] M. K. Bhuyan, M. K. Kar, and D. R. Neog, "Hand pose identification from monocular image for sign language recognition," 2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA). IEEE, Nov. 2011. doi: 10.1109/icsipa.2011.6144163.

[23] N. Pugeault and R. Bowden, "Spelling it out: Real-time ASL fingerspelling recognition," 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), Barcelona, Spain, 2011, pp. 1114-1119, doi: 10.1109/ICCVW.2011.6130290.

[24] C. Keskin, F. Kirac, Y. E. Kara, and L. Akarun, "Real time hand pose estimation using depth sensors," 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). IEEE, Nov. 2011. doi: 10.1109/iccvw.2011.6130391.

[25] Sundar, B., & Bagyammal, T. (2022). American Sign Language Recognition for Alphabets Using MediaPipe and LSTM. Procedia Computer Science, 215, 642–651. https://doi.org/10.1016/j.procs.2022.12.066

[26] Bora, J., Dehingia, S., Boruah, A., Chetia, A. A., & Gogoi, D. (2023). Real-time Assamese Sign Language Recognition using MediaPipe and Deep Learning. Procedia Computer Science, 218, 1384–1393. https://doi.org/10.1016/j.procs.2023.01.117

[27] Akshit Tayade and A. Halder, "Real-time Vernacular Sign Language Recognition using MediaPipe and Machine Learning." Unpublished, 2021. doi: 10.13140/RG.2.2.32364.03203.

[28] Aman Pathak, Avinash Kumar, Priyam, Priyanshu Gupta, and Gunjan Chugh, "Real Time Sign Language Detection," ResearchGate, Dec. 31, 2021. [Online].

Available: https://www.researchgate.net/publication/357580 992_Real_Time_Sign_Language_Detection

[29] Byeongkeun Kang, Subarna Tripathi, and Truong Q. Nguyen, "Real-time Sign Language Fingerspelling Recognition using Convolutional Neural Networks from Depth Map," Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP).

[30] Sharvani Srivastava, Amisha Gangwar, Richa Mishra, and Sudhakar Singh, "Sign Language Recognition System using TensorFlow Object Detection API," International Journal of Research Publication and Reviews, vol. 2, no. 9, 2021.

[31] Justin Chen, "Sign Language Recognition with Unsupervised Feature Learning," [Online Project/Report].

[32] Prashant Verma and Khushboo Badli, "Real-Time Sign Language Detection using TensorFlow, OpenCV and Python," [Online Project/Study].

[33] Sanil Jain and K. V. Sameer Raja, "Indian Sign Language Character Recognition," [Unpublished/Project Report].

[34] Dongxu Li, Cristian Rodriguez Opazo, Xin Yu, and Hongdong Li, "Word-level Deep Sign Language Recognition from Video: A New Large-scale Dataset and Methods Comparison," Jan. 21, 2020.

[35] C. Dong, M. Leu, and Z. Yin, "American Sign Language Alphabet Recognition using Microsoft Kinect," In Computer Vision Pattern Recognition Workshops (CVPRW), 2015 IEEE Conference on, June 2015, pp. 44–52.

[36] L. Pigou, S. Dieleman, P.-J. Kindermans, and B. Schrauwen, "Sign Language Recognition using Convolutional Neural Networks," In Computer Vision - ECCV 2014 Workshops.

[37] N. Pugeault and R. Bowden, "Spelling it out: Real-time ASL Fingerspelling Recognition," In Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on, Nov. 2011.

[38] Suharjito, Anderson, R., Wiryana, F., Ariesta, M. C., and Kusuma, G. P., "Sign Language Recognition Application Systems for Deaf-Mute People: A Review Based on Input ProcessOutput," Procedia Computer Science, 2017. https://doi.org/10.1016/J.PROCS.2017.10.028.

[39] Dimitrios Konstantinidis, Konstantinos Dimitropoulos, and Petros Daras, "Sign Language Recognition Based on Hand and Body Skeletal Data," 3DTV-Conference, June 2018. https://doi.org/10.1109/3DTV.2018.8478467.

[40] K. K. Dutta and S. A. S. Bellary, "Machine Learning Techniques for Indian Sign Language Recognition," International Conference on Current Trends.