EAI Endorsed Transactions
on AI and Robotics

Research Article EAI.EU

A Comprehensive Approach to Indian Sign Language Recognition: Leveraging LSTM and MediaPipe Holistic for Dynamic and Static Hand Gesture Recognition

Prachi Rawat[1], Papendra Kumar[2], Vivek Kumar Tamta[3], Anuj Kumar[4, *]

[1,2,3]Computer Science And Engineering, G. B. Pant Institute of Engineering and Technology, Pauri Garhwal, India
[4]Department of Computer Science, School of Technology, Doon University, Dehradun, Uttarakhand, India

Abstract

Recognizing Indian Sign Language (ISL) gestures effectively is crucial for improving communication accessibility for the deaf community. This study introduces an innovative approach that integrates a Sequential Long Short-Term Memory (LSTM) model with MediaPipe Holistic for accurate and real-time gesture recognition. This work outlines a straightforward approach to recognizing Indian Sign Language (ISL) gestures effectively. The process is divided into three steps: Extracting features from data, cleaning, labeling, and identifying gestures using MediaPipe Holistic. The system tracks landmarks on the face, hands, and body across video frames, capturing essential details such as temporal and spatial features for interpreting gestures. Firstly, data cleaning and labeling are done by eliminating unclear, fuzzy images and null entries. Then, the processed data is passed into a Sequential LSTM model, which has two LSTM layers and a dense output layer. The proposed approach improves the model's performance by integrating techniques such as early stopping and categorical cross-entropy. The model is trained and tested using a customized ISL dataset that included 11 distinct gestures, and it achieved a high accuracy rate of 96.97%. The framework emphasizes the model's robustness across diverse lighting conditions and real-world scenarios, ensuring its applicability in sectors such as healthcare, education, and public service. By enhancing communication for ISL users, it effectively addresses existing gaps and improves accessibility in these domains.

Keywords: Sign Language Recognition, LSTM, Indian Sign Language, MediaPipe Holistic, Computer Vision, Deep Learning, Gesture Recognition

Received on 12 February 2025, accepted on 30 April 2025, published on 19 May 2025

doi: 10.4108/airo.8693

*Corresponding author. Email: dranujdhiman@gmail.com

1. Introduction

Sign language recognition using AI techniques is an emerging field that aims to make communication more accessible for people who can't hear. Sign language plays a vital role as a primary mode of communication for many deaf individuals, and detecting gestures in sign language accurately and in real-time can drastically improve how well they are able to interact with others. By incorporating body movements, gestures using hands, and expressions on the face, sign language effectively communicates meaning through visual means. Traditional methods for interpreting sign language rely on human interpreters, which can be expensive, time-consuming, and not always available. Roughly 430 million people worldwide need rehabilitation therapies to correct their hearing loss, which is more than 5% of the world population [1]. By 2050, the number of individuals suffering from debilitating hearing loss is projected to surpass 700 million or one in every ten of the world's population [1]. Hearing loss causes deaf and hard-of-hearing persons a variety of challenges in their everyday life. The communication gap is one main difficulty as these people may find it difficult to grasp speech, especially in loud background noise settings or when the speaker is not immediately facing them. This might complicate interaction with others, especially in professional and social contexts. Other common challenges include - Employment discrimination: Deaf and hard-of-hearing individuals may

face discrimination in the workplace, particularly in jobs that require good hearing skills, resulting in limited career opportunities, Educational challenges: assessing education is hard particularly if they attend schools that do not provide specialized resources or accommodations, Access to information: these specially-abled individuals may struggle to access information conveyed through auditory means, such as television, radio, and phone calls and, Social isolation: Communication barriers can lead to social isolation and a sense of disconnection from the larger community. Deaf-mute individuals face significant obstacles in receiving adequate healthcare due to a lack of social interaction and communication [2].

Despite popular belief, deaf people who communicate through sign language may not possess the same level of reading and comprehension skills as those who communicate verbally, primarily as a result of grammatical and structural variations in sentences [3]. However, it is a reality that the people for whom sign language is their main means to communicate typically learn their native language as a secondary language, leading to potential challenges in reading and writing fluency. This can be exemplified by the translation of the sentence in English "I don't understand" into sign language, where it is represented as "I understand no". The primary use of sign language can make it challenging for individuals to read and write in other languages, as other words may be seen as noise [4]. The upward trend in deaf students enrolling in colleges and universities over the course of the last two decades has led to a communication challenge, affecting both teacher-student interactions and peer-to-peer communication [5], [6]. Even though some countries have taken steps to address the needs of deaf students by investing in and adapting their institutions and staff [7], there is still considerable work to be done to ensure that deaf students have a fluid learning experience comparable to that of non-deaf students. The complexity of sign language poses a challenge for hearing individuals to comprehend, underscoring the vital role of human interpreters during emergency scenarios. However, this interpreting service has its limitations, as there are only around 300 certified interpreters in India [8], making it difficult to provide interpretation services in educational settings, training courses, and urgent circumstances. In recent years, online interpretation services have become popular, but they are heavily reliant on the strength of internet connections on both the end of the signer and the human interpreter. Furthermore, the availability of human interpreters also affects the effectiveness of this service. Therefore, an automatic hand sign language recognition system that does not rely on special equipment is necessary to ensure equal communication opportunities between the deaf and hearing communities.

Although conventional computer vision-based algorithms have shortcomings in accuracy and robustness, action recognition systems have demonstrated promise in spotting complicated and dynamic movements. Action recognition is important and beneficial for sign language recognition as it helps the hearing-impaired population to communicate more easily. The main goal of the proposed method is to apply a vision-based tool to recognize words using dynamic and static gesture in ISL (Indian sign language). The present study looks at how to use Long Short-Term Memory (LSTM) networks with the Mediapipe Holistic pipeline to detect and comprehend sign language gestures in real time. In legal environments, education, and healthcare, where precise and real-time interpretation of sign language is vital, this may be quite useful. By increasing communication accessibility and offering greater access to information and services, sign language recognition employing action recognition can overall improve the quality of life for the hearing-impaired population.

The contributions of the proposed work are as follows:

***Dataset Creation*:** A novel dataset is developed, encompassing Indian Sign Language gestures, both dynamic and static, meticulously annotated with various classifications.

Pre-processing: The data undergoes pre-processing, organized into sequences and labels, and divided into training and testing sets. The proposed sequential LSTM model is subsequently trained using the Adam optimizer and categorical cross-entropy loss, incorporating early stopping.

***Model Design and Training*:** The proposed LSTM-based model captures temporal dependencies in hand movements through two LSTM layers and fully connected layers utilizing 'SELU' activation. It is trained with the Adam optimizer and categorical cross-entropy loss, employing early stopping.

***Integration of MediaPipe Holistic*:** Incorporate MediaPipe Holistic to identify and monitor hand landmarks, which function as inputs for precise gesture detection.

***Experimental Evaluation*:** The performance of the LSTM-based model is assessed using categorical accuracy. Training is done on a subset of the dataset, and evaluation is performed on unexplored data.

Results and Analysis: In-depth detailed analysis is done considering strengths, limitations, and potential applications of Indian Sign Language recognition.

2. Background and Related work

Traditional sign language recognition (SLR) systems have been based on image processing techniques, such as template matching or edge detection. These methods are often computationally expensive and are not always effective at capturing the nuances of sign language gestures. In recent years, machine learning algorithms for improving the efficiency and accuracy of SLR systems have been created. The success of transformer-based models in modeling sequential data underscores their potential for capturing temporal dependencies, a principle we adapt for temporal gesture modeling in LSTM networks to address the dynamic nature of sign language. One such approach is the implementation of LSTM networks. When processing sequential data, like sign language gestures, LSTM neural networks are especially effective. They have the capacity to capture long-term relationships between inputs and outcomes, making them particularly effective at recognizing

complex patterns in sign language. Mediapipe Holistic is a pipeline developed by Google that uses machine learning to perform full-body pose estimation, hand tracking, and face detection in real time. By combining the capabilities of LSTMs and Mediapipe Holistic, it is feasible to create a real-time SLR system that is capable of accurate detection and interpreting gestures of sign language. Automating sign language recognition involves keeping track of body parts like the face and hands, extracting pertinent features, and applying computer vision and machine learning to recognize movement patterns matching certain signs. For training and validation, the system's accuracy is dependent on a sizable collection of annotated datasets, in our case its sign language videos. The paper presents an economical and real-time approach for recognizing Indian sign language that is independent of the signer. Whether they include one hand or both hands, the system successfully recognizes static and dynamic gestures.

Two artificial intelligence methods that have shown great promise in precisely identifying sign language gestures in real time are deep learning and computer vision. Using these techniques, machine learning algorithms are trained on large datasets made up of sign language videos, therefore enabling them to identify the patterns and differences defining sign language gestures. Artificial intelligence-driven sign language recognition holds significant promise for use in many sectors, including education, legal environments, and healthcare. For example, AI-based SLR systems can provide real-time captioning for online lectures or meetings, so allowing deaf and hard-of-hearing people to fully engage in these events. In healthcare environments, systems of artificial intelligence-based sign language recognition can facilitate communication between doctors and patients who are deaf or hard of hearing.

A variety of technologies, including vision-based and glove-based approaches, are used in gesture recognition systems to record gestures made with the hands. By employing sensors embedded in the glove, hand movements are detected and the associated data is transmitted to a computer, achieving high accuracy in gesture recognition in the data glove-based method [9]. However, this approach is expensive and inconvenient. The Deora and Bajaj [10] suggest a glove-based instrumentation strategy that recognizes ISL alphabets and numbers but only focuses on static gestures. For seamless hand segmentation, the implementation of this technique necessitates the wearer to put on a glove that is colored blue and red, and the recognition process attains a recognition rate of 94% utilizing PCA. Signs, where both hands overlap, cannot be recognized by this method.

The use of vision-based techniques offers enhanced user convenience, and these methods can be mainly segmented into two primary categories: 3D hand model-based approaches and appearance-based approaches. The 3D hand model-based technique [11], [12] employs 3D data of body parts to extract crucial parameters such as joint angles and palm position. Although this method requires a considerable amount of storage space to handle multiple features, it offers improved accuracy and faster computational speed. Several techniques advocate using depth cameras to capture data, as proposed by Suarez and Murphy [13] and Kapuscinski et al. [14]. Popular depth cameras for capturing images include Kinect, ASUS Xtion, and others, with Kinect being the most widely used [14], [15]. The depth-based SLR system presented by Kim et al. [16] employed SVM as the classifier and utilized hand direction, length of the finger, and radius of palm as features. These features are used to construct a decision tree for recognizing hand gestures. Because of the constraints of glove-based methods, many research works have shifted their focus toward appearance-based approaches such as Discrete Wavelet Transform and Hidden Markov Model [17], Artificial Neural Networks [18], [19], Fingertip-based Gesture Recognition [20], [21], Scale Invariant Feature Transform [22], etc. Real-time performance and faster processing time are key advantages of the appearance-based approach, attributable to the incorporation of 2-D image features.

Islam and Akhter [23] targeted ASL alphabet recognition by introducing a novel technique that demonstrates promising results. PCA-based features, an orientation-based hash code, and a Gabor filter are employed in this approach to represent different ASL alphabets. The classification of the extracted features is performed by the artificial neural network (ANN). They used their own dataset of 24 static gestures to test the performance in this article. Aly et al. [24] present a technique for ASL fingerspelling recognition which employs a depth sensor in the research they conducted. Effective feature extraction and learning from depth images are achieved through the adoption of the Principle Component Analysis network (PCANet). The classification task is accomplished using a linear support vector machine (SVM) of the 24 static ASL gestures. Tao et al. [25] proposed method utilizes a convolutional neural network (CNN) with inference fusion and multiview augmentation for SLR. The acquisition of depth images for the gestures involved the use of a Microsoft Kinect camera. It was recommended use augmented data as a training strategy for the CNN model. Although this technique achieves commendable recognition accuracy, it demands substantial computational resources.

Numerous researchers in the literature have also embraced contact-based approaches for gesture recognition. Kim and Chong [26] introduced a method for identifying ASL with a wearable device. In this study, twenty-eight ASL words were obtained using 6 inertial measuring units (IMU), and classification was done using the LSTM algorithm. In order to facilitate ISL translation, Abraham et al. [27] designed a real-time system for hand gesture recognition utilizing sensors. The data from the sensor has been analyzed in this research to extract hand orientation and finger movements, which are subsequently transmitted wirelessly to a processing device. The classification task is accomplished using an LSTM network. Performance of the model is evaluated using a dataset made up of 26 often used ISL gestures. Gupta and Kumar [28] proposed a new sensor-based approach to ISL identification. Signers had IMUs and electromyograms attached to both their forearms to collect information on the signs. Using a multi-label classification

approach that considers the lexical attributes of signs, this system achieved an error rate of just 2.73%.

Kaur et al. [29] especially examined the influence of orthogonal moment-based local features to understand their significance in ISL categorization. The study found that these features were user-independent in terms of rotation, scaling, and translation and offered good accuracy for the recognition of the ISL dataset. Kumar and Kumar [30] in their work generate the ISL database of 26 alphabets by using one sample from every one of the 12 different signers. The recognition of signs was carried out using a traditional machine learning technique, which involved utilizing HOG features and training the model using the extreme learning machine method. Xiao et al. [31] has an RNN-based gesture recognition technique for effective Chinese sign language translation. Using the signer's skeleton sequence, the approach enabled communication bi-directionally. Standard RGB-depth images containing a range of static gestures were used to conduct the performance assessment of this approach. Lianyu Hu et al. [32] presented CorrNet, a model intended for continuous sign language recognition (CSLR) which captures movements of the body over a series of frames. CorrNet improves spatial-temporal reasoning by emphasizing hand and facial motions, resulting in superior accuracy on extensive datasets like PHOENIX14 and CSL. Zhao et al. [33] proposed a Motion-Aware Masked Autoencoder with Semantic Alignment (MASA) to enhance sign language detection. MASA employs self-supervised learning to extract dynamic motion cues and global semantic information.

Transformer-based designs have shown much more promise in SLR lately. For example, Hu et al. [34] suggested a transformer-based model for continuous sign language recognition that produced state-of-the-art outcomes on large-scale datasets such as PHOENIX14 and CSL. Saproo and Aggarwal [35] suggested a transformer-based method for Indian Sign Language (ISL) recognition, hence attaining 92% accuracy. Using pre-trained models—VGG16, EfficientNet, and MobileNet—they classified the latent embeddings extracted from frame-by-frame video processing using a Transformer network. Recent advancements in deep learning highlight the transformative potential of sequential models for accessibility; building on this foundation, our work aims to democratize communication for the deaf community through scalable, real-time sign language recognition using LSTMs and MediaPipe Holistic.

Additionally, self-supervised learning has drawn interest because to its ability to enhance model performance by using vast quantities of unlabeled data. Zhao et al. [33] proposed a self-supervised learning strategy for sign language recognition that employs a motion-aware masked autoencoder with semantic alignment. This technique emphasizes the potential of self-supervised learning to improve the resilience and accuracy of SLR models. Sandoval-Castaneda et al. [36] explored various self-supervised transformer methods for isolated sign language recognition on the WLASL2000 dataset, finding MaskFeat to achieve superior performance with a top-1 accuracy of 79.02%.

Building on these advances, we provide a Sequential LSTM model for real-time gesture recognition that works with MediaPipe Holistic. Our method employs LSTM networks to capture temporal dependencies in sign language gestures, in contrast to transformer-based models. Our model also uses MediaPipe Holistic for precise feature extraction, guaranteeing strong performance in a variety of real-world settings and lighting conditions. Our suggested model is successful in recognising Indian Sign Language (ISL) gestures, as evidenced by its high accuracy rate of 96.97%. Table 1 shows the comparison of various techniques for gesture recognition.

Table 1. A brief comparison of the various recognition methods for hand gestures

Author and Year	Data acquisition method	Dataset utilized	Classifiers and Features
Aly et al. [24] (2019)	Kinect Sensor	24 static gestures of ASL	Principle component analysis network (PCANet) and SVM
Tao et al. [25] (2018)	Kinect Sensor	24 static gestures of ASL	CNN
Chong & Kim [26] (2020)	Contact based	28 static gestures of ASL	Long short-term memory (LSTM)
Abraham [27] (2019)	Contact based	26 gestures of ISL	LSTM
Gupta & Kumar [28] (2020)	Contact based	100 isolated signs of ISL	Multi-label classification (MLC)
Kaur et al. [29] (2017)	Vision-based, using RGB pictures	26 signs of ISL Jochen-Triesch's dataset of ASL	Dual-Hahn and Krawtchouk moments using four distinct classifiers
Kumar & Kumar [30] (2021)	Vision-based, using RGB pictures	26 signs of ISL	(HOG), Extreme learning machine
Xiao [31] (2020)	Vision-based, using RGB pictures	Chinese Sign Language	Recurrent Neural Network (RNN)
Lianyu Hu et al. [32] (2023)	Vision-based, using RGB pictures	German Sign Language- 6841 sentences, vocabulary of 1295 signs; German Sign Language- 8247 sentences, vocabulary of 1085 signs; Chinese Sign Language- 20654 sentences, 10 signers; Chinese Sign Language- 25000 videos, vocabulary of 178 signs, 100	ResNet18 (2D CNN), 1D CNN, BiLSTM

		sentences;	
Zhao et al. [33] (2024)	Vision-based, using RGB pictures	American Sign Language – 2000 words, 100 signers; American Sign Language – vocabulary size of 1000; Chinese Sign Language- vocabulary size of 1067; Chinese Sign Language- 500 words, 50 signers	Transformer encoder, Graph Convolutional Network (GCN), motion-aware masked autoencoder (MA) and semantic alignment (SA) module
Saproo and Aggarwal [35] (2024)	Vision-based	INCLUDE	Transformer-based
Sandoval-Castaneda et al.[36] (2023)	Vision-based	WLASL2000	Transformer based approaches to self-supervised learning

3. Materials and Methods

The proposed approach for accurately recognizing sign motions and converting them into readable text involves following key steps: Data pre-processing and Feature extraction, Cleaning and Labeling, and Gesture recognition. The MediaPipe framework is used for data pre-processing and feature extraction. From input frames sequence collected by a web camera, the MediaPipe framework employs built-in data augmentation techniques to extract key points and landmarks as features from the face, body, and hands. After the initial stage of extracting key points, the second stage focuses on saving these key points in a file. Subsequently, null entries are identified and removed from the data, and the process proceeds with data labeling. The translated sign gestures are displayed as text on the screen in the third step, where our proposed Sequential LSTM model is utilized to train and classify the cleaned and labeled gestures for ISL recognition.

3.1. Standard LSTM

For the majority of computer vision problems, it is important to account for temporal dependencies between inputs and effectively model short-term as well as long-term sequences. Such sequential data may be effectively managed and processed by recurrent neural networks (RNNs). Unlike conventional neural networks, RNNs target on manipulating learning contextual relationships in and between sequential data using state neurons. Due to a number of restrictions and the vanishing and exploding gradient challenges, training RNNs is a challenging process. The Long Short-Term Memory (LSTM) [37] is popular in the deep learning field thanks to its efficient modeling and sequential data processing. LSTM [37] addressed the problem of vanishing gradients which is frequently seen in traditional RNNs. LSTM are one of the most popular neural nets that capture and maintain information over prolonged sequences compared to simple RNNs, due to its memory cells. The flow of information into, out of and inside the memory cell of LSTM is controlled by using gating mechanisms including input gate, forget gate, and output gate. The LSTM, owing to its selective manage knows how to keep and retrieve records successfully, is especially optimal for sequences with long-time period dependencies [38]. In addition, LSTM is crucial to training very deep neural networks (many layers) because of being able to deal with the vanishing gradient problem. By preserving gradients across extensive sequences, LSTM demonstrates its ability to proficiently propagate error signals and capture dependencies that extend over numerous time steps. This exceptional feature has cemented LSTM's stance as a sought-after choice for various applications, such as NLP, time series analysis, and speech recognition [39]. Within the LSTM architecture, memory blocks play a crucial role in retaining the prior network states' memory and facilitating the hidden states to adaptively update by selectively incorporating or disregarding past information. The input gate (i), output gate (o), and forget gate (f) are the three multiplicative components that make up these blocks, together with a memory cell that is connected to itself. The following update occurs in the LSTM when it receives an input:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (1)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \quad (2)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \quad (3)$$

$$g_t = \tanh(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (4)$$

$$c_t = f_t \Theta c_{t-1} + i_t \Theta g_t \quad (5)$$

$$h_t = o_t \Theta \tanh(c_t) \quad (6)$$

where W stands for the matrix representing the connection weights between two units, and the outputs of the input gate, forget gate, cell, output gate, and block at time t are respectively represented by i_t, f_t, c_t, o_t, and h_t respectively. Θ denotes element-wise multiplication.

$$\delta(x) = \frac{1}{1+e^{-x}} \quad (7)$$

$$\tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \quad (8)$$

In Eq. (7) real-valued inputs are condensed to the [0, 1] range through sigmoid nonlinearity. Eq. (8) represents the hyperbolic tangent activation function (tanh).

3.2. MediaPipe

MediaPipe developed by Google, is an open-source framework designed to facilitate the development of applications related to perceptual computing. With its hybrid

platform, MediaPipe constructs pipelines specifically designed for the processing of perceptual data, encompassing images, videos, and audio. It provides pre-built machine learning models and processing modules for tasks like pose estimation, face detection, and hand tracking. Real-time hand tracking and gesture detection are accomplished by this extensive approach utilizing ML. It ensures precise detection of sign gestures, resulting in advanced finger and hand tracking solutions. Our approach involved the utilization of the MediaPipe Holistic pipeline, which enables successful landmarks extraction from hands, body pose, and face. A representation of dataset collection is shown in Figure 1.

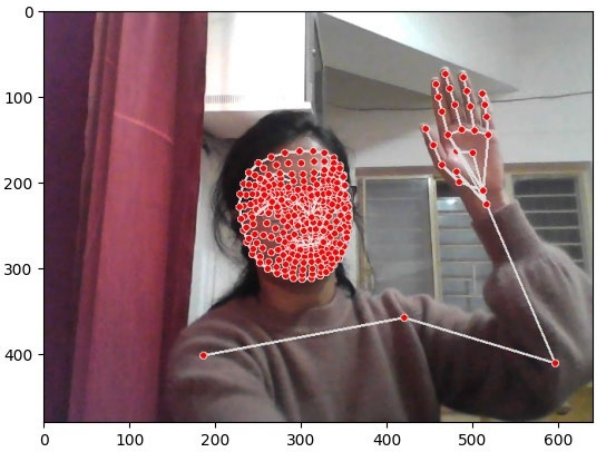

Figure 1. Representation of image dataset collection process

MediaPipe holistic pose landmarks. Leveraging the capabilities of the BlazePose detector, the MediaPipe Holistic framework extracts around 33 3D landmarks that encompass x, y, and z coordinates, enabling accurate body pose estimation from the provided image or video as input. Moreover, the framework efficiently identifies the regions of interest (ROI) associated with the detected pose, facilitating targeted analysis and robust localization of key pose regions. The input to the framework involves ROI-cropped frames, which enable the successive detection of poses through the utilization of pose landmarks and division masks within the defined region of interest. It is therefore well-suited for SLR applications since it identifies and locates a larger number of key points accurately.

MediaPipe holistic hand landmarks. To ensure accurate estimation within a single frame of approximately 21 3D hand landmarks, which involve x, y, and z coordinates, this framework incorporates two models: the hand keypoint localization model and the palm detection model. The Blaze Palm single-shot detector is initially utilized to efficiently detect palms and fists, focusing on the essential rigid parts instead of irrelevant objects in the input image. The palm detection output is then utilized for hand keypoint localization, resulting in three outputs: 21 knuckle points on the hand in two or three dimensions, the likelihood of hand presence in the input image is determined using a hand flag, and a left and right-hand binary classification.

MediaPipe holistic face landmarks. It presents a cutting-edge approach for real-time face geometry estimation, enabling the calculation of 468 3D face landmarks by leveraging just a single input camera, negating the reliance on additional depth sensors. Two deep neural network models are used in this advanced system: a detector for identifying face locations throughout the entirety of the image and based on the identified locations, a 3D face landmark model is utilized to predict the surface geometry. Coordinate prediction accuracy can be given priority in the network by accurate face cropping and reduction of data augmentation operations like rotation, scaling, and translation.

The proposed model and the general framework of proposed SLR system is shown in Figure 2 and Figure 3.

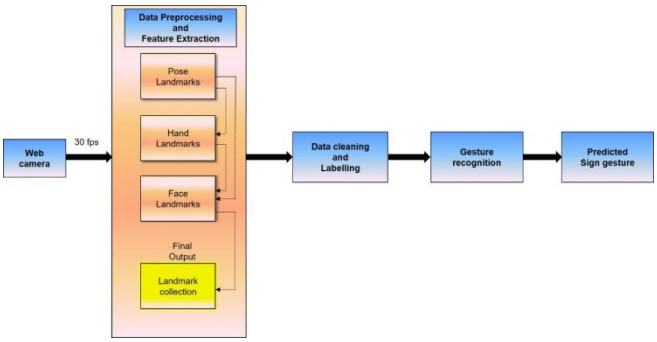

Figure 2. Block representation of proposed sequential LSTM model

Stage 1: For our data preprocessing and feature extraction approach, we employed the MediaPipe Holistic framework as a multistage pipeline. The hands, face, and pose components are managed by individual models within this framework. It employs an image resolution specific to each component's region to guarantee optimal performance. MediaPipe Holistic uses integrated data augmentation techniques to extract features from keypoints and landmarks in the image or video frames, which are then saved for further processing.

We utilized BlazePose's pose detector to estimate the human pose and landmark model. The entire list of 540+ landmarks was generated after the integration of all the landmarks.

Stage 2: Once stage 1 is completed, the per-frame landmarks (21 x 3 + 21 x 3 + 33 x 4 + 468 x 3 = 1662) extracted as features are flattened, concatenated, and stored in a file for identification as well as the elimination of any null elements within the data. In order to prevent null entries in the dataset resulting from failed feature detection, data cleaning is a crucial step. When working with blurry pictures, this is especially crucial because it might result in bias and decreased prediction accuracy during training.

The acquired data is processed for the subsequent stage, which involves the process of assigning labels to each class and their respective frame sequences are stored, facilitating training, testing, and validation.

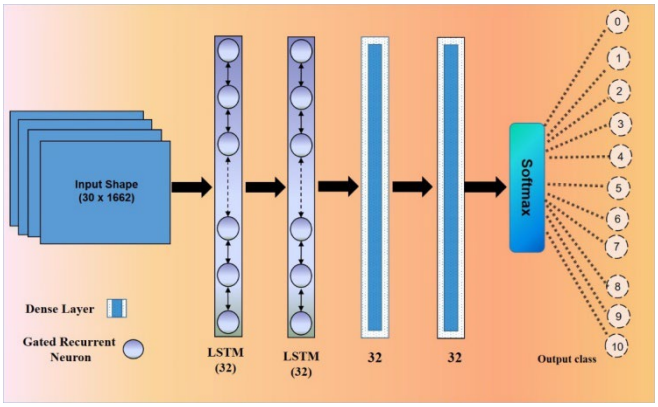

Figure 3. Gesture recognition system of proposed sequential LSTM model

Stage 3: After the completion of the cleaning and labeling process in stage 2, the data is transferred to stage 3. A 32-unit LSTM layer forms the initial layer of the model, accepting input sequences of 30 timesteps and 1662 features. SELU serves as the activation function for the LSTM layer. By setting the return sequences parameter to True, the component returns sequences instead of the last output, and the shape of the input data is specified by the input shape parameter. The second LSTM layer, which has also 32 units, returns only the final output of the sequence as the return sequence parameter is set to False. One dense layer with the SELU activation function is appended to the model. This dense layer has 32 units.

The last dense layer implements a softmax activation function and contains units equal to the total of the classes (actions) involved in the classification task. With a learning rate of 0.001, the Adam optimizer [40] is used to build the model. The utilization of the categorical cross-entropy loss function allows for handling the multiple classes in the classification problem. During training, the outcomes of the model is assessed using accuracy as the metric, and callbacks are implemented to perform certain actions at specific points.

To modify the optimizer's learning rate, a learning rate scheduler is implemented, while early stopping is used to halt training if the accuracy of the model on the validation set does not show improvement after a certain predetermined number of epochs, with the goal to optimize the process of training. The use of Tensor Board notifications facilitates the documentation of the training process and enables the display of results in Tensor Board.

3.3. Hyperparameter Tuning

To enhance model performance, we carefully adjusted key hyperparameters through systematic testing and manual refinements. The final selection of hyperparameters, including LSTM units, learning rate, batch size, optimizer, activation function, and dropout rate, epochs is summarized in Table 2. We tried different amounts of LSTM units (16, 32, and 128) and discovered that 32 was the best choice for balancing the ability to understand time-related patterns and not using too much computing power. The learning rate was tested at 0.0001, 0.0005, and 0.001, with 0.0001 yielding the most stable training, preventing both slow convergence and instability. We explored batch sizes of 16, 32, and 64, ultimately selecting 32 due to its stable learning process and memory efficiency. We chose the Adam optimizer after comparing it with Stochastic Gradient Descent (SGD) and Root Mean Square Propagation (RMSprop), as it facilitated faster and more consistent convergence.

Table 2. Final Hyperparameter Selection

Hyperparameter	Final choice	Justification
LSTM Units	32	Balanced accuracy and efficiency
Learning Rate	0.0001	Ensured smooth convergence and stability
Batch Size	32	Stable learning and memory efficiency
Optimizer	Adam	Faster and more stable convergence
Activation Function	SELU (LSTM, Dense), Softmax (Output)	Improved stability and classification
Dropout Rate	None	Early stopping effectively prevented overfitting
Epochs	2000 (Early Stopping: Patience 10)	Prevented overtraining and reduced computation

We preferred SELU over ReLU for activation functions because of its ability to self-normalize, which ensures smoother training. While dropout values of 0.2, 0.3, and 0.5 were tested to mitigate overfitting, dropouts were ultimately excluded since early stopping effectively prevented overtraining. The early stopping mechanism tracked categorical accuracy and halted training after 10 consecutive epochs without any improvement. Although training was initially set for up to 2000 epochs, the model typically converged much earlier due to early stopping. These optimizations collectively improved the model's ability to generalize while maintaining efficient training and computational feasibility.

4. Results and Discussion

4.1. Dataset Description.

Total 60 videos for each sign gesture were assembled to create a real-time dataset employing a web camera, where each video consists of 30 frames and maintains a consistent

size of 640×480 pixels. A variety of the lighting environment and camera angles were used during the recording process. The dataset encompasses 11 distinct sign gestures, which are comprehensively detailed in Table 3, and Figure 4 and Figure 5. Collected data is partitioned in the ratio of 90:10 to create corresponding training and testing datasets. Hence, 180 images out of the total 1800 images acquired for each sign gesture, were allocated to the training set using the train-test split technique. Moreover, the shuffle parameter, set to True, ensures that the data undergoes random shuffling before the split. This eliminates any systematic order from the resulting training and testing sets, ensuring that there are no biases in subsequent analysis. The class distribution of the original dataset is preserved in both the training and testing sets due to the *Stratify* parameter, which permits stratified sampling. The splitting process is stratified based on the labels by passing the labels variable as the argument to stratify, maintaining representative proportions of various classes in the final subsets. These parameter selections improve the model's generalizability and dependability by guaranteeing that the training and testing sets include a varied representation of the underlying data, which is essential for the precise assessment and validation of the model's performance.

Table 3. A compilation of 11 sign movements, each accompanied by its respective label in Indian Sign Language

Labels	Sign Gestures
0	Accident
1	Allergy
2	Doctor
3	hello
4	help
5	Love
6	Money
7	Pain
8	Police
9	Thank you
10	wait

To ensure the model does not rely on specific persons for recognition, our dataset was designed to be signer-independent. This was also validated during testing since the model's generalizability was evidenced by its accurate classification of motions from unfamiliar signers.

4.2. Experimental settings.

The simulation was performed on a system Intel Core i5 processor with a clock speed of 2.50 GHz and 8 GB of RAM. The operating system was 64-bit Windows 10 Pro, and the simulation was conducted utilizing Python version 3.7. A web camera capable of capturing RGB images with a resolution of 720 pixels/30 fps was employed to capture the input image for the simulation. The model was constructed using the Keras Sequential API. It consisted of a sequence of layers designed to process sequential data.

The model architecture consisted of four layers, including two LSTM (Long Short-Term Memory) layers and two dense layers. The initial LSTM layer was set up to return sequences and had 32 hidden units. It utilized the Scaled Exponential Linear Unit (SELU) activation function and accepted input sequences of landmark key points (1662) extracted from video frames. Every video contained a sequence of 30 frames, resulting in an input shape of (30, 1662). The second LSTM layer, also with 32 hidden units, returned a single output sequence. This layer also utilized the SELU activation function.

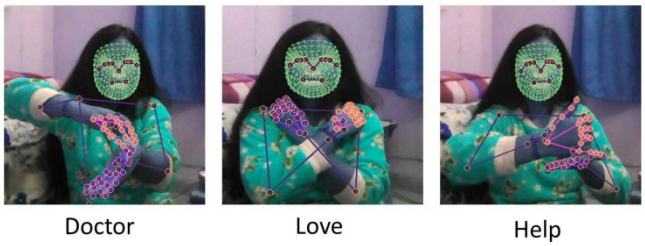

Figure 4. ISL dataset featuring Static motions

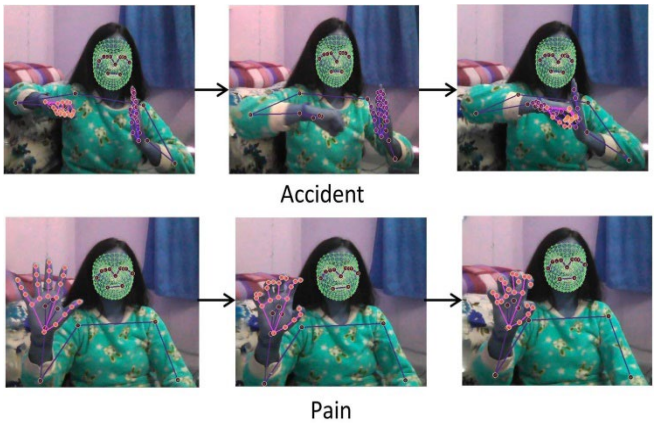

Figure 5. Dataset of dynamic gestures in ISL

A dense layer was added after the LSTM layers, comprising 32 units and utilizing SELU activation. With the number of units matching the number of actions in the dataset, a dense layer employing the softmax activation function was utilized as the output layer. With a learning rate of 10^{-4} for the Adam optimizer, models were trained. The model's loss function of choice was categorical cross-entropy, and its performance was measured using categorical accuracy.

To monitor the training progress and apply early stopping, an early stopping callback was defined, which monitored the categorical accuracy metric and allowed a patience of 10 epochs before stopping if no improvement was observed. The training process was carried out for a maximum of 2000 epochs, and the model was trained using the fit() function on the training dataset. Additionally, TensorBoard was utilized as a callback to visualize and monitor the process of training. The class scores were determined using the Softmax activation function. For the model, 32 and 32 hidden units per layer were set respectively. Table 4 shows

the model has a total of 226,699 parameters, all of which are trainable.

Table 4. Summary of our proposed Sequential LSTM Model

Layer (type)	Output Shape	Param #
lstm_32 (LSTM)	(None, 30, 32)	216,960
lstm_33 (LSTM)	(None, 32)	8,320
dense_30 (Dense)	(None, 32)	1,056
dense_31 (Dense)	(None, 11)	363
Total params: 226,699 Trainable params: 226,699 Non-trainable params: 0		

4.3. Evaluation metrics

To evaluate the performance of proposed Sequential LSTM model, we used the mean absolute error (MAE), mean squar ed error (MSE), and coefficient of determination (R^2) valu es displayed in Table 5.
MAE is the average of the absolute differences between the actual and predicted values of the dataset. It can be achieved by employing the subsequent formula.

$$\text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}| \quad (9)$$

The MSE formula is used to calculate the average of the squared difference between the dataset's predicted values and actual values:

$$\text{MSE} = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2 \quad (10)$$

The R^2 score ranges from 0.0 to 1.0, with 0.0 representing the least favorable fit and 1.0 denoting the most ideal fit, and evaluates the model's fit to the given dataset. It can be calculated utilizing the specified formula:

$$R^2 = 1 - \frac{\sum(y_i - \hat{y})^2}{\sum(y_i - \bar{y})^2} \quad (11)$$

where the mean value of y is denoted by $\bar{y}$ and the predicted value is represented by $\hat{y}$. Eqs. (9), (10), and (11) were used to calculate errors.

Table 5. MAE, MSE, and R^2 of proposed LSTM model

MAE	0.0303
MSE	0.0303
R^2	0.9969

Table 6. Classification result of each label of 11 ISL gesture

ISL Gesture	Precision	Recall	F1 score	Support
Accident	1.00	1.00	1.00	6
Allergy	1.00	1.00	1.00	6
Doctor	1.00	1.00	1.00	6
Hello	1.00	1.00	1.00	6
Help	0.86	1.00	0.92	6
Love	1.00	0.83	0.91	6
Money	0.86	1.00	0.92	6
Pain	1.00	0.83	0.91	6
Police	1.00	1.00	1.00	6
Thank You	1.00	1.00	1.00	6
Wait	1.00	1.00	1.00	6

4.4. Quantitative analysis

The evaluation of the predictive performance of each sign gesture involved the use of classification metrics, including precision, recall, and F1-score and computed using TP, FP, TN, and FN values.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + FP + TN + FN)} \quad (12)$$

$$\text{Precision} = \frac{\text{TP}}{(\text{TP} + \text{FP})} \quad (13)$$

$$\text{Recall} = \frac{\text{TP}}{(\text{TP} + \text{FN})} \quad (14)$$

$$\text{F1 Score} = 2 * \frac{(\text{Precision} * \text{Recall})}{(\text{Precision} + \text{Recall})} \quad (15)$$

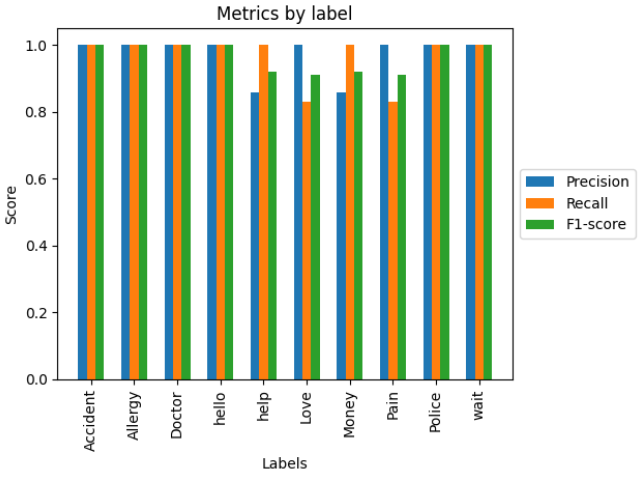

Figure 6. Evaluation metrics for each label in the 11 gestures ISL dataset

The classification report in Table 6 is obtained using the calculations provided in Eqs. (12), (13), (14), and (15). Figure 6 showcases the evaluation of each label of the 11 gestures dataset in Indian Sign Language in terms of mentioned evaluation metrics. From a thorough evaluation of classification metrics, it was evident that our Sequential LSTM model achieved remarkable precision, recall, and F1-score values, approaching 1 in most cases. While two instances of precision and recall as well as four instances of the F1-score, displayed minor deviations, they still maintained a favourable proximity to the ideal value. These results provide compelling evidence of the model's ability to adeptly learn from the entirety of the training data. When considering the R^2 value, our proposed Sequential model

attains a high score of 0.99. This score, close to 1, indicates a notable fit of the model. Insights into the training accuracy and loss of our model during the classification of 11 individual Indian sign gestures from the dataset can be obtained by referring to Figure 7. From the insights obtained from Figure 7, it can be inferred that the model we propose performs exceptionally well, showcasing a smooth and fast training process, which efficiently learns from the data. Consequently, our proposed model excels in performance, minimizing the loss to a significant extent.

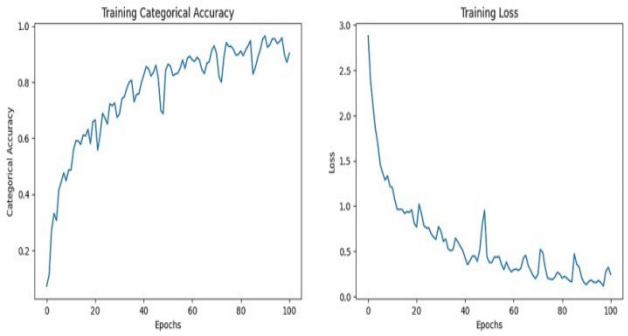

Figure 7. Progress of training accuracy and loss over 101 epochs, utilizing an early stopping callback

It can be concluded from Figure 8 that the LSTM MediaPipe architecture shows excellent performance in ISL gesture recognition, with high accuracy, precision, recall, F1 score, and low errors. Our proposed SLR model performed exceptionally well with an accuracy of 96.97%. When analyzing Table 6, the metrics selected to assess our model's performance on all of the 11 dynamic gestures of ISL, the results show exceptionally good performance. The live stream of our webcam's real-time Indian sign language detection during testing is displayed in Figure 9.

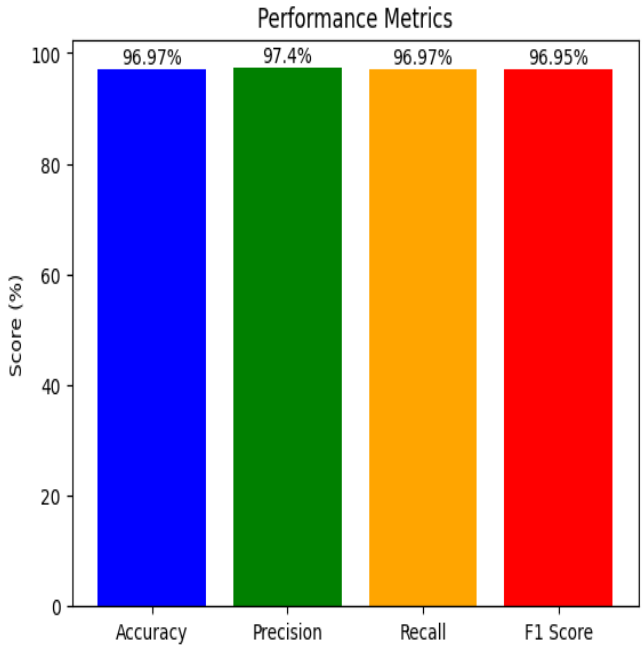

Figure 8. Performance metrics of the proposed LSTM model

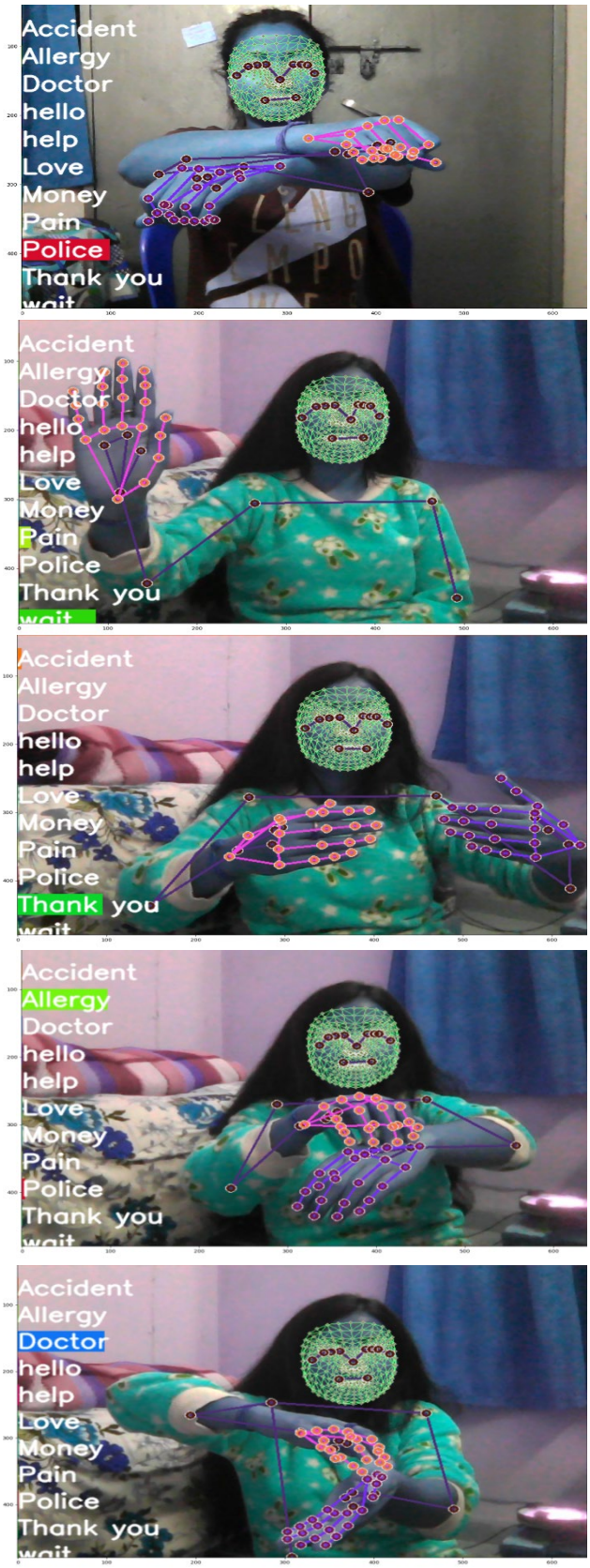

Figure 9. Live feed from a webcam and detection results

4.5. Comparison of proposed approach with the existing approaches

In our experiments, we implemented and evaluated several models on our custom dataset of 11 gestures to discover the most effective method for sign language recognition. We compared our proposed approach with Basic Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), Support Vector Machines (SVM), k-Nearest Neighbors (k-NN), and Basic LSTM without optimizations. Our proposed method produced the highest level of accuracy (96.97%), precision (97.4%), recall (96.97%), and F1-score (96.95%), as determined by the results gathered. This superior performance is mostly attributable to the model's capacity to capture temporal and spatial relationships in gesture sequences, which is further improved by the use of early stopping techniques and SELU activation.

Table 7. Evaluation Metrics for Different Models

Methodology	Accuracy	Precision	F1-Score	Recall
SVM	78%	83.0%	82.7%	81.2%
k-NN	75.0%	76.5%	77.2%	76.8%
Basic RNN	82.5%	83.0%	85.2%	84.5%
CNN	93.5%	92.40%	91.5%	92.8%
Basic LSTM	92.5%	93.0%	93.7%	93.2%
Proposed	96.97%	97.4%	96.95%	96.97%

In contrast, the vanishing gradient problem caused Basic RNN to struggle with long-term dependencies, leading to an accuracy of 82.5%. CNN, while adept at capturing spatial features, lacked the ability to represent temporal dependencies, resulting in an accuracy of 93.5.0%. Complex patterns in sequential data were harder to identify using traditional machine learning techniques like SVMs and k-NN, which had accuracy rates of 78.0% and 75.0%, respectively. Basic LSTM without optimizations outperformed these techniques but fell short of our proposed model, obtaining an accuracy of 92.5%.

A detailed comparison of the evaluation metrics for each model is provided in Table 7. Taking everything into account, our suggested Sequential LSTM model with MediaPipe Holistic showed robustness and excellent performance, so qualifying it as the perfect choice for sign language recognition uses.

***Impact of Environmental conditions on Model Performance*:** Tests showed that the rec recognition accuracy is somewhat affected by partial occlusion and lighting changes. In low light, the model outperforms existing techniques in obtaining necessary features. We used various levels of occlusion and evaluated their impact on accuracy. In minor occlusions, when just a little piece of the hand was blocked, accuracy dropped by 5–10%. On the other hand, major occlusions completely hid vital landmarks, which caused an identification accuracy drop of at least 10%. These results highlight the need for adaptive pre-processing methods like dynamic contrast modifications and occlusion-aware models. The dataset may be augmented to include a wider range of lighting settings and occlusion situations to improve robustness and guarantee consistent performance in real uses.

4.6. Discussion on Limitations.

Operating independently of individual signers, our model efficiently identifies 11 isolated gestures. Still, actual implementation presents several challenges.

Computational Constraints: A system with an Intel Core i5 CPU and 8GB RAM ran the model through training and testing. Although the design is simple, real-time inference on edge devices or mobile platforms may need more optimization, such as quantization or pruning, to reduce the computational cost. Training on larger datasets also requires notable processing power, which might be enhanced using the best training techniques.

Dataset Limitations: Dataset Constraints: The dataset was collected under a range of lighting settings to guarantee adaptability to different illumination levels. Real-world scenarios might impact recognition accuracy as signers could gesture in chaotic environments or with backdrop distractions. Increasing the dataset to include a broad spectrum of gestures and environments will help to further generalize and strengthen the model.

Despite these limitations, our approach remains very scalable. Future advances in dataset expansion and model optimization will help to further increase the system's flexibility hence making it more appropriate for practical uses.

5. Future Work

SLR can be expanded in the future by including several modalities like vision, depth, and audio, therefore aiming to raise the preciseness and dependability of SLR models. Furthermore, the use of larger and more varied sign language video datasets can significantly improve the competency and adaptability of the models. Moreover, the future offers opportunities to look into SLR applications in other fields like education, healthcare, and assistive technology. Furthermore, it is vital to consider the cultural and geographical variations in sign languages as these might increase the relevance and efficacy of SLR systems even more.

Multilingual Adaptability and Scalability:

The lightweight design of our methodology facilitates real-time recognition of sign language. It must, however, incorporate many sign languages and an expanded vocabulary to enhance its adaptability. Transfer learning enhances scalability by allowing the model to swiftly adapt to new signals with minimum training. Furthermore, hierarchical classification might enhance the organization and recognition of gestures. Moreover, dynamic feature

extraction may improve flexibility with diverse signing styles and gestures.
Transformer-based models, domain adaptability, and training on several datasets for multilingual recognition help the system to understand sign languages from several linguistic origins. Future modifications to increase the applicability and inclusiveness of the model will focus on these components.

6. Conclusion

This paper presents a Sequential LSTM model for Indian Sign Language (ISL) identification. Designed with fewer layers and neurons, the model uses the SELU activation function to increase accuracy. Additionally, by eliminating irrelevant information from the dataset, the model effectively focuses on essential features, thereby improving overall performance. The proposed approach enhances the processing capacity, learning efficiency, and predictive accuracy of conventional LSTM networks through a cost-effective implementation.
Experimental results demonstrate that the Sequential LSTM model achieves remarkably low MAE and MSE values, along with high R-squared scores, confirming its superior performance. Especially, the Sequential LSTM MediaPipe model's capacity to grasp temporal dependencies in time series data produced an amazing prediction accuracy of 96.97% along with faster convergence.
However, the study is constrained by the dataset size. Future studies will aim to increase the dataset with a more comprehensive vocabulary to enable the prediction of continuous sign language sentences. Aiming to improve both learning efficiency and recognition accuracy, this improvement pushes the creation of strong ISL recognition systems forward. Our suggested method efficiently identifies isolated gestures; real-world sign language runs continuously without obvious interruptions. This makes it difficult to know where one sign finishes and the next starts. Future developments might thus concentrate on more intelligent techniques for segmentation and advanced models like transformers to better grasp the context. Including linguistic cues will also enable the algorithm to more naturally comprehend sign sequences, hence facilitating more accurate recognition. These advancements will move the system closer to real-time CSLR, making it more practical for everyday use.

References

[1] "Deafness and hearing loss," World Health Organization. Accessed: Apr. 26, 2023. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/deafness-and-hearing-loss

[2] Yousaf K, Mehmood Z, Saba T, Rehman A, Rashid M, Altaf M, Shuguang Z. A Novel Technique for Speech Recognition and Visualization Based Mobile Application to Support Two-Way Communication between Deaf-Mute and Normal Peoples. Wireless Communications and Mobile Computing. 2018;2018(1):1013234.

[3] Martins TM. *A letra e o gesto: estruturas linguísticas em Língua Gestual Portuguesa e Língua Portuguesa*,"2011.

[4] Oliveira T, Escudeiro P, Escudeiro N, Rocha E, Barbosa FM. Automatic sign language translation to improve communication. In2019 IEEE Global Engineering Education Conference (EDUCON) 2019 Apr 8 (pp. 937-942). IEEE.

[5] Richardson JT, Barnes L, Fleming J. Approaches to studying and perceptions of academic quality in deaf and hearing students in higher education. Deafness & Education International. 2004 Jun;6(2):100-22.

[6] Riddell S, Weedon E. Disabled students in higher education: Discourses of disability and the negotiation of identity. International Journal of Educational Research. 2014 Jan 1;63:38-46.

[7] S. C. Daroque and A. M. L. Padilha, "Alunos Surdos no Ensino Superior: Uma Discussão Necessária," *Comunicações*, vol. 19, no. 2, pp. 23–32, Dec. 2012, doi: 10.15600/2238-121X/comunicacoes.v19n2p23-32.

[8] Ghotkar AS, Kharate GK. Dynamic hand gesture recognition and novel sentence interpretation algorithm for indian sign language using microsoft kinect sensor. Journal of pattern recognition research. 2015 Jul;1:24-38.

[9] Wang RY, Popović J. Real-time hand-tracking with a color glove. ACM transactions on graphics (TOG). 2009 Jul 27;28(3):1-8.

[10] Deora D, Bajaj N. Indian sign language recognition. In2012 1st international conference on emerging technology trends in electronics, communication & networking 2012 Dec 19 (pp. 1-5). IEEE.

[11] Cheng H, Yang L, Liu Z. Survey on 3D hand gesture recognition. IEEE transactions on circuits and systems for video technology. 2015 Aug 18;26(9):1659-73.

[12] Prisacariu VA, Reid I. Robust 3D hand tracking for human computer interaction. In2011 IEEE International Conference on Automatic Face & Gesture Recognition (FG) 2011 Mar 21 (pp. 368-375). IEEE.

[13] Suarez J, Murphy RR. Hand gesture recognition with depth images: A review. In2012 IEEE RO-MAN: the 21st IEEE international symposium on robot and human interactive communication 2012 Sep 9 (pp. 411-417). IEEE.

[14] Kapuscinski T, Oszust M, Wysocki M, Warchol D. Recognition of hand gestures observed by depth cameras. International Journal of Advanced Robotic Systems. 2015 Apr 14;12(4):36.

[15] Dong C, Leu MC, Yin Z. American sign language alphabet recognition using microsoft kinect. InProceedings of the IEEE conference on computer vision and pattern recognition workshops 2015 (pp. 44-52).

[16] Kim K, Kim SK, Choi HI. Depth based sign language recognition system using SVM. Int. J.

Multimed. Ubiquitous Eng. 2015 Feb;10(2):75-86..
[17] Tripathi K, Baranwal N, Nandi GC. Continuous dynamic Indian Sign Language gesture recognition with invariant backgrounds. In 2015 international conference on advances in computing, communications and informatics (ICACCI) 2015 Aug 10 (pp. 2211-2216). IEEE.
[18] Adithya V, Vinod PR, Gopalakrishnan U. Artificial neural network based method for Indian sign language recognition. In2013 IEEE conference on information & communication technologies 2013 Apr 11 (pp. 1080-1085). Ieee.
[19] Sharma M, Pal R, Sahoo AK. Indian sign language recognition using neural networks and KNN classifiers. ARPN journal of Engineering and Applied Sciences. 2014 Aug;9(8):1255-9.
[20] Kumar A, Thankachan K, Dominic MM. Sign language recognition. In2016 3rd international conference on recent advances in information technology (RAIT) 2016 Mar 3 (pp. 422-428). IEEE.
[21] Hussain I, Talukdar AK, Sarma KK. Hand gesture recognition system with real-time palm tracking. In2014 Annual IEEE India Conference (INDICON) 2014 Dec 11 (pp. 1-6). IEEE.
[22] Patil SB, Sinha GR. Distinctive feature extraction for Indian Sign Language (ISL) gesture using scale invariant feature Transform (SIFT). Journal of The Institution of Engineers (India): Series B. 2017 Feb;98(1):19-26.
[23] Akhter S. Orientation hashcode and articial neural network based combined approach to recognize sign language. In 2018 21st International Conference of Computer and Information Technology (ICCIT) 2018 Dec 21 (pp. 1-5). IEEE.
[24] Aly W, Aly S, Almotairi S. User-independent American sign language alphabet recognition based on depth image and PCANet features. IEEE Access. 2019 Sep 2;7:123138-50.
[25] Tao W, Leu MC, Yin Z. American Sign Language alphabet recognition using Convolutional Neural Networks with multiview augmentation and inference fusion. Engineering Applications of Artificial Intelligence. 2018 Nov 1;76:202-13.
[26] Chong TW, Kim BJ. American sign language recognition system using wearable sensors with deep learning approach. The Journal of the Korea institute of electronic communication sciences. 2020;15(2):291-8.
[27] Abraham E, Nayak A, Iqbal A. Real-time translation of Indian sign language using LSTM. In2019 global conference for advancement in technology (GCAT) 2019 Oct 18 (pp. 1-5). IEEE.
[28] Gupta R, Kumar A. Indian sign language recognition using wearable sensors and multi-label classification. Computers & Electrical Engineering. 2021 Mar 1;90:106898.
[29] Kaur B, Joshi G, Vig R. Identification of ISL alphabets using discrete orthogonal moments. Wireless Personal Communications. 2017 Aug;95:4823-45.
[30] Kumar A, Kumar R. A novel approach for ISL alphabet recognition using Extreme Learning Machine. International Journal of Information Technology. 2021 Feb;13(1):349-57.
[31] Xiao Q, Qin M, Yin Y. Skeleton-based Chinese sign language recognition and generation for bidirectional communication between deaf and hearing people. Neural networks. 2020 May 1;125:41-55.
[32] Hu L, Gao L, Liu Z, Feng W. Continuous sign language recognition with correlation network. InProceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2023 (pp. 2529-2539).
[33] Zhao W, Hu H, Zhou W, Mao Y, Wang M, Li H. Masa: Motion-aware masked autoencoder with semantic alignment for sign language recognition. IEEE Transactions on Circuits and Systems for Video Technology. 2024 Jun 5.
[34] Gao L, Shi P, Hu L, Feng J, Zhu L, Wan L, Feng W. Cross-modal knowledge distillation for continuous sign language recognition. Neural Networks. 2024 Nov 1;179:106587.
[35] Saproo V, Aggarwal RK. A Transformer Based Indian Signed Language Recognition. In2024 First International Conference on Pioneering Developments in Computer Science & Digital Technologies (IC2SDT) 2024 Aug 2 (pp. 170-174). IEEE.
[36] Sandoval-Castaneda M, Li Y, Brentari D, Livescu K, Shakhnarovich G. Self-supervised video transformers for isolated sign language recognition. arXiv preprint arXiv:2309.02450. 2023 Sep 2.
[37] S. Hochreiter S, Schmidhuber J. Long short-term memory. Neural computation. 1997 Nov 15;9(8):1735-80.
[38] Gers FA, Schmidhuber J, Cummins F. Learning to forget: Continual prediction with LSTM. Neural computation. 2000 Oct 1;12(10):2451-71.
[39] Graves A, Mohamed AR, Hinton G. Speech recognition with deep recurrent neural networks. In 2013 IEEE international conference on acoustics, speech and signal processing 2013 May 26 (pp. 6645-6649). Ieee.
[40] Kingma DP, Ba J. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. 2014 Dec 22.