



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

# IJIRSET

International Journal of Innovative Research in  
**SCIENCE | ENGINEERING | TECHNOLOGY**

# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 12, Special Issue 2, April 2023

**ISSN** INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INDIA

9940 572 462

6381 907 438

[ijirset@gmail.com](mailto:ijirset@gmail.com)

[www.ijirset.com](http://www.ijirset.com)



# Sign Language Recognition using Action Recognition with LSTM Deep Learning Model

Almelu S, Pravin Kumar R, Ganesh S, Rishikumar R,

Assistant Professor, Department of Computer Science and Engineering, Velammal Engineering College, Chennai, Tamil Nadu, India

U.G. Student, Department of Computer Science and Engineering, Velammal Engineering College, Chennai, Tamil Nadu, India

U.G. Student, Department of Computer Science and Engineering, Velammal Engineering College, Chennai, Tamil Nadu, India

U.G. Student, Department of Computer Science and Engineering, Velammal Engineering College, Chennai, Tamil Nadu, India

**ABSTRACT:** Sign language is a visual language that is mainly used by deaf and people with hearing defects to communicate with each other and with people who could hear without any difficulties. In recent years, sign language has gained more recognition as a legitimate language and become popular in various fields. Sign language varies in structure and syntax depending on the country or region with each having its own unique grammar and vocabulary and this difference from region to region makes the sign recognition models difficult to build. In this paper we are recognizing Sign language using deep learning models. Sign language recognition systems are designed to provide more access and better communication for individuals who are deaf and hard of hearing. The system first detects the key points of the model and uses the trained LSTM Deep Learning models to make real time predictions based on the key points detected from the model. These systems have a great potential to improve communication and create more opportunities for individuals who can only use sign language to participate in society. To recognize the sign language we used technologies like OpenCV which lets the computer to use its sensors to capture data from life or from real-time, MediaPipe which is a machine learning framework used to tell the computer what the data is about, and LSTM framework to make the system learn from the provided data from which we can make the system to predict the result in real-time.

**KEYWORDS:** Sign Language, Deep learning. Sign recognition.

## I.INTRODUCTION

Sign language recognition is a rapidly evolving field that aims to bridge the communication gap between people having difficulties in hearing and those who don't. It involves the use of technology, particularly machine learning and computer vision, to recognize and interpret sign language gestures. The development of sign language recognition system has the potential to greatly enhance accessibility and inclusion for the deaf and hard of hearing community. It can be used to enhance the quality of life for the deaf and hard of hearing peoples. However, sign language recognition is a complex and challenging task due to its diversity and variability of sign languages across different regions and individuals. So, this might require the use of advanced machine learning techniques to accurately interpret and recognize the subtle nuances of sign language gestures. Thus we made use of LSTM deep learning – a variety of Recurrent Neural Networks (RNNs) which helps us to make better decision in recognizing sign language gestures in real-time.

Long Short-Term Memory (LSTM) is a type of deep learning algorithm that has given promising results in various fields, including computer vision and natural language processing. It is particularly suitable for analyzing sequences of data and thus making an ideal choice for sign language recognition. It includes training the model with a large dataset of sign language gestures with their corresponding meanings. The system includes training the model with a sequence of data for each gesture. These data are passed to the LSTM model where it is preprocessed using neural networks. The LSTM processes the data using deep learning neural networks to produce the data, which is a probability between 0 to 1 with the help of which we make a decision on the sign language gestures in real-time. The more data sequence we provide, the more accurate the result will be. In our system, we passed 30 sequences of data, which is enough to produce a decent accuracy. We chose LSTM because it is faster to train and to make decisions. It also can be built with less amount of data, which is another added advantage. Before the data are passed to TensorFlow's LSTM model, they were preprocessed by MediaPipe to mark the key points.

MediaPipe is an open source framework which is used to perform computer vision interface over arbitrary sensory data such as video or audio. In our system, we used MediaPipe to plot the key-point values from the provided data. The output from the MediaPipe will be as an object, which is for convenience then converted to an array type using NumPy. The LSTM can do its work only with the help of numerical data and not just with frames from a video. As technology continues to advance, we can get to see further improvements in sign language recognition systems and their applications.

Paper is organized as follows. Section II describes the systems related to Sign language recognition from the past followed by methodology used in this system. Section IV shows the journey of the system from training the model to testing in real-time. And finally, a conclusion on this Sign language recognition system.

## II. RELATED WORKS

Sharma et al. [1] developed a system on the basis of the Indian Sign Language for communicating with persons who had impaired speech or hearing. First, the data were pre-processed after capturing the image by converting it from RGB to grayscale, using a Matlab environment [1]. Then, a Sobel detector was used to detect the edges of the image using a  $3 \times 3$  filter. Finally, a hierarchical centroid method was applied to the reduced image with 600 elements; 124 features were obtained using this method. The used classification techniques were KNN and neural networks. The accuracy obtained by this methodology was 97.10%.

Mariappan and Gomathi [2] proposed a novel approach for recognizing real-time signing for Indian Sign Language (ISL) [2]. For this, they proposed an approach consisting of a region of interest (ROI)-based method for identifying and tracking signs using the skin segmentation feature of OpenCV. For sign prediction, they employed a fuzzy C-means (FCM) clustering technique. They collected a dataset of 80 words and 50 sentences from 10 different people for training and testing purposes. To reduce noise from the digital source, they also applied filtering on the colored images and morphological operations on the binary image generated after applying skin segmentation to enhance the features. As a result, they achieved an accuracy of 75% in recognizing 40 words from ISL using the FCM technique.

Agarwal et al. [3] aimed to bridge the gap between people with speech impairment and people with normal speech abilities by using a sensor glove for signing, processing those signs, and presenting the output in a meaningful sentence. The subjects performed the gestures using the sensor gloves [3]. The gestures were matched with the database, and then the recognized gesture was passed on to be parsed to generate a sentence. In version 1 of the application, the accuracy was 33.33%. In version 2, a keyword was added for the required tense; hence, 100% accuracy was achieved to tackle simple and continuous tenses.

Mekala et al. [4] proposed neural network architecture for recognizing and tracking sign language and generating real-time text from the video stream. The system architecture consists of different phases, such as framing, pre-processing images, extracting the features based on hand movement and hand position, etc. These hand attributes are represented using a point of interest (POI) of the hand [4]. By using this method, 55 different features were extracted, which were used as an input for the neural network architecture proposed by the authors, consisting of CNN layers that predicted the signs. In their study, they trained and tested the model on the English alphabet, from A to Z, and claimed to achieve a 100% recognition rate and 48% of noise immunity.



Chen [5] proposed a model with a self-made dataset of 1200 samples of ten static signs/letters. In order to recognize the gestures, pre-processing was performed initially through edge segmentation to detect the edges of the hand, and for skin color segmentation, the approach was to convert RGB images to YUQ or YIQ color spaces [5]. Then, the fingers were detected from the already identified hand using the convex hull method. Finally, neural networks were used as the classification method. The final accuracy obtained by this model was 98.2%.

Camgoz et al. [6] proposed a system that showed how sign language and its translation into spoken language is not one-to-one mapping but instead sequence-to-sequence mapping. The authors introduced a new vision method in which they mirrored the tokenization and embedding steps of the standard neural machine translation. The neural machine translation step, which translates sign videos to spoken language, integrates the CNN architecture [6] with an attention-based encoder and decoder that models the conditional probability of generating a spoken language from a given signing video. The authors started from word embedding and translated a sparse vector into a denser form where words with similar meanings were closer. The encoder-decoder phase was used to maximize the conditional probability. Encoding generated a fixed-size vector of the sign videos' features. The inputs were the previously hidden state and word embedding in the decoding phase. This helped to predict the next word. In addition to this, to avoid the problems of long-term dependencies and vanishing gradients, the authors added an attention mechanism in the decoding phase. They created a continuous sign language translation dataset called PHOENIX14T.

Mittal et al. [7] proposed a system that used an LSTM model with leap motion for continuous sign language recognition [7]. They employed a four-gated LSTM cell with a 2D CNN architecture for sign sentence recognition and assigned a particular label to every word. When the three basic gates were used as an input gate, a forget gate that took the output at the t-1 time generated output at the output gate and returned the label showing the word that was specifically associated with that label. They added a special symbol, \$, which indicated the transition between the two consecutive signs in the video. The RESET flag was used when they encountered the transition between two signs, which was indicated by the \$ sign [7]. They trained this modified LSTM model over a dataset and achieved a maximum accuracy of 72.3% using a 3-layer LSTM model for sign sentence recognition. For sign word recognition, the achieved accuracy was 89.50%.

Indian Sign Language is a visual-spatial language that was developed in India. Indian Sign Language is a natural language with its own phonology, morphology, and grammar. It uses arm motions, hands, facial expressions, and the body/head, which generate semantic information conveying words and emotions [8].

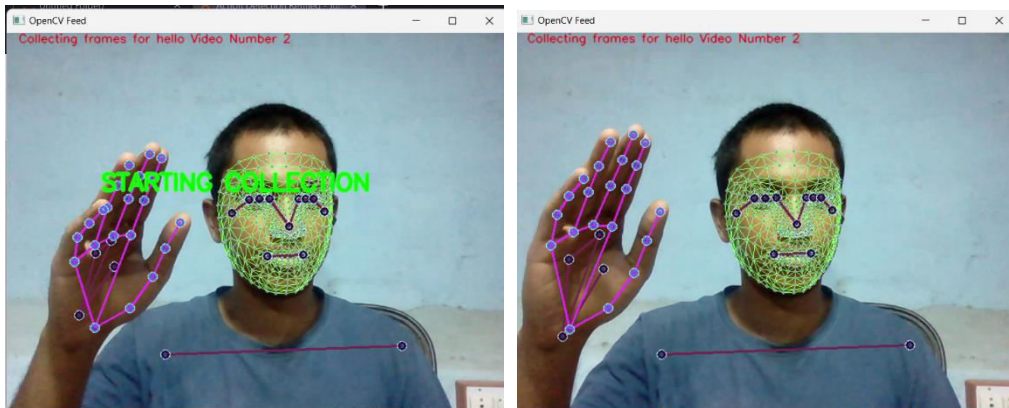
De Coster et al. [9] used a transformer network to recognize sign language from videos, including non-manual parts such as the mouth and eyebrow orientation [9]. They proposed a pose transformer network, video transformer network, and multimodal transformer network to recognize signs over different neural networks.

### III.METHEDODOLOGY

To collect key points for training the model we capture 'n' number of videos. Here we used videos instead of images because sign language has gestures motions included and most of them are actions. For each action from sign language we captured 30 videos of 30 frames each. We set a time gap between each video so that we can get to position from previous pose. The key-points from each frame are collected using MediaPipe holistic as a NumPy Array where the key points are stored. The key-points produced here will be in a two-dimensional array, which cannot be used to train the LSTM model using tensor flow. So, we preprocessed the key-points into three-dimensional array containing sequence, frames and key-points. Using tensor flow's Kera's Sequential API we created 3 layers of LSTM models. These models are stacked up together and the sequence produced by one layer is passed on to the next layer and so on followed by 3 dense layers. Dense layer is just another neural network that has deep connection, meaning that each neuron in dense layer receives input from all neurons of its previous layer. The resulting data from the LSTM model will be in the dimension of our number of actions and with the probability value of 0 to 1. A similar probability value will be generated by processing the real-time data which can be used to compare with the final value from the LSTM model and thus we can make the system decide between which sign language gesture is being performed before our detection system.

### IV. EXPERIMENTAL RESULTS

Figures 1(a), 1(b) shows the training process, where we made an indication when the system starts reading the data so that we can get to know whether the system is reading or not. Once the indication went off, every frame will be captured for further process. This process will be repeated 30 times for each action, in our case each session captures a total of 30 frames and these frames are used to map keypoints on required areas. These generated key points are passed to train using LSTM model.



1(a)

1(b)

Figure 2(a) and 2(b) shows the keypoints generated by the MediaPipe with the provided video samples. These data are which is then passed to LSTM framework for further processing.

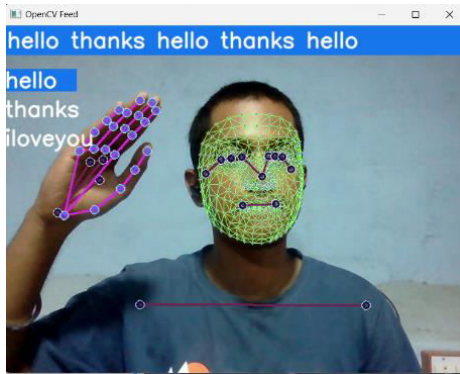
Name	Date modified	Type	Size
0	17/06/2021 12:30 PM	File folder	
1	17/06/2021 12:31 PM	File folder	
2	17/06/2021 12:31 PM	File folder	
3	17/06/2021 12:31 PM	File folder	
4	17/06/2021 12:31 PM	File folder	
5	17/06/2021 12:31 PM	File folder	
6	17/06/2021 12:31 PM	File folder	
7	17/06/2021 12:31 PM	File folder	
8	17/06/2021 12:31 PM	File folder	
9	17/06/2021 12:31 PM	File folder	
10	17/06/2021 12:31 PM	File folder	
11	17/06/2021 12:31 PM	File folder	
12	17/06/2021 12:31 PM	File folder	
13	17/06/2021 12:31 PM	File folder	
14	17/06/2021 12:31 PM	File folder	
15	17/06/2021 12:31 PM	File folder	
16	17/06/2021 12:31 PM	File folder	
17	17/06/2021 12:32 PM	File folder	
18	17/06/2021 12:32 PM	File folder	
19	17/06/2021 12:32 PM	File folder	
20	17/06/2021 12:32 PM	File folder	
21	17/06/2021 12:32 PM	File folder	
22	17/06/2021 12:32 PM	File folder	
23	17/06/2021 12:32 PM	File folder	
24	17/06/2021 12:32 PM	File folder	
25	17/06/2021 12:32 PM	File folder	
26	17/06/2021 12:32 PM	File folder	
27	17/06/2021 12:32 PM	File folder	
28	17/06/2021 12:32 PM	File folder	
29	17/06/2021 12:32 PM	File folder	

2(a)

Name	Date modified	Type	Size
0.npy	17/06/2021 12:30 PM	NPV File	14 KB
1.npy	17/06/2021 12:30 PM	NPV File	14 KB
2.npy	17/06/2021 12:30 PM	NPV File	14 KB
3.npy	17/06/2021 12:30 PM	NPV File	14 KB
4.npy	17/06/2021 12:30 PM	NPV File	14 KB
5.npy	17/06/2021 12:30 PM	NPV File	14 KB
6.npy	17/06/2021 12:30 PM	NPV File	14 KB
7.npy	17/06/2021 12:30 PM	NPV File	14 KB
8.npy	17/06/2021 12:30 PM	NPV File	14 KB
9.npy	17/06/2021 12:30 PM	NPV File	14 KB
10.npy	17/06/2021 12:30 PM	NPV File	14 KB
11.npy	17/06/2021 12:30 PM	NPV File	14 KB
12.npy	17/06/2021 12:30 PM	NPV File	14 KB
13.npy	17/06/2021 12:30 PM	NPV File	14 KB
14.npy	17/06/2021 12:30 PM	NPV File	14 KB
15.npy	17/06/2021 12:30 PM	NPV File	14 KB
16.npy	17/06/2021 12:30 PM	NPV File	14 KB
17.npy	17/06/2021 12:30 PM	NPV File	14 KB
18.npy	17/06/2021 12:30 PM	NPV File	14 KB
19.npy	17/06/2021 12:30 PM	NPV File	14 KB
20.npy	17/06/2021 12:30 PM	NPV File	14 KB
21.npy	17/06/2021 12:30 PM	NPV File	14 KB
22.npy	17/06/2021 12:30 PM	NPV File	14 KB
23.npy	17/06/2021 12:30 PM	NPV File	14 KB
24.npy	17/06/2021 12:30 PM	NPV File	14 KB
25.npy	17/06/2021 12:30 PM	NPV File	14 KB
26.npy	17/06/2021 12:30 PM	NPV File	14 KB
27.npy	17/06/2021 12:30 PM	NPV File	14 KB
28.npy	17/06/2021 12:30 PM	NPV File	14 KB
29.npy	17/06/2021 12:30 PM	NPV File	14 KB

2(b)

Figure 3(a), 3(b), and 3(c) shows the real-time sign detection by this system. Here, the result value generated from real-time is compared with the probability values of each gesture generated by deep learning neural networks by feeding some data samples while training. We had also added an UI element where the sign language is displayed with its corresponding meaning in a sentence, which helps the user to read and understand easily.



3(a)

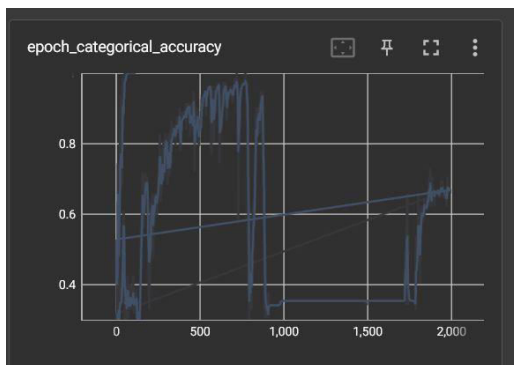


3(b)

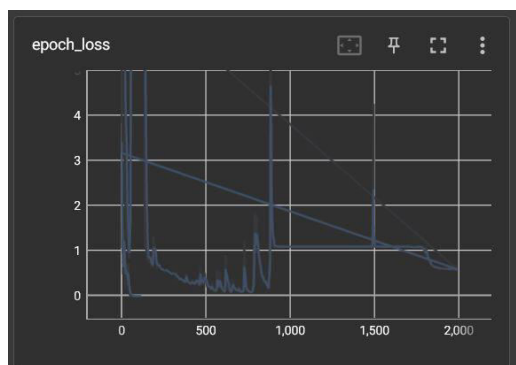


3(c)

Figures 4(a), 4(b) shows the graph on how the data is preprocessed by the LSTM model and the metric of the model such as the categorical accuracy and loss are displayed. These datas are fetched from Tensorboard.



4(a)



4(b)

## V.CONCLUSION

We have developed a sign language recognition system using deep learning techniques in Python. Our System was able to accurately recognize signs from Sign Language. We trained the model on a large dataset to support recognizing American Sign Language. With further development and refinement, our model could be used to create user-friendly and accessible systems for real-time sign language communication and translation which can be shipped to the end-



users. As further advancements made in this field, sign language recognition systems with deep learning models have great potential to provide better communication experience for peoples who depends on sign language as their primary mode of communication. Increasing availability of these digital technologies, these types of systems can provide more opportunities for individuals who are being deaf and facing difficult in hearing.

## REFERENCES

- [1] Sharma, M.; Pal, R.; Sahoo, A. Indian sign language recognition using neural networks and KNN classifiers. *J. Eng. Appl. Sci.* 2014, 9, 1255–1259.
- [2] Muthu Mariappan, H.; Gomathi, V. Real-Time Recognition of Indian Sign Language. In Proceedings of the International Conference on Computational Intelligence in Data Science, Haryana, India, 6–7 September 2019.
- [3] Agarwal, S.R.; Agrawal, S.B.; Latif, A.M. Article: Sentence Formation in NLP Engine on the Basis of Indian Sign Language using Hand Gestures. *Int. J. Comput. Appl.* 2015, 116, 18–22.
- [4] Mekala, P.; Gao, Y.; Fan, J.; Davari, A. Real-time sign language recognition based on neural network architecture. In Proceedings of the IEEE 43rd Southeastern Symposium on System Theory, Auburn, AL, USA, 14–16 March 2011.
- [5] Chen, J.K. Sign Language Recognition with Unsupervised Feature Learning; CS229 Project Final Report; Stanford University: Stanford, CA, USA, 2011.
- [6] Camgoz, N.C.; Hadfield, S.; Koller, O.; Ney, H.; Bowden, R. Neural Sign Language Translation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2018, Salt Lake City, UT, USA, 18–22 June 2018; IEEE: Piscataway, NJ, USA, 2018.
- [7] Mittal, A.; Kumar, P.; Roy, P.P.; Balasubramanian, R.; Chaudhuri, B.B. A Modified LSTM Model for Continuous Sign Language Recognition Using Leap Motion. *IEEE Sens. J.* 2019, 19, 7056–7063.
- [8] Papastratis, I.; Chatzikonstantinou, C.; Konstantinidis, D.; Dimitropoulos, K.; Daras, P. Artificial Intelligence Technologies for Sign Language. *Sensors* 2021, 21, 5843.
- [9] De Coster, M.; Herreweghe, M.V.; Dambre, J. Sign Language Recognition with Transformer Networks. In Proceedings of the Conference on Language Resources and Evaluation (LREC 2020), Marseille, France, 13–15 May 2020; pp. 6018–6024.





**SJIF: 8.423**



INTERNATIONAL  
STANDARD  
SERIAL  
NUMBER  
INTERNATIONAL CENTRE



# INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 **9940 572 462**  **6381 907 438**  **ijirset@gmail.com**



[www.ijirset.com](http://www.ijirset.com)

Scan to save the contact details