International Journal of Scientific Research in Computer Science, Engineering and Information Technology

ISSN : 2456-3307 OPEN ACCESS

Available Online at : www.ijsrcseit.com

doi : https://doi.org/10.32628/CSEIT2390517

Enhancing Accessibility with LSTM-Based Sign Language Detection

Azees Abdul [1], Adithya Valapa[2], Abdul Kayom Md Khairuzzaman[3]

[1, 2, 3] School of Electronics Engineering, VIT-AP University, Amaravati, Andhra Pradesh, India

ARTICLEINFO

Article History:

Accepted: 05 Sep 2023

Published: 22 Sep 2023

Publication Issue

Volume 9, Issue 5

September-October-2023

Page Number

130-139

ABSTRACT

Sign language serves as a vital means of communication for the deaf and hard of hearing community. However, identifying sign language poses a significant challenge due to its complexity and the lack of a standardized global framework. Recent advances in machine learning, particularly Long Short-Term Memory (LSTM) algorithms, offer promise in the field of sign language gesture recognition. This research introduces an innovative method that leverages LSTM, a type of recurrent neural network designed for processing sequential input. Our goal is to create a highly accurate system capable of anticipating and reproducing sign language motions with precision. LSTM's unique capabilities enhance the recognition of complex gestures by capturing the temporal relationships and fine details inherent in sign language. The results of this study demonstrate that LSTM-based approaches outperform existing state-of-the-art techniques, highlighting the effectiveness of LSTM in sign language recognition and their potential to facilitate communication between the deaf and hearing communities.

Keywords : Long short term memory, recurrent neural network, sign language identification

I. INTRODUCTION

Throughout human history, from the Stone Age to our modern era of economic prosperity, communal cooperation has been a cornerstone of our progress. Communication has been the glue binding these communities together. In these groups, knowledge sharing transpires through language, the medium of communication. Each region has its own language, fostering free communication among its inhabitants. However, not everyone is endowed with the power of speech. There are individuals in our world who are deaf and mute, with an estimated 300 million being deaf and 1 million mute as of 2023 (according to the World Health Organization - WHO). To integrate the deaf and mute community into our society, an innovative form of communication has been devised to facilitate interactions with others and within their own community. This gave rise to sign language.

1.1 Challenges of Sign Language

Sign language serves as the primary mode of communication for those with hearing impairments. For deaf individuals to fully participate in society and engage in daily activities, they must comprehend and communicate in sign language. Yet, due to its intricacy, diversity, and the absence of a global standard, recognizing sign language remains a formidable task. A recent survey revealed that only 5% of the global population is proficient in sign language. Learning sign language can be challenging, given its complexity and limited usage, which is often inconsistent.

In recent years, there has been promising progress in sign language motion detection, driven by machine learning algorithms like Long Short-Term Memory (LSTM) [8]. This signifies a step forward in making sign language more accessible and understandable to a broader audience.

From the above discussion, it is evident that sign language, though not widely used, plays a critical role as the primary means of communication for the deaf and mute. To enhance its accessibility, a system must be developed to recognize the signs being made. This project's primary objective is to create such a system [1,2,3]. The central goal is to develop a system capable of predicting real-life sign language gestures and displaying them on a screen. Moreover, the project aims for high accuracy [4,5,6,7]. It is designed to be cost-effective, accurate, and time-efficient, with optimization planned for future enhancements to ensure smooth performance on low-end computers.

II. Literature Survey

In a study referenced in [9], the field of Bangla sign language recognition is explored, highlighting the scarcity of research achieving high accuracy. A modified convolutional neural network (CNN) approach is introduced, demonstrating significant accuracy in recognizing digits, alphabets, and combined sign usage. The study underscores the importance of accurate recognition and suggests avenues for future research.

[10] focuses on the development and implementation of an application for recognizing American Sign Language (ASL) signs using deep learning algorithms based on convolutional neural network (CNN) architectures. The network achieves a remarkable 99% accuracy rate for the training set. The article outlines the tools, libraries, and a dataset of 50,000 ASL letter photos used in the study.

The study in [11] delves into the challenges faced by researchers in developing real-time sign language recognition systems. It reviews various machine learning methodologies and their comparative performance, aiming to identify the most accurate and effective approach. This survey informs future directions in sign language recognition research.

In [12], a hand gesture recognition system is proposed, encompassing hand tracking, feature extraction, hidden Markov model (HMM) training, and gesture detection. Various methodologies, including neural networks, finite state machines, and HMMs, are explored. The proposed approach achieves a recognition rate of over 90% for 20 different gestures.

Authors in [13] present a sign language recognition system, specifically tailored for Arabic Sign Language. Leveraging a visual hand dataset of Arabic sign language alphabets and applying preprocessing techniques and data augmentation, the EfficientNetB4 model achieves a maximum accuracy of 98% in training and 95% in testing. This study highlights the potential of automated sign language interpretation in improving communication for sign language users.

[14] offers a review of research on intelligent systems in sign language recognition (SLR) over the past two decades. It examines 649 articles related to decision support and intelligent systems in SLR, exploring chronological and geographical distributions, collaboration networks, and institutions involved. The evaluation underscores the need for implementing intelligent solutions in SLR systems and acknowledges the ongoing challenge of achieving ideal intelligent systems for SLR. This review serves as a roadmap for future research in the field.

Lastly, the article in [15] introduces a technique for animating written text in HamNoSys, a lexical sign language notation, into signed posture sequences. It utilizes transformer encoders to generate posture predictions, considering spatial and temporal information. Weak supervision is employed for training, and a novel distance measurement is introduced to assess the quality of produced posture sequences. The project aims to reduce communication barriers between hearing and hearing-impaired individuals.

III. METHODS AND MATERIAL

Long-Short Term Memory - SLD Model

This work proposes a unique LSTM-based method for sign language detection. LSTM is an advanced form of RNN. While RNN is basic method for sequence detection it has many flaws such as it short storage which is addressed by lstm model (as shown in fig-2.1). The project aims to analyse movies of sign language and successfully identify the signs using LSTM.

From fig-2.1 we can say that lstm is the same as rnn but with a added long team memory to the mix.Wherernn fails to have memory which can store multiple important key strings lstm succeeds. In recent years lstm have achieved notable success in a number of sequence modelling challenges. We aim to ride the trend and suggest a new model for sign language detection rather than the conventional method such as cnn.

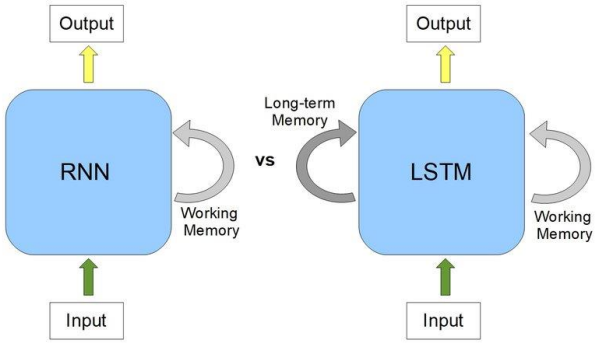

Fig 2.1 LSTM vs RNN

2.2 LSTM

Recent developments in machine learning algorithms, particularly deep learning, have demonstrated considerable promise in the recognition of sign language. The detection and recognition of sign language motions has been accomplished using deep learning models like convolutional neural networks (CNNs) and recurrent neural networks (RNNs). Long Short-Term Memory (LSTM) among RNNs has been demonstrated to to handle long-term dependencies and model sequential data very well.

The importance of this study rests in its innovative use of LSTM to recognize sign language. The suggested method entails extracting characteristics from sign language films before successfully classifying sign language motions using an LSTM-based classification model. The community of the deaf and hard of hearing will be greatly impacted by the suggested approach's potential to increase the precision and effectiveness of sign language recognition.

Accurate understanding of sign language gestures can help hearing-impaired and deaf people communicate with the general public. Additionally, it can make it easier for people without hearing loss to access possibilities like school, job, and others also sign to improve accessibility and diversity, sign language

recognition can be included into a variety of products and programmers, including virtual assistants, learning resources, and communication devices. Overall, the proposed method has the potential to significantly advance the field of sign language understanding and raise the standard of living for those with hearing impairments.

In this study, we analyse movies of sign language and successfully identify the signs using LSTM.The prepossessed sign language movies are then used to extract features for an LSTM-based classification model in our suggested method. On a data set of sign language that is freely available, we assess the performance of our method and contrast it with current state-of-the-art techniques. The outcomes demonstrate that our method performs better than current approaches and achieves high accuracy in sign language detection.

Why LSTM over CNN?

There is a common misunderstanding that SLD is image recognition but that is not entirely true and the reason for this misconception is, when most people test sigh language they mainly use ASL (American sign language) in which alphabets for ASL are stationary hand signs but in real word we don't use alphabets but use action to convey our will. So we can conclude that SLD is more about action recognition rather than image recognition.

Since the object is not to recognize an image but the action performed, CNN will not be the best option for this project. As in CNN key features are obtained by applying multiple filters to the image and using them to make a future prediction. Which can work for sign languages which has constant sign which doesn't have much motion in them such as ASL. But real time sign languages uses our hand movement to recognize action. Now let's take action such as basic thank you and hello which have same hand pose but different monument as shown in below. So in CNN when we train it takes one frame from the action and work on it. But her for CNN both thank you and hello are same frame wise if we do not consider previous frames together. This problem can be overcome by if we use sequence detection models such as LSTM.

Fig 2.2.1

Fig 2.2.2

LSTM other application examples

On the basis of previous information, LSTM may be employed to foresee stock prices. Take into consideration for one, a situation in which we wish to forecast the closing price of a specific stock using its prior values. Using a set of data with characteristics like prior close prices, trade volume, along with other pertinent indications, we may develop an LSTM model. To create forecasting, the machine learning algorithm understands about the connections and trends in the data. The LSTM approach develops how to identify dependence over time, including the impact of market patterns, headlines, and investor reactions on the price of a stock, by being fed past data. The data in sequence is processed by the model while taking into consideration both the connection between previous

and present prices. Following practise, utilizing the given past data, the algorithm may be used to forecast the final price on an upcoming day.

In the area of natural language interpreting, LSTM is frequently employed for analyzing feelings. Let's look at an instance where we wish to categories evaluations of movies as favourable or unfavourable. Given an array of labeled film evaluations, whereby each assessment is classified as either good or bad, we may develop a model with LSTM. The model developed by LSTM analyses the text information's linear structure while taking the sequence of words or phrase interdependence into account. It gains the ability to identify linguistic patterns and link them to emotions. The LSTM model can comprehend the emotions conveyed in a text after being trained on a large amount of data.

We provide the algorithm an annotated evaluation in order to generate projections, and it categorizes the feeling as either favourable or adverse. Programmers include online sentiment assessment, client input evaluation, and reviewer compilation will all profit from it. The LSTM algorithm may be used for identification of handwriting, for example, to translate written words into electronic representation. If we have an archive of handwritten letters and their related labels, let's imagine this. Utilizing this collection of data, we may develop a model with LSTM to identify characters that are handwritten.

The series of strokes used to write a word are the input for the model called LSTM. It gains the ability to recognize temporal relationships in the strikes and link them to the appropriate letters. The machine learning algorithm becomes resilient in identifying multiple writing structure after training on a wide data featuring various forms of writing.

The LSTM approach estimates the description for each new printed text we input while using the algorithm for identification. Apps like handmade note digesting, identity authentication, and automated forms filling can all benefit from this.

Algorithms for recognizing words frequently employ the LSTM algorithm Let's look at an instance where we wish to translate spoken text into text. A collection of spoken words and their copies can be used to train an LSTM model. The LSTM approach analyses the audio input sequentially, gathering grammatical trends and language fluctuations. In order to accurately trans-crib language elements, it learned to connect auditory aspects with them. The algorithm can recognize a variety of linguistic characteristics after being trained on a big and varied data.We send the sound clip to a the LSTM technique approach, which then forecasts the associated text, to transcribing speech. Companies include assistants that speak, interpretation offerings, and gadgets that can be spoken can all benefit from this.

Machine translation tasks, in which we attempt to translate phrases by one tongue to a different one, benefit from the use of LSTM. Take into account a case where we wish to translate sentences. Given an array of comparable phrases in English and French, we may develop a model based on LSTM. The linear aspect of the language's syntax is processed by this model, which also captures long-range relationships and contextual data. It gains the ability to properly translate phrases and terms by aligning them in the source and destination tongues. Speech translation skills are improved in the model by instruction using a sizable bilingual corpus.

IV. The Proposed System

By using our camera vision of our computer we take test and train data for our model. Here our data is a group of pictures and 30-pic together we get for a single action for recognition. By using medapipe we first detect the key-points on each frame. And the key-points are saved as a sequence in a single array.

Afterwords we can pass the array to a lstm layer which is connected to a deep layer. After training we save the weights of the model in h5 file. Finally we can use an inter-pace that will collect real-time data and convert them in to single numpy arrays. Using lstmnn we can find and display an action which is there in given data set.

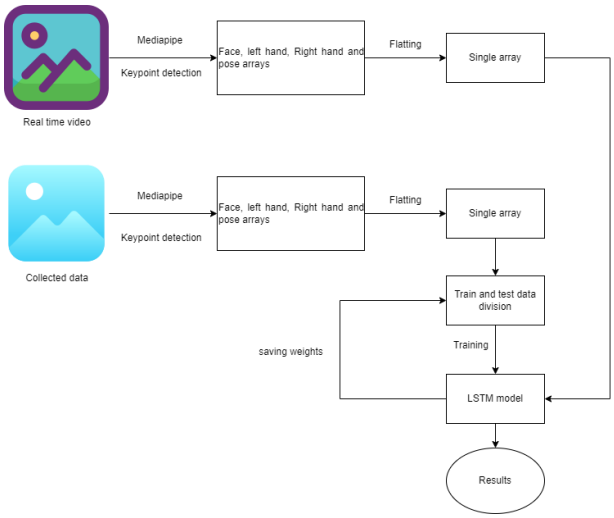

Fig - 3.1 design model

Let's try to understand the enter work flow of the project with a sample example. The following steps are followed by the code for collecting pictures and using them to using them for prediction: At start after we install all the libraries from CMD we can start running the code snippets which are above. First import the library's we need, we import mediapipe as well which is the most important part of our project and gives a new path for action detection. The below figure shows images in which mediapipe was applied.

Fig - 3.2 mideapipe

- Lets first see how mediapipe works, when we take give picture to mediapipe solutions we get an array in which has some coordinates to it. Mediapipe has manly four types of array which is Face, Right hand, Left hand, pose. Each of this array gives us coordinates of their respective of respective part.
- In mediapipe our hands are represented by 21 key-points each having three values,x (i.e. x axis value), y (i.e. y axis value) and z (i.e. deapth value in picture). Same for the face but it has 468 key-points and three values. In pose we have 33 key-points but have four values.
- Now that we got four numpy arrays named face, lh, rh, pose. We flatten each array and concatenate into 1-D array. This is the 1-D array is the key-points of the picture which we will save as .numpy file.
- For each action we take 30 picture and 30 sample each. So we get 900 .numpy files for an action.
- For the first part of the design we need to collect the picture for our data set. In code a loop has been created so that when run the next 30 frames will be recorded and saved as action we need.
- After creating the respective directory for saving action we can start traning. Each picture that has been recorded will have mediapipe solutions applied and saved.
- Using each numpy array as a sequence we can pass it to lstm model which contains 3-lstm layer and 3 fully connected layers. And the weights are adjusted to be saved in .h5 file.

To run in real time model, we will load .h5 file and take real time image from CV, This image is converted into numpy array using the same method as data set. We will compare them with our original training data set we get the one close to our action and will be displayed.

V. RESULTS AND DISCUSSION

Epoch 1/480
8/8 [==============================] - 5s 128ms/
categorical_accuracy: 0.1535
Epoch 2/480
8/8 [==============================] - 1s 127ms/
categorical_accuracy: 0.0789
Epoch 3/480
8/8 [==============================] - 1s 113ms/
categorical_accuracy: 0.2412
Epoch 4/480
8/8 [==============================] - 1s 115ms/
categorical_accuracy: 0.2061
Epoch 5/480
8/8 [==============================] - 1s 122ms/
categorical_accuracy: 0.3377
Epoch 6/480
8/8 [==============================] - 1s 111ms/
categorical_accuracy: 0.3640
Epoch 7/480
8/8 [==============================] - 1s 114ms/
categorical_accuracy: 0.4167
Epoch 8/480
...
Epoch 479/480
8/8 [==============================] - 1s 122ms/st
categorical_accuracy: 1.0000
Epoch 480/480

A. Figures and Tables

Fig 4.1: The Model Training

The above section is generated after running the module for the first time. During this process, we split the data using sklearn, and weights and predictions are made. The number of epochs to be taken is recommended to be a large number, depending on the data. We generally stop training when the accuracy exceeds 90% for a significant amount of data, as continuing beyond this point may lead to overfitting and a loss of accuracy.

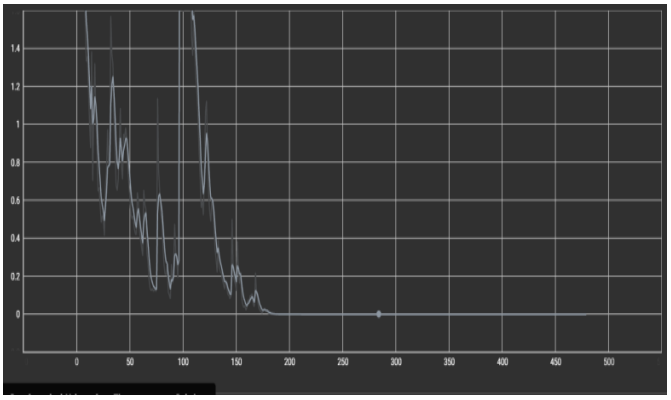

Fig 4.2 Epoch Loss

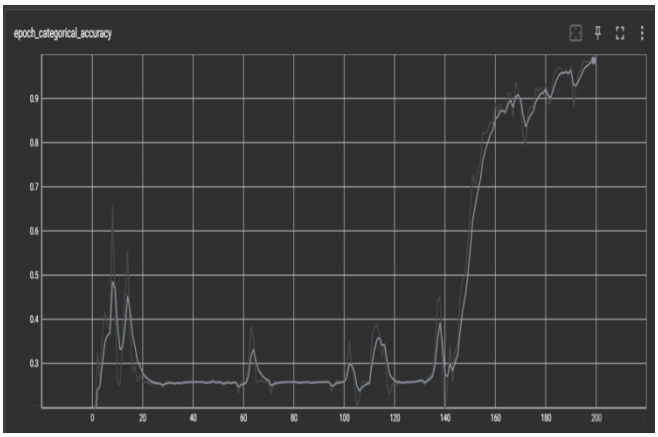

Fig 4.3 Epoch Categorical Accuracy

Model: "sequential_4"

Layer (type)	Output Shape	Param #
lstm_12 (LSTM)	(None, 30, 64)	442112
lstm_13 (LSTM)	(None, 30, 128)	98816
lstm_14 (LSTM)	(None, 64)	49408
dense_12 (Dense)	(None, 64)	4160
dense_13 (Dense)	(None, 32)	2080
dense_14 (Dense)	(None, 8)	264

Total params: 596,840
Trainable params: 596,840
Non-trainable params: 0

Fig 4.4: The Model Parameters

In our training model, we observe that we start with low accuracy in the first hundred steps or so, but at around step 140, there is a decrease in loss and an

increase in accuracy. By step 200, we achieve almost 100% accuracy with negligible loss. Our model uses three LSTM layers and three dense layers. From our data sample, we have identified and trained a total of 597,840 parameters. The number of samples and actions determine the number of parameters and remains unaffected by other factors, as the MediaPipe library counts the same number of key-points from any type of sample. The typical *ANN parameters decrease from layer to layer, as shown in Fig 4.3.

Comparison Table Showing Different Techniques:

Ref no.	Research paper	Methodology	Accuracy
16	Pansare et al. (2012)	Euclidean distance	90.19%
08	Sharma et al. (2020)	Multi layer Perceptron SVM (MLP)	96.96%
08	Sharma et al. (2020)	K-NN	96.15%
09	Aloysius and Geetha (2020)	CNN	91.76%
13	Nguen et al. (2019)	ResNet-based CNN	94.1%
-	-	LSTM	98.86%

Finally, this effort suggests a novel LSTM-based approach for sign language recognition. The proposed method first derives features from sign language gestures, which are then effectively classified using an LSTM-based classification model. The suggested approach has the potential to make major strides in the comprehension of sign language as well as improve the quality of life for those who have hearing loss. Accurate comprehension of sign language gestures may facilitate communication between hearing-impaired and deaf persons and the general public, as well as making it simpler for those who do not have hearing loss to make use of opportunities like education and employment. To increase accessibility and multiculturalism, identification of sign languages may also be included into a range of goods and initiatives, such as virtual aids, learning materials, and tools for interaction. In general, the suggested technique is an important advancement in the field of sign language recognition since it is inexpensive, reliable, and time-effective as feasible.

VI. CONCLUSION

At last this effort suggests a novel LSTM-based approach for sign language recognition. The proposed method first derives features from sign language gestures, which are then effectively classified using an LSTM-based classification model. The suggested approach has the potential to make major strides in the comprehension of sign language as well as improve the quality of life for those who have hearing loss. Accurate comprehension of sign language gestures may facilitate communication between hearing-impaired and deaf persons and the general public, as well as making it simpler for those who do not have hearing loss to access opportunities like education and employment. A computer program may also recognize signs in sign language Virtual aids, educational materials, and communication tools are just a few of the goods and initiatives that have been developed to increase accessibility and diversity. Overall, the suggested approach is affordable, precise, and as quick as feasible, which makes it an important advancement in the field of sign language recognition.

VII. REFERENCES

[1]. Rastgoo, R., Kiani, K., & Escalera, S. (2021). Sign language recognition: A deep survey. Expert Systems with Applications, 164, 113794.

[2]. Cooper, H., Holt, B., & Bowden, R. (2011). Sign language recognition. Visual Analysis of Humans: Looking at People, 539-562.

[3]. Von Agris, U., Zieren, J., Canzler, U., Bauer, B., & Kraiss, K. F. (2008). Recent developments in visual sign language recognition. Universal Access in the Information Society, 6, 323-362.

[4]. Nair, A. V., & Bindu, V. (2013). A review on Indian sign language recognition. International journal of computer applications, 73(22).

[5]. Yang, S., & Zhu, Q. (2017, July). Continuous Chinese sign language recognition with CNN-LSTM. In Ninth international conference on digital image processing (ICDIP 2017) (Vol. 10420, pp. 83-89). SPIE.

[6]. Bantupalli, K., & Xie, Y. (2018, December). American sign language recognition using deep learning and computer vision. In 2018 IEEE International Conference on Big Data (Big Data) (pp. 4896-4899). IEEE.

[7]. Elhagry, A., & Elrayes, R. G. (2021). Egyptian sign language recognition using CNN and LSTM. arXiv preprint arXiv:2107.13647.

[8]. Sharma, S., Gupta, R., & Kumar, A. (2020). Trbaggboost: An ensemble-based transfer learning method applied to Indian sign language recognition. Journal of Ambient Intelligence and Humanized Computing, 2018

[9]. Aparna, C., & Geetha, M. (2020). CNN and stacked LSTM model for Indian sign language recognition. In Machine Learning and Metaheuristics Algorithms, and Applications: First Symposium, SoMMA 2019, Trivandrum, India, December 18–21, 2019, Revised Selected Papers 1 (pp. 126-134). Springer Singapore.

[10]. Hillis, M. E., Aubrey, B., Blanchet, J., Shao, Q., Zhou, X., Balkcom, D., & Kraemer, D. J. (2022). Overlapping semantic representations of sign and speech in novice sign language learners. In Proceedings of the Annual Meeting of the Cognitive Science Society (Vol. 44, No. 44).

[11]. Kozyra, K., Trzyniec, K., Popardowski, E., & Stachurska, M. (2022). Application for Recognizing Sign Language Gestures Based on an Artificial Neural Network. Sensors, 22(24), 9864.

[12]. Singh, J., & Singh, D. (2022, October). A Comprehensive Review on Sign Language Recognition Using Machine Learning. In 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions)(ICRITO) (pp. 1-6). IEEE.

[13]. Nguen, N. T., Sako, S., & Kwolek, B. (2019). Deep CNN-based recognition of JSL finger spelling. In C. L. M. Q. P. H. C. R. E. Perez Garcia H. Sanchez Gonzalez L. (Ed.). Lecture notes in computer science. Springer International Publishing (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics): Vol. 11734 LNAI.

[14]. Chen, F. S., Fu, C. M., & Huang, C. L. (2003). Hand gesture recognition using a real-time tracking method and hidden Markov models. Image and vision computing, 21(8), 745-758.

[15]. Zakariah, M., Alotaibi, Y. A., Koundal, D., Guo, Y., & Mamun Elahi, M. (2022). Sign language recognition for Arabic alphabets using transfer learning technique. Computational Intelligence and Neuroscience, 2022.

[16]. Adeyanju, I. A., Bello, O. O., & Adegboye, M. A. (2021). Machine learning methods for sign language recognition: A critical review and analysis. Intelligent Systems with Applications, 12, 200056.

[17]. Shalev-Arkushin, R., Moryossef, A., & Fried, O. (2022). Ham2Pose: Animating Sign Language Notation into Pose Sequences. arXiv preprint arXiv:2211.13613.

[18]. Pansare, J. R., Gawande, S. H., & Ingle, M. (2012). Real-time static hand gesture recognition for American sign language (ASL) in complex background. Journal of Signal and Information Processing, 03(03), 364–367.

Cite this article as :

Azees Abdul, Adithya Valapa, Abdul Kayom Md Khairuzzaman, "Enhancing Accessibility with LSTM-Based Sign Language Detection", International Journal of Scientific Research in Computer Science, Engineering and Information Technology (IJSRCSEIT), ISSN : 2456-3307, Volume 9, Issue 5, pp.130-139, September-October-2023. Available at doi : https://doi.org/10.32628/CSEIT2390517
Journal URL : https://ijsrcseit.com/CSEIT2390517