ISSN (Online) 2395-2717

International Journal of Engineering Research in Electrical and Electronic Engineering (IJEREEE)

Vol 11, Issue 3, March 2024

A Deep Learning Based Human Computer Interface for Sign Language Recognition

[1] Arashta Hussain, [2] Nimakhi Saikia, [3] Chandana Dev

[1] [2] [3] Jorhat Institute of Science and Technology, Jorhat, Assam, India
Corresponding Author Email: [1] arashta10@gmail.com, [2] nimakhisaikia182@gmail.com, [3] chandanajist@gmail.com

Abstract*— Sign language is a major way of communication for people with hearing and speech impairments. It is very helpful for them in order to communicate with others and themselves. Hence, the necessity to develop a human computer interface for sign language recognition has gain immense popularity in recent times. There are numerous sign languages that are used throughout the world, the most common one being American Sign Language (ASL). Neural network systems may be used to tackle a wide range of problems in the subject of deep learning. This research work aims to design a real-time American Sign Language recognition system using computer vision and deep learning techniques with user built dataset. The built system uses Gaussian blur filter and a Convolutional Neural Network (CNN) classifier. Keras is used to train the dataset used in the model. The built model in the proposed work uses about 600 images for each of the 26 alphabet. The proposed system converts the hand gestures of the ASL fingerspelling alphabets into English text alphabets, alphabets to words and then words to a complete sentence. The accuracy that the model is able to achieve is approximately 99.4%. Thus, the result achieved from this system infers that the same could be helpful to improve the quality of life for deaf and speech impaired people.

Index Terms*— American Sign Language, Recognition, CNN, Keras.

I. INTRODUCTION

According to World Health Organisation (WHO) data, 360 million people, including 32 million children and 328 million adults, have hearing impairment, which represents more than 5% of the global population. By 2050, the WHO predicted that the population would have doubled, totalling 900 million people [1]. The Census of 2011 mentions that out of the population of disabled people of 2.68 crores (2.21% of the total population) in India, there are around 1027835 people, including 545179 males and 482656 females, who suffer from hearing and speech impairment. This group of people has developed their own language to communicate among themselves, known to us as sign language [1]. Human body language and gestures are used in sign languages to convey ideas visually. It varies from region to region. In order to solve the communication gap experienced by those with hearing problems, a sign language recognition system has been implemented. Sign Language Recognition (SLR) is a computer vision task that involves recognizing and translating sign languages into written or spoken language [2]. SLR receives a lot of attention at present as it can reduce the frustration between the hand- talk community and leads to effective communication[3]. The SLR involves data collection, pre-processing, feature extraction and classification phase [3]. In this fast growing demographic, the communication barrier that adversely affects the social interactions and quality of life for deaf-mute individuals must be removed.

Fig.1: Sign Language Recognition System

SLR has two main ways of identifying things: static and dynamic. Static recognition looks for hand movements that represent letters and numbers, while dynamic recognition investigates a sequence of hand motions properties [4]. SLR is typically designed using two techniques: vision-based and glove-based. Vision-based methods use images or videos to identify signs, while glove-based methods use data gloves to capture hand movements. Each method has its own cost, accuracy, and required equipment [2, 5].

Sign language is a visual language and consists of 3 major components [6]:

Table I: Major Components of Sign Language

Fingerspelling	**Word level sign Vocabulary**	**Non-manual features**
Used to spell words letter by letter.	Used for the majority of communication.	Facial expressions and body gestures.

The aim of this work is to create a deep learning system capable of recognizing sign language and converting hand gestures into phrases. Initially, a user-built dataset with 26 English alphabets was employed to achieve this goal. The Python 3.7 software tool was utilized to implement the system. Deep learning-based techniques need datasets to be

ISSN (Online) 2395-2717

International Journal of Engineering Research in Electrical and Electronic Engineering (IJEREEE)

Vol 11, Issue 3, March 2024

trained and tested. Utilizing the Tensor Flow 2.11 module has satisfied this criteria. The goal is to use deep learning techniques to create a reliable and adaptable sign language recognition system that can quickly and reliably process a wide range of motions.

II. LITERATURE REVIEW

Sign language recognition is a topic which has been addressed multiple times and is not new. Over the last few years, different classifiers have been applied to solve this problem including linear classifiers, neural networks and Bayesian networks. Linear models are easy to work with, but require complex feature extraction for increased accuracy.

Kakoty et al. provide a real-time translation of the Indian and American Sign Language alphabet and numbers based on hand kinematic assessment with an accuracy of 96.7%. The finger and wrist joint angles were obtained using an indigenously developed data glove. A radial-based function kernel Support Vector Machine (SVM) with 10-x cross-validation was utilized for recognition [2].

Sahoo used machine learning to learn Indian Sign Language (ISL). They used static hand motions that corresponded to the numbers 0 to 9 to train their model. A collection of 500 images was produced using a digital RGB sensor, one image for every digit. They used supervised learning techniques, such as Naive Bayes, to train their models. The average accuracy was 98.36%, and the slightly higher rate was 97.79% [5].

Keskin et al. used object identification using components to identify ASL numerals. Their dataset included 30,000 samples divided into ten (10) classes. Keskin et al. focused on using object identification based on components to recognize American Sign Language (ASL) numerals [7].

Zhou ren et al. 2011, developed a robust model to recognize sign language by using a kinect sensor. To measure the distance of hand dissimilarity, they used a metric called Finger Earth Mover's Distance (FEMD) [8].

Sundar b. et al. came up with a way to recognize ASL alphabets using Mediapipe. They were able to do it with 99% accuracy using (long-short time memory network) LSTM, which is a way of recognizing hand gestures. It's really useful for Human-Computer Interaction (HCI), since it can turn gestures into text [9].

Kai li et al. 2018, presented an HCI which uses recognize depth-sensing camera to recognize sign language. It predicts the hand gesture using the kalman filter and depth informationn from the kinect, resulting in a smooth and reliable tracking system [10].

Das et al. 2018, figured out a way to use deep learning to recognize ASL by processing static images of motion. They trained a CNN on the dataset using the Inception V3 algorithm, and it was able to recognize ASL with an accuracy of over 90%. They also found that the dataset had 24 classes that represented alphabets, from A to Z, with a best-case scenario of 98%. They also used a correctly cropped image dataset to prove that Inception V3 was good enough for ASL recognition [11]

Ansari et al. 2016, utilised photos with 3D depth information to construct a model that can categorise static motions in ISL. A dataset of 5041 hand motion photographs was produced using Microsoft Kinect, which was utilised to collect both 2D and 3D images. These photos were grouped in 140 categories. The model developed using K-meaning clustering algorithm was able to recognise 16 English alphabets what was able to attain 90.8% average accuracy [12].

Rekha et al. 2011, did a study using a bunch of ISL signs, including 23 static ones and 3 dynamic ones. They used color segmentation to figure out which ones were hands. They then trained the SVM with features like edge orientation and texture, and it was able to detect hands 86.3% of the time. Unfortunately, it wasn't able to do it in real time because it wasn't fast enough [13].

Jyotishman Bora et al. 2023, designed an advanced machine learning model capable of comprehending Assamese Sign Language. This model was trained by combining two-dimensional and three-dimensional images and the MediaPipe hand tracking solution, which enabled it to accurately identify Assamese gestures with 99% accuracy. The MediaPipe system, which is lightweight and adaptable to a number of devices, offers a high degree of precision in hand tracking and classification without sacrificing speed or accuracy [14].

Bhuyan et al. 2011, employed a skin color segmentation approach to identify hands, which was supplemented by the use of the nearest neighbor classifier. Utilizing a dataset of 400 photographs with eight ISL movements, the accuracy of the results was estimated to be more than 90% [15].

Pugeault et al. 2011, used a massive collection of 48,000 3D-depth pictures collected with a Kinect sensor to construct a real-time Alphabet recognition system for ASL. They used special filters and a random forest to get really precise classification rates [16]

Kinjal et al. 2021 proposed the first Indian Sign Language video dataset, INSIGNVID. Using this dataset as input, a unique approach is presented that uses transfer learning to translate video of ISL sentences into acceptable English sentences. Using MobilNetV2 as the pretrained model, the suggested method produced encouraging results on the dataset[17].

Akash et al. 2023 suggested an approach that translates hand motions into suitable text by using computer vision and deep learning techniques. A mixture of data pre-processing, labeling, feature generation, key point detection using MediaPipe, and LSTM neural network training was used in the construction of the system[18].

Karthika et al. 2021 proposed a system that converts speech and text from American Sign Language (ASL). In order to detect the ASL hand movements, this work extracts

ISSN (Online) 2395-2717

International Journal of Engineering Research in Electrical and Electronic Engineering (IJEREEE)

Vol 11, Issue 3, March 2024

efficient hand features using convolutional neural networks (CNN). An accuracy of 88% is provided by the suggested model[19].

Tanya et al. 2023 proposed a model based in ASL fingerprint-based real-time method utilizing Convolutional Neural Networks (CNN). It is implemented using CNN as it is very effective at addressing computer vision issues. This proposed model achieved 99% accuracy on the first (MNIST) dataset and 96% on the second (ASL) dataset[20].

Aruna et al. 2022 designed a CNN model in order to classify the different photos in this project according to their alphabetical equivalent and used to identify the signs in the American Sign Language[21].

Victoria a. Adewale 2018 goes through several stages, including text-to-speech (TTS) conversion, picture segmentation, feature detection and extraction from ROI, supervised and unsupervised image classification using K-Nearest Neighbor (KNN) algorithms, and data collection utilizing the KINECT sensor to design a system that recognizes ASL[22].

III. CONVOLUTIONAL NEURAL NETWORK

A Convolutional Neural Network (CNN) is a Deep Learning neural network architecture used in Computer Vision, a field of Artificial Intelligence that interprets image or visual data. It is made up of several layers that use gradient descent and backpropagation to train the best filters, including an input layer, a convolutional layer, a pooling layer, and a fully connected layer[23].

The primary component of a CNN is the convolutional layer, where the majority of calculations are carried out. It is a layer for feature extraction that uses filters to extract local features, creates a feature map using a convolution kernel function, and outputs the result to the pooling layer.

Sentences or documents formatted in a matrix are fed into CNN. Every row in the matrix represents a single token, which is usually a word but can also be a character. In other words, a word is represented by a vector in each row. The most typical kind of these vectors are word embeddings, or low-dimensional representations, such as word2vec or GloVe; but, they can also be one-hot vectors that index the word into a lexicon. When it comes to visual identification tasks, convolutional neural networks are great. Prominent computer vision networks such as VGG, LeNet, AlexNet, Inception, and ResNet are available. Usually, image recognition is the purpose of artificial neural networks. In order to extract the outline properties from images, these networks are composed of many neural networks. These networks need a significant amount of memory and processing power to do this [24].

However in our proposed CNN model we have used two convolutional layers.

IV. EXECUTION

A. Tools Used

We have made use of Pycharm Environment (a free and open source web application) for performing the experiments. It facilitates the creation and exchange of documents with text, mathematics, code, and graphics. It also covers statistical modeling, data processing, cleansing, and machine learning, among other things. The CNN models have been developed using Python package Keras (a deep learning library). We have also used the OpenCV platform for building the dataset.

B. Workflow

The three stages of our proposed system's workflow—data preparation and collection, training, testing and output display of the system—are depicted in Fig.2

Below is a quick explanation of each of its stages.

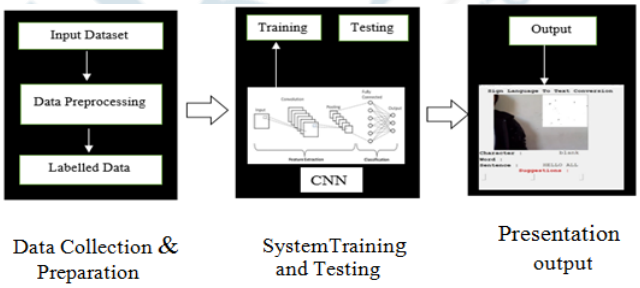

Fig.2: Workflow of the system

Data Collection and Creation:

In this stage, user build dataset was created in jpg format. It consists of American Sign Language alphabets from A-Z with 400 pictures for training and 200 for testing. Each frame of hand gesture was recorded with a designated Region of Interest. Several characteristics are extracted from the collected image by using the Gaussian Blur Filter, resulting in an image with the following appearance. A rich set of features is produced by preprocessing the gathered data to eliminate unnecessary information.

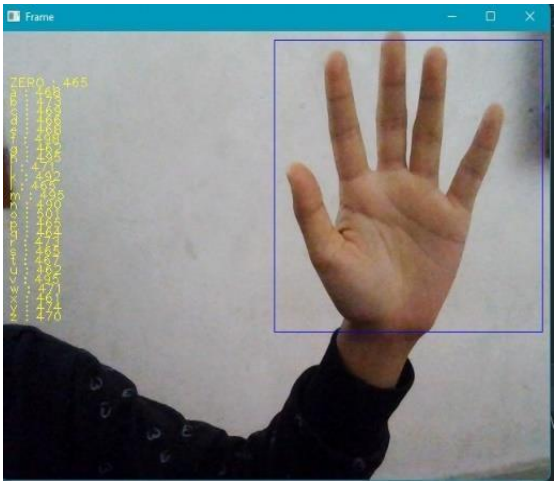

Fig.3: ROI Frame

ISSN (Online) 2395-2717

International Journal of Engineering Research in Electrical and Electronic Engineering (IJEREEE)

Vol 11, Issue 3, March 2024

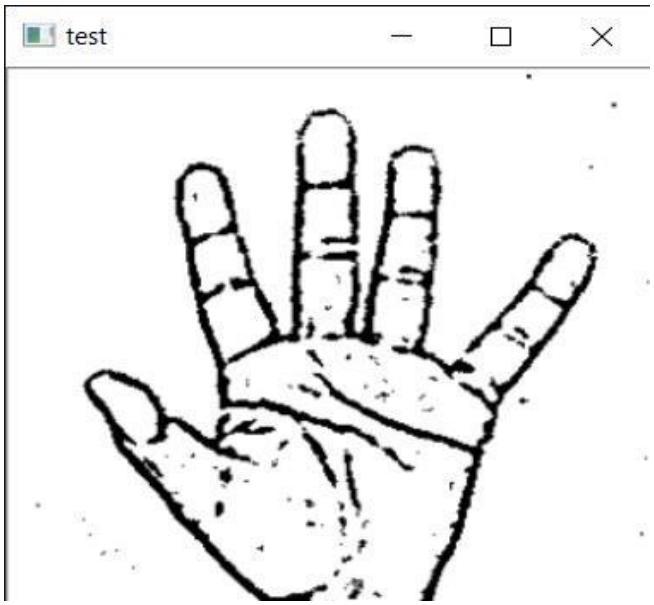

Fig.4: Image After Gaussian Blur Filter

Training and Testing:

The input, convolution, global max pool, and fully connected layers make up the four fundamental layers of the CNN model, which are briefly described below:

a) Input Layer: The input layer in a CNN typically contains an image or sequence of images with general dimensions of 32x32, 32x32, and 3x3.

b) Convolution Layer: The layer extracts features from the input dataset using learnable filters called kernels. These smaller matrices are generally of 2x2, 3x3, or 5x5 in shape. The dot product between kernel weight and image patch is computed, resulting in feature maps.

c) Pooling Layer: A pool size of (2, 2) is used and Max pooling is applied to the input picture. This lowers the number of parameters, which lowers the cost of computing and reduces overfitting.

d) Fully Connected Layer: It takes the input from the previous layer and computes the final classification or regression task.

Presentation of output:

The output results are displayed through the creation of line charts in Fig.5, which facilitate the comparison of two CNN models. The next section contains the various parameter configurations for carrying out experiments.

V. EXPERIMENTAL SETUP

Two CNN models have been constructed in this study, with multiple parameters for each layer. The CNN model's parameter choices, which include batch size, activation function, train test splitting ratio, loss function, optimizer, dropout etc are specified in table III.

Table II: Hyper parameters used in the proposed model

No of epochs	10
Batch size	128
Activation Function	ReLU
Train test splitting ratio	80:20
Loss function	Categorical Cross entropy
Optimizer	Adam
Dropout	0.4

One or two convolution layers and 32 number of filters have been used in the experiment. Additionally, the studies have been carried out using other filter sizes, including 3 × 3, 4 × 4 etc. These parameters values were determined by examining the research that other writers in this field had done [25].

Along with other variables like the quantity and size of filters, each model contains a different number of convolutional layers. Table 4 describes the configuration parameters for each of the 2 CNN models. The outcomes of our model's experimentation with various CNN parameter choices are examined in the following table 4:

Table III: Configuration Settings of 2 CNN Model

Model Name	Convolution layers	Hidden layers	No. of filters	Filter size
CNN1	2	13	32	3,4
CNN2	2	11	32	3,3

VI. RESULT AND DISCUSSION

The validation accuracy and loss score of all CNN models are listed in Table 5 along with its training time (in seconds).

Table IV: Accuracy and Loss Score

Model Name	**Validation accuracy**	**Validation loss**	**Training time (s)**
CNN1	91	0.23	112
CNN2	99.4	0.028	266

Figure 5 shows the average validation accuracy and loss score of the proposed CNN model with 99.4% accuracy. In Fig. 5a, b, the X-axis indicates the quantity of training iterations, while the Y-axis indicates the percentage of accuracy and loss score, respectively. Models are learning from data, as seen by the average learning curve in Fig. 5(a), which demonstrates a progressive improvement in the accuracy percentage with an increase in training. The total number of mistakes that the model anticipated is known as the loss score. As the number of training steps grew, Figure 5(b) illustrates that the mistakes dropped from the beginning.

ISSN (Online) 2395-2717

International Journal of Engineering Research in Electrical and Electronic Engineering (IJEREEE)

Vol 11, Issue 3, March 2024

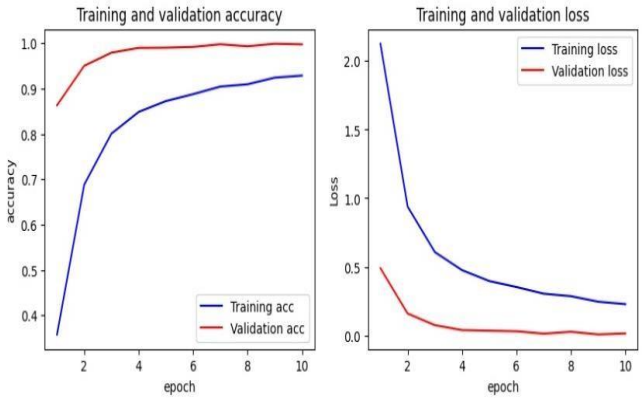

Fig.5: a) Training and validation accuracy and b) Training and validation loss

The other performance parameters such as precision, recall, F-measure, for the suggested CNN model is specified in the table 6.

Table VI: Precision, Recall and F score of the CNN model

Classes	Precision	Recall	F1- Score
A	0.99	0.99	0.99
B	1.00	1.00	1.00
C	1.00	1.00	1.00
D	1.00	1.00	0.99
E	0.98	0.99	0.98
F	0.99	0.99	0.99
G	0.99	0.99	0.99
H	0.99	0.99	0.99
I	1.00	1.00	1.00
J	1.00	1.00	1.00
K	1.00	0.99	0.99
L	1.00	1.00	1.00
M	0.99	0.99	0.99
N	0.98	0.98	0.98
O	1.00	1.00	1.00
P	0.99	0.99	0.99
Q	0.99	0.99	0.99
R	0.99	0.99	0.99
S	0.98	0.98	0.98
T	0.98	0.98	0.98
U	0.99	0.99	0.99
V	0.99	0.99	0.99
W	0.99	0.99	0.99
X	1.00	1.00	1.00
Y	0.99	0.99	0.99
Z	0.99	0.99	0.99
ACCURACY			0.99
MACRO AVG	0.99	0.99	0.99
WEIGHTED AVG	0.99	0.99	0.99

Tables VII offer a comparison between the results obtained in this work and the works in the literature.

Table VII: Comparative analysis of this work and other image based approaches

Dataset	Approach	Accuracy
American Sign Language[26]	CNN	90%
American Sign Language[16]	SVM	99%
American Sign Language[27]	SVM	75%
American Sign Language[28]	CNN	88%
American Sign Language[This work]	CNN	99.4%

VII. CONCLUSION AND FUTURE ASPECTS

This research work highlights the potential of technology-based solutions to enhance communication and accessibility for the deaf and hard-of-hearing community. Future work may include improving sign language recognition systems, particularly for Indian Sign Language, by developing new machine learning and deep learning algorithms and improving data capture and annotation procedures. It may include recognizing gestures with both hands and using common words signs to simplify sentence formation.

REFERENCES

[1] S, D., K B, K. H., M, A., M, S., S, D., & V, K. (2021). An efficient approach for interpretation of indian sign language using machine learning. 2021 3rd International Conference on Signal Processing and Communication (ICPSC), 130–133. https://doi.org/10.1109/ICSPC51351.2021.9451692

[2] Kakoty, N. M., & Sharma, M. D. (2018). Recognition of sign language alphabets and numbers based on hand kinematics using a data glove. Procedia Computer Science,133, 55–62. https://doi.org/10.1016/j.procs.2018.07.008

[3] Madhiarasan, M., & Roy, P.P. (2022). A Comprehensive Review of Sign Language Recognition: Different Types, Modalities, and Datasets. ArXiv, abs/2204.03328.

[4] Pigou, L., Dieleman, S., Kindermans, P.-J., & Schrauwen, B. (2015). Sign language recognition using convolutional neural networks. In L. Agapito, M. M. Bronstein, & C. Rother (Eds.), Computer Vision—ECCV 2014 Workshops (Vol. 8925, pp. 572– 578). Springer International Publishing. https://doi.org/10.1007/978-3-319-161785_40

[5] Sahoo, A. K. (2021). Indian sign language recognition using machine learning techniques. Macromolecular Symposia, 397(1), 2000241. https://doi.org/10.1002/masy.202000241

[6] Philippe d, (2011), "research- sign language recognition" on https://wwwi6.informatik.rwthaachen.de/~dreuw/database.php.

[7] Keskin, C., Kıraç, F., Kara, Y. E., & Akarun, L. (2011). Real time hand pose estimation using depth sensors. 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops), 1228–1234. https://doi.org/10.1109/ICCVW.2011.6130391

[8] Ren, Z., Yuan, J., & Zhang, Z. (2011). Robust hand gesture recognition based on finger-earth mover's distance with a commodity depth camera. Proceedings of the 19th ACM International Conference on Multimedia, 1093–1096. https://doi.org/10.1145/2072298.2071946

[9] Sundar, B., & Bagyammal, T. (2022). American Sign Language recognition for alphabets using mediapipe and lstm. Procedia Computer Science, 215, 642–651. https://doi.org/10.1016/j.procs.2022.12.066

[10] Li, K., Cheng, J., Zhang, Q., & Liu, J. (2018). Hand gesture tracking and recognition based human-computer interaction system and its applications. 2018 IEEE International Conference on Information and Automation (ICIA), 667–672. https://doi.org/10.1109/ICInfA.2018.8812508

[11] Das, A., Gawde, S., Suratwala, K., & Kalbande, D. (2018). Sign language recognition using deep learning on custom processed static gesture images. 2018 International Conference on Smart City and Emerging Technology (ICSCET), 1–6. https://doi.org/10.1109/ICSCET.2018.8537248.

[12] Ansari, Z. A., & Harit, G. (2016). Nearest neighbour classification of Indian sign language gestures using kinect camera. Sadhana, 41(2), 161–182. https://doi.org/10.1007/s12046-015-0405-3.

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

[14] Bora, J., Dehingia, S., Boruah, A., Chetia, A. A., & Gogoi, D. (2023). Real-time assamese sign language recognition using mediapipe and deep learning. Procedia Computer Science, 218, 1384–1393. https://doi.org/10.1016/j.procs.2023.01.117.

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

[16] Pugeault, N., & Bowden, R. (2011, November). Spelling it out: Real-time ASL fingerspelling recognition. 2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops). https://doi.org/10.1109/iccvw.2011.6130290

[17] Mistree, K., Thakor, D., & Bhatt, B. (2021). Towards indian sign language sentence recognition using insignvid: Indian sign language video dataset. International Journal of Advanced Computer Science and Applications, 12(8). https://doi.org/10.14569/IJACSA.2021.0120881

[18] Kamble, A. (2023). Conversion of sign language to text. International Journal for Research in Applied Science and Engineering Technology, 11(5),1963–1968. https://doi.org/10.22214/ijraset.2023.51981

[19] Bharathi, C. U., Ragavi, G., & Karthika, K. (2021). Signtalk: Sign language to text and speech conversion. 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), 1–4. https://doi.org/10.1109/ICAECA52838.2021.9675751

[20] Kemkar, T., Rai, V., & Verma, B. (2023). Sign language to text conversion using hand gesture recognition. 2023 8th International Conference on Communication and Electronics Systems (ICCES), 1580–1587. https://doi.org/10.1109/ICCES57224.2023.10192820

[21] A. Bhat, V. Yadav, V. Dargan and Yash, "Sign Language to Text Conversion using Deep Learning," *2022 3rd International Conference for Emerging Technology (INCET)*, Belgaum, India, 2022, pp. 1-7, doi: 10.1109/INCET54531.2022.9824885.

[22] Adewale, V., & Olamiti, A. (2018). Conversion of sign language to text and speech using machine learning techniques. JOURNAL OF RESEARCH AND REVIEW IN SCIENCE, 5(1). https://doi.org/10.36108/jrrslasu/8102/50(0170)

[23] Kush K, (2023), "Introduction to Convolutional Neural Network" on https://www.geeksforgeeks.org

[24] Shirani-Mehr, H.: Applications of deep learning to sentiment analysis of movie reviews. Technical report, Stanford University (2014)

[25] Svensson, K.: Sentiment analysis with convolutional neural networks: classifying sentiment in Swedish reviews. Bachelor Dissertation, Linnaeus University, Sweden (2017)

[26] A. Das, S. Gawde, K. Suratwala and D. Kalbande. (2018) "Sign language recognition using deep learning on custom processed static gesture images," in International Conference on Smart City and Emerging Technology (ICSCET)

[27] A. Halder and A. Tayad. (2021) "Real-time vernacular sign language recognition using mediapipe and machine learning," Journal homepage: www. ijrpr. com ISSN, vol. 2582, p. 7421

[28] Bharathi, C.U., Ragavi, G., & Karthika, K. (2021). Signtalk: Sign langiage to text and speech conversion. 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), 1-4. https://doi.org/10.1109/ICAECA52838.2021.9675751.