2023 International Conference on the Confluence of Advancements in Robotics, Vision and Interdisciplinary Technology Management (IC-RVITM) | 979-8-3503-0608-8/23/$31.00 ©2023 IEEE | DOI: 10.1109/IC-RVITM60032.2023.10435143

MuteSign: Deep Learning-driven Sign Language Recognition for Inclusive communication

Savitha G[1], Chetana A T [1], Abhishek T [1], Rohith Gowda D [1], Pratiksha Soori[1], Prajwal V[2]
[1]Department of Computer Science and Engineering, RVITM, Bangalore-560076, India
[2]Department of Computer Science and Engineering, BNMIT, Bangalore-560070, India
Corresponding author. Email: savithamadhusudan@gmail.com

Abstract- Recognition of Sign Language is a typical area of experimentation, but it offers people with physical disabilities a great deal of convenience and support. Data collection, pre-processing, segmentation, feature extraction, and classification are just a few of the numerous steps that go into identifying sign language. Recurrent neural networks (RNN) with long short-term memory (LSTM) and convolutional neural networks (CNN) are two well-liked methods for recognizing sign language. CNNs are incredibly precise in their picture recognition and classification, while RNNs process sequential data extremely quickly. The significance of each frame is extracted by The Signs, which are programmed in two subunits called frames. The method involves dividing continuous signs into smaller components, referred to as frames, and deriving meaning from each frame.

Keywords—*Convolutional Neural Network, Feature Extraction, Long Short-term Memory, Recurrent Neural Network.*

1. INTRODUCTION

The goal of the Sign Language Recognition model is to enable deaf and hard of hearing persons to interact with regular people. Globally, around 6700 languages are spoken, and about 140 sign languages are recognized. But there's no official dataset for Indian Sign Language (ISL). Approximately 3 million Indians are unable to communicate with others because of this gap. Furthermore, there are two million deaf persons. Sign Language becomes widely accepted in society as individuals begin communicating with one another.

Single hand is used for American Sign Language (ASL) recognition, and two hands is used for British Sign Language (BSL) recognition. ISL is inspired by BSL. The process of translating an input sign into matching text heavily relies on recognition of sign language. While several static sign language recognition systems are available, not many have been created for continuous gesture sequences.

However, hand gestures can be incredibly useful in emergency situations, such as accidents or when medical assistance is needed. Sign Language Recognition model is implemented through the deep learning approaches, with CNN being used extensively to process visual data. Deep learning models help with recognition and classification tasks, enabling quick data processing. While films and a series of images are needed to teach dynamic gestures in static sign language identification, each frame of the video must have a name attached to it. For static sign language recognition, the model is trained with photographs, whereas dynamic motions require a sequence of pictures and videos with a label on every frame. It is possible to classify a series of frames using RNN equipped with long short-term memory (LSTM). More precise sign language recognition is made possible by the employment of both CNN and RNN for classifying various frames and feature extraction.

2. LITERATURE SURVEY

Qazi et al., [1] have implemented a Sign Language Recognition (SLR) model designed specifically for emergency situations. The dataset used by the authors consists of videos depicting eight different emergency situations, such as theft, doctor, and help, among others. Three-Dimensional Convolution Neural Network(3DCNN) along with Recurrent Neural Network(RNN) with LSTM are the two models used for classification. Additionally, the model performs Object detection through You Only Look Once (YOLO) model. The SLR system was tested with 412 videos for the eight emergency situations. The authors reported high accuracy rates for the classification models, with the 3DCNN achieving 82% accuracy and the RNN-LSTM achieving 98% accuracy. The model used for object detection, YOLO, which received a 99.6% accuracy.

Sruthi C.J et al., [2] proposed a model for recognition of continuous sequence of gestures. The dataset used in this model has gone through the CNN training and CNN testing phase. The model proposed has achieved an accuracy of 98.64%. The act of the government which is Rights of People with Disabilities Act 2016 inspired this model. This will avoid their isolation from the society to a great extent.

Anshul Mittal et al., [3] implemented and ISL model. The model is developed to overcome the limitations of isolated sign Gestures. ISL sign phrases are used to test the suggested model. To distinguish sign sentences, 35 distinct sign words are used. For signed sentences, an accuracy of 72.03% and for signed words, an accuracy of 89.55% have been reported.

Pratik Likhar et al., [4] has implemented the Sign Language Recognition system using two models. One is through semantic segmentation using normal RBG camera. CNN were chosen are used for the implementation and it uses Depth Information to get the hand regions. The models of CNN are trained using 90000 RBG and Depth images that have been generated using Kinect camera. LSTM used for Dynamic gestures. 1080 videos of 10 dynamic gestures were trained theyhave claimed that the model has achieved an F1 Scoreof 0.9959 and Intersection Over Union (IOU) score of0.9920. These scores tell us that the model performed exceptionally well in accurately segmenting the cells in the image sequence.

To create comparable applications, Kunal Chhabria et al. [5] have presented a deep learning method that translates ISL alphabets in real time together with a pipeline. The model goes through the following steps: create the dataset, train the model, and categorize the data frame by frame. The voice control drawbacks are addressed by the suggested model. While gesture detection can occur quickly, voice control has a noticeable latency.

I Convolutional Neural Network with Long Short-term Memory based Sign Language Recognition

Thanekar Aadit et al., [6] have made an application that enables mutes to communicate via video chat. It is responsible for the detection of ISL using different CNN Deep Learning algorithms. The CNN would identify the appropriate word, number, or letter in the application and send it to display.

Thanasekhar B et al., [7] implemented a dictionary for ISL through CNN model by identifying the words that are communicated. The model is composed of programming keywords that are gathered from a manual dataset with 500 photos assigned to each term. It generates a syntax similar to python.

Ming Jin Cheok et al., [8] have used dataset of Arabic Sign Language. A modified version of k-NN algorithm is used which classifies 40 signed sentences. The process has undergone in the user-dependent mode.

R. Elakkiya et al., [9] proposed the Machine Language based Sign Language Recognition. The article aims to solve three main issues in software design for life sciences (SLR) such as the selection of robust features, the handling of epenthesis movements, and the implementation of a subunit sign modelling framework. The first research is based on the complexity of hand segmentation and grouping, which involves using short sleeves and complex backgrounds. The second frontier dwells into the complexity of epenthesis movements when it comes to the classification and pruning of sign gestures. The third frontier involves the development of a novel signer adaptation framework that can recognize large vocabulary words using a combination of sign extraction and sign classification. The novel framework is suitable for enhancing the software's performance in addressing these issues.

II Sign Language Recognition using Support Vector Machine

J.L. Raheja et al., [10] have proposed an ISL using SVM. The preprocessing is done by changing the captured video into Hue, Saturation, and Value color space. Skin pixels are used for the segmentation process. Better results are claimed by the use of depth information. Then the final step of gesture classification is executed by the Support Vector Machine. MS Kinect and webcam areused for the testing process.

T Raghuveera et al., [11] have used the Microsoft Kinect Camera to develop a depth-based sign language recognition model. The model outputs both English text and speech. Via Microsoft Kinect, the system is utilized to record hand movements. An English-sentence-generating system for recognition hand gesture was created in a recent study. The Support Vector Machine (SVM) is used for training three feature classifiers to get an improved accuracy of 71.85%.

The study reported that the ensemble of three feature classifiers has made the accuracy to a significant. The model has achieved a remarkable improvement in accuracy.

Lim et al., [12] have utilized a CNN hand modelling and Hand Energy Image (HEI) for isolated Sign Language Recognition. Two Phases has been proposed: Hand tracking and Hand Representation. In the first phase, the system uses an algorithm to locate the hand in real time. In the second phase, it computes the HEI by calculating the average of segmented hand region. The input to the CNN models is provided by the HEI.

III Sign Language Recognition using Visual Geometry Group-19

R. Sreemathy et al., [13] proposed Recognition of Sign Language using Artificial Intelligence. Gradient features with an orientation towards the histogram and the Back Propagation Network (BPN) are used for training. Using the model, Real Time Signs are tested. For 5184 input features, an accuracy of 89.05% is attained. Deep learning models Google Net, VGG-16, Alex Net, and VGG-19 are employed, with attained accuracies of 95.84%, 98.42%, 99.11%, and 99.11%.

Ankita Wadhawan et al., [14] have used Microsoft Kinect Sensors and Leap Motion sensors for recognizing ISL. Hand movements and key points of fingers are collected bythe authors for some isolated sign gestures. The key points that are extracted for the testing purpose are classified using the HMM classifiers.

IV Artificial Neural Network based Sign Language Recognition

Nimisha et al., [15] has used two models for Sign Language Recognition. One uses an image-based model, and the other makes use of sensors to record the indicators. They used a camera and an Xbox Kinect sensor to compile the dataset. For a categorization strategy, ANN offers pros and cons of its own. For future extraction and classification, SVM and PCA are employed. This model has attained high accuracy.

Li et al., [16] implemented a continuous SLR model by proposing a framework through the usage of video sequence dataset. The authors have implemented two models. The First implementation uses Kinect sensors for recognizing the gestures and generating the data. To sperate the hand area from the received data they have done 3D reconstruction and affine transformation. Static hand gestures have received a 98.81% classification accuracy and dynamic gestures received a 99.08% classification accuracy.

To separate the hand area from the received data they have done 3D reconstruction and affine transformation. Static hand gestures have received a 98.81% classification accuracy and dynamic gestures received a 99.08% classification accuracy.

IV Sign Language Recognition using 3D Convolutional neural Network and Long Short-term Memory.

Okan et al., [17] has used two CNN's. One CNN is used to detect hand gestures, and another is 3D CNN to predict the text. A classification accuracy of 94.04% and 83.82% on the Ego Gesture benchmark and NVIDIA benchmark

Sanil et al., [18] predicts static gestures such as alphabets and numbers based on the sign given. The author investigated both single- and double-handed actions. The features for the Scale Invariant Feature Transform (SIFT) and Histogram of Oriented Gradients (HOG) have been manually extracted from images and generated into a dataset for training. First, imitations of the actual alphabet using one-handed movements are evaluated. When utilizing HOG for hierarchical classification, the average accuracy obtained from SIFT is 32.74%, while 52.23% is obtained from HOG.

Sridhar et al., [19] have collected a large dataset that has 15 categories of 4287 videos that consists of 263 words. An Open Pose Library-retrieved dataset is used in a proposed model for Recognition of Sign Language, wherein Bidirectional LSTM and MobileNetV2 are used for gesture classification, and features are extracted. While their dataset has an accuracy of 94.5%, the American Sign Language Lexicon Video Dataset (ASLLVD) has an accuracy of 92.1%.

Kothadiya et al., [20] proposed hybrid models implemented through LSTM and Gated Recurrent Unit (GRU) which gets a higher accuracy in ISL. Various datasets has been tested using the model.The datasets for Greek Sign Language (GSL), ASL, and IISL2020 have been tested and have achieved an accuracy rate exceeding 90%.

3.METHODOLOGY

3.1 Data Source

Zenodo's INCLUDE is the dataset used in this example. Frequently, the collection consists of video recordings featuring different colors and descriptors. There are 5000 videos altogether in the dataset, which is composed of 100 labels with 50 videos each. Ten signers each produced five videos for each label. There were two sets in the database. In the first, 23 one-handed motions were recorded. 41 signals were introduced by the Second, 22 of which were two-handed and 19 of which were one-handed.

To show variations in illumination across signs, the first video was recorded outside under natural lighting, while the second recording was produced indoors under artificial lighting. Subject 10, who was unavailable for the second set of recordings, was replaced by a new subject. The classes that were recorded in the first and second sessions are different, therefore this update has no bearing on the dataset's usefulness. In both sets of recordings, subjects were dressed in black and made the signals while standing or sitting up against a white wall. Gloves with fluorescent colors were worn by the subjects to simplify the intricacy of hand segmentation within a picture. These greatly lessen the challenge of identifying the hand's location and segmentation along with any issues related to variations in skin tone, all while maintaining the challenge of identifying the hand's form.

3.2 Data Preprocessing

One of the most crucial stages in creating a recognition system for sign language is data preparation. To preprocess the data, we converted the unprocessed video data into labels, with 50 videos per label. For every 100 words, 100 labels have been made and added to the video's dataset.

The following Python libraries have been utilized for preparing the data:

3.2.1 OpenCV

OpenCV (Open-Source Computer Vision Library) is suitable for systems that recognize sign language to process and analyze video or image data and extract relevant information for understanding sign language gestures. OpenCV provides a set of functions to capture video streams from cameras or process pre-recorded videos. This functionality is used to acquire video input for sign language recognition.

3.2.2 Keras

The creation, training, and assessment of deep learning models can be aided by the use of Keras, a high-level neural networks API, for sign language recognition. For creating deep learning model architectures, Keras offers an easy-to-use and simple interface. It lets you specify the neural network's layers, linkages, and setups that will be utilized to recognize sign language. To create the required network structure, you may select from a range of layer types, such as convolutional layers, pooling layers, and fully connected layers.

scikit-learn

A comprehensive machine learning toolkit called scikit-learn may be used for a variety of tasks related to sign language recognition, such as feature extraction, data preparation, model training, and assessment. Scikit-learn has a number of methods and tools for preprocessing data, which is useful for recognition of sign language. StandardScaler and MinMaxScaler routines can be put into use to normalize the input data, ensuring that various features have equivalent scales. Scikit-learn also provides procedures to handle missing data, encode categorical variables, and use dimensionality reduction or feature selection methods.

3.2.3 Pandas

Pandas, a powerful data manipulation and analysis library, can be put into use in sign language recognition to handle, preprocess, and organize data efficiently. Pandas has functions for loading data from databases, CSV files, and Excel files, among other file types. These functions are used to import sign language gesture data into pandas data structures, such as Data Frames, which provide a path way for easy manipulation and analysis.

3.2.4 Tensorflow

TensorFlow, an open-source machine learning framework, is suitable for systems that recognize sign language to build and train deep learning models that can effectively recognize and classify sign language gestures. TensorFlow's image processing features are suitable for preprocessing the input data. This might entail scaling the pictures, normalizing the values of the pixels, and using augmentation methods like rotation, translation, or flipping to increase the variety of the training set and strengthen the resilience of the model.

3.3 Feature Extraction

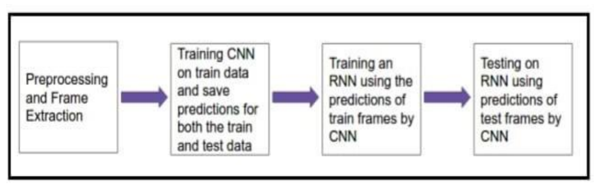

Fig 1: Feature Extraction and Training

Fig 1 represents the methodology for sign language classification, illustrating the steps involved in the process recognizing and classifying sign language gestures. In Feature Extraction, we have transformed the video data into several frames. Each video is segmented into 200 frames. For 3500 videos, the frames extracted were 700000.

For each frame bottlenecks creation has been done which is the gray-scale conversion of the image. The folders are created for the train frames, the test frames, and the bottlenecks. According to their labels, the target folder for the frames is the train frames folder, where they are kept. To store the frames in the test frames folder, continue the same procedure for the test videos. Bottlenecks are created by assembling quick NumPy array functions, and this is the next stage.

After the frames and bottlenecks extraction, it will create two files retrained_labels.txt and retrained graph.pb. The next step is succeeded by the intermediate representation of videos. Each video is represented by a sequence of n dimensional vectors one for each frame. This will create a python pickle file(.pkl) that serializes a tuple of two NumPy arrays (feature, label). The python pickle file will be used by the RNN for training.

4. ARCHITECTURE

4.1 RNN

The model being used in our paper is the Recurrent Neural Networks (RNN). Since our Python pickle file is a tuple of arrays, which is sequential data that can be trained fast, RNN is better for analyzing temporal, sequential data, such as text or movies. The finaloutput is in form of text and RNN is very fast in text classification whereas CNN is faster in image classification.

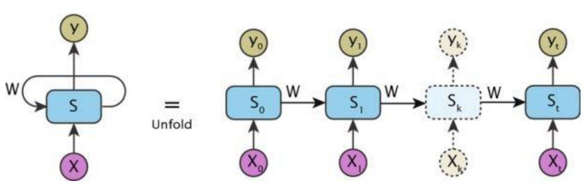

Fig 2: Architecture of RNN

Fig 2 illustrates an unrolled representation of a RNN. Unrolling an RNN means expanding in the time dimension, making it easier to visualize the flow of information and computation. The input layer of an RNN is represented by X, the hidden layer by S, the output layer by Y, and the weight by W. You may conceptualize a recurrent neural network as many feedforward network copies that are each sending a message to a successor. It is as if S_0 is the input for each step and each X is some additional control signal at each step. It is seen that the weight Wis repeated at every layer. So, it's like a deep networkwith the same shared weights between each layer. Conventional neural networks have separate inputs and outputs from each other. Nonetheless, there are scenarios in which words that come before are necessary, such as when a sentence's next word is anticipated. This necessitates that you remember them.

Thus, RNN was created, and it used a Hidden Layer to tackle this problem. What makes an RNN unique and important is its hidden state, which holds particular information about a sequence.

The formula for calculating the current state is

$h_t=f\ (h_{t-1}, x_t)$ where,

h_t=current state,

h_{t-1}=previous state,

x_t=input state.

One step at a time, the input is transmitted to the network. The current ht becomes ht-1 for the next time step. Depending on the circumstances, one can compile data from all prior scenarios and travel back as far in time as necessary. The final current state is utilized to compute the result once all time steps have been completed. An error arises when the output is contrasted with the desired or actual outcome. The network then receives the error back, changes the weights, and trains the network (RNN).

4.2 LSTM

A unique type of RNN called an LSTM is able to identify long-term relationships in data. This is made possible by the model's repeating module, which consists of four interconnected levels. An STM-module may selectively learn, unlearn, or retain information from each of the units owing to its three gates and cell state. The LSTM cell state facilitates unaltered information transfer between units by permitting limited linear interactions.

5. PERFORMANCE ANALYSIS

5.1 Analysis using the Confusion Matrix

For the performance analysis of this model, we have used sklearn metrics and used the confusion matrix.

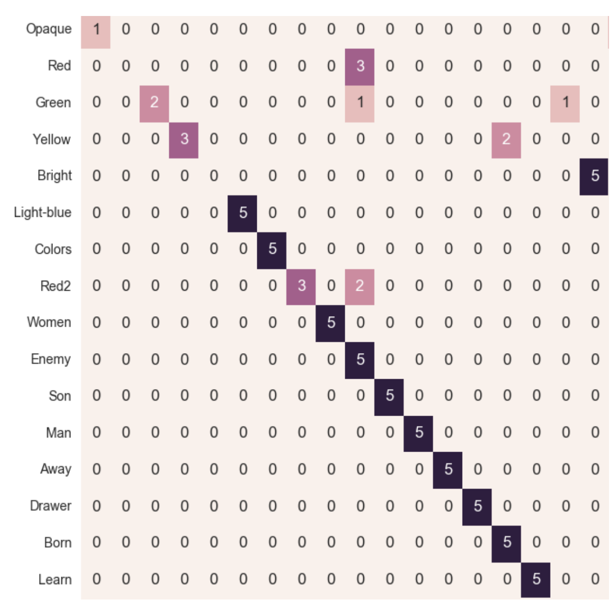

Fig 3: Confusion matrix for words

The confusion matrix for words is shown in Fig. 3. A sign language recognition system's performance is assessed using a confusion matrix. It provides a tabular representation that summarizes the system's predictions against the actual sign language gestures or words.

A table used to assess model errors in classification problems is known as a confusion matrix. The columns show the expected results, and the rows show the real classes that the actual outcomes should be. There aren't many key terms in the confusion matrix. False Positive (FP), False Negative (FN), True Positive (TP), and True Negative (TN) are the four types. TP stands for either the expected class or the predicted value matches the actual class. TN denotes both a negative actual value and a negative predicted value. FP is a type 1 error that denotes a positive actual value but a negative anticipated value. The type 2 error known as FN occurs when a positive value is observed, while a negative value is expected. The formula for calculating the accuracy is:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

Model	**%Accuracy**
Qazi at li [1] (3DCNN)	82
Anshul Mittal at li [3] (CNN with LSTM)	89.05
T Raghuveera at li [11] (SVM)	71.85
Sanil at li [18] (SIFT-HOG)	52.23
Sridhar at li [19] (LSTM and MobileNetV2)	92.1
Kothadiya at li [20] (LSTM and GRU)	90
Proposed System (CNN-RNN)	96

Table 1: Comparative Analysis of diff models

The accuracy percentages of the various approaches are displayed in Table 1. The suggested architecture provides a greater accuracy rate than the other models and architectures listed above, as the table illustrates. Thus, it is possible to achieve reduced error rates by using the suggested model.

6. CONCLUSION

The Indian Sign Language (ISL) generally a complexSign language consisting of gestures, facial expressions, and body movements used by the deaf and hard of hearing community in India. However, identifying and recognizing these sign language gestures poses many challenges due to the lack of standardization, complexity, and variety of signs. To address this issue, several computer vision-based approaches have been proposed that utilize different features, classifiers, and modalities. In this work, we looked at many deep learning models for recognizing sign language. CNN and LSTM, two deep learning models, are the best. The CNN-LSTM model recognizes complicated and dynamic motions because it takes into account both the spatial and temporal information of the sign language gestures. With the CNN and LSTM Sign Language translation technology, communication between the common and disabled is ensured. Such a system can enable the deaf community to communicate effectively with others, breaking downcommunication barriers and improving their quality of life.

Finally, our research demonstrates the promise of deep learning models for ISL detection in addition to the significance of establishing accurate and robust systems to assist the deaf and hard of hearing community. Further improvements in accuracy, robustness, and scalability can lead to the development of more advanced and effective SLR systems for ISL recognition. Our model has an accuracy of 98%, which is better than the prior model's 96% accuracy.

Future Scope would be to expand the capacity to identify continuous sign language words or phrases would provide more expressive and natural communication. It will also investigate the usage of wearable technology, such as wristbands or smart gloves, for sign language identification.

7. REFERENCES

[1] Areeb Q. M., Marayam, Mohammed Nadeem, Roobaea Alroobaea, and Faisal Anwer, "Helping Hearing-Impaired in Emergency Situations: A Deep Learning-Based Approach." *IEEE Access (Accepted on January 8, 2022; date of publication January 13, 2022).* DOI: 10.1109/ACCESS.2022.3142918.

[2] Sruthi C.J and Lijiya A, "Signet: A Deep Learning Indian Sign Language Recognition System." *In International Conference on Communication and Signal Processing*, April 4-6-2019, India.

[3] Mittal A, and Kumar P, "A Modified-LSTM Model for Continuous Sign Language Recognition using Leap motion", *IEEE Sensors Journal*, 2019. DOI: 10.1109/JSEN.2019.2909837.

[4] Likhar P, and Bhagat N.K, "Deep Learning Methods for Indian Sign Language Recognition", *In IEEE 10th International Conference on Consumer Electronics*, ICCE-Berlin,2020.

[5] Chhabria, Kunal and Priya V, "Gesture Recognition Using Deep Learning", *International Conference on Emerging Trends in Information Technology and Engineering* (icETITE), 2020.

[6] Thanekar, Aadit and Deepak J, "Video Chat Application for Mutes", *International Conference on Emerging Smart Computing and Informatics (ESCI)*, AISSMS Institute of Information Technology, Pune, India, Mar 5-7, 2021.

[7] Thanasekhar B, Kumar G and Deepak, "Real Time Conversion of Sign Language using Deep Learning for Programming Basics", *2019 11th International Conference on Advanced Computing (ICoAC)*, DOI: 10.1109/ICoAC48765.2019.246807.

[8] Cheok Ming Jin, "A review of hand gesture and sign language recognition techniques", *International Journal of Machine Learning and Cybernetics*, 2019, pp: 131-153, DOI: 10.1007/s13042-017-0705-5.

[9] Elakkiya R, "Machine learning based sign language recognition: a review and its research frontier", *Journal of Ambient Intelligence and Humanized Computing* 12, 2021, pp: 7205-7224, DOI: 10.1007/s12652-020-02396-y.

[10] Rahejaa J. L, and Mishrab A. "Indian Sign Language Recognition Using SVM", *Pattern Recognition and Image Analysis* 26, no. 2 ,2016, DOI: 10.1134/S1054661816020164.

[11] Raghuveera T and Deepthi R, "A depth-based Indian Sign Language recognition using Microsoft Kinect", *Sadhana* 45:34, 2020, DOI: 10.1007/s12046-019-1250-6.

[12] Lim Kian Ming, "Isolated sign language recognition using Convolutional Neural Network hand modeling and Hand Energy Image", *Multimedia Tools and Applications*,2019, 78:19917-19944, DOI: 10.1007/s11042-019-7263-7.

[13] Sreemathy R, "Sign language recognition using artificial intelligence." *Education and Information Technologies,* 2022, DOI: 10.1007/s10639-022-11391-z.

[14] Wadhawan Ankita and Kumar Parteek, "Deep learning-based sign language recognition system for static signs", *Neural Computing and Applications* 32:7957-7968, 2020, DOI: 10.1007/s00521-019-04691-y.

[15] Nimisha K.P and Jacob A, "A brief review of the recent trends in sign language recognition." *In International Conference on Communication and Signal Processing (ICCSP)* (2020), 186-190.

[16] H. Li and M. Greenspan, “Model-based segmentation and recognition of dynamic gestures in continuous video streams. Pattern Recognition”, 44(8):1614–1628, 2021.

[17] Okan Kpkl, Ahmet Gunduz, Neslihan Kose and Gerhard Rigoll, “Real Time Hand Gesture Detection and Classification using Convolutional eural Networks”, paper accepted to *IEEE International Conference on Automatic Face and Gesture Recognition* (FG 2019).

[18] Sanil Jain and K.V.Sameer Raja, “Indian Sign Language and Character Recognition”, *vinsam,* Available:https://cse.iitk.ac.in/users/cs365/2015/_submissions/vinsam/report.pdf.

[19] Sridhar A, Ganesan R. G, Kumar P and Khapra M, “Include: A large scale dataset for indian sign language recognition”. In Proceedings of the *28th ACM international conference on multimedia*2020(pp. 1366 – 1375).

[20] Kothadiya D, Bhatt C, Sapariya K, Patel K,Gil-González and Corchado JM, “Deepsign: Sign language detection and recognition using deep learning”, Electronics, 11(11), 1780, 2022.