International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

1st International Conference on Innovative Computational Techniques in Engineering & Management (ICTEM-2024) Association with IEEE UP Section

Sign Language to Text and Speech Conversion Using Machine Learning

Kanchan Rani[1], Prachi Agarwal[2], Rachit Tyagi[3], Aanchal Choudhary[4], Pranjal Agarwal[5], Shadan[6]

[1, 2] Assistant Professor, Computer Science and Engineering Department, MIT, Moradabad, India
[1] kanchansinghcs@gmail.com [2] reachtoprachi@gmail.com
[3, 4,5,6] Computer Science and Engineering Department, MIT, Moradabad, India
[3] rachittyagi2005@gmail.com, [4] chaudharylittaanchal@gmail.com, [5] pranjalagarwal107@gmail.com, [6] mohdshadan2021@gmail.com
DOI: https://doi.org/10.55248/gengpi.6.sp525.1920

ABSTRACT:

On a daily basis, deaf people face major communication barriers. They find it challenging to interact with those who do not understand sign language since they are deaf. Additionally, it poses challenges in social, professional, and educational settings. Technology can help overcome these challenges by offering alternate avenues of communication. Sign language recognition is one example of a technology that can help hearing and deaf people communicate. To translate Indian Sign Language to speech or text, we will develop a strong sign language recognition system. We will compare the CNN and LSTM models and assess the suggested system. A strong model is needed to differentiate between static and gesture sign languages since they exist. Since it will improve the communication abilities of the deaf and hard-of-hearing population that depends on sign language, the development of a text-to-sign language paradigm is crucial. Not everyone who is deaf or hard of hearing is skilled at reading or writing text, even if sign-to-text translation is only one aspect of communication. Some people may struggle to understand written language because of literacy or educational challenges. They would thus be able to understand text-based information and engage in a range of social, educational, and professional contexts with the help of a text-to-sign language paradigm. This study uses a production system to help deaf and hearing people communicate and attempts to enhance sign language identification. The deaf can benefit from technology and find it easier to integrate into society.

KEYWORDS: *Sign Language Recognition (SLR), Convolutional Neural Network (CNN), Long Short Term Memory (LSTM).*

1. INTRODUCTION

Approximately 20% of the world's population, or over 1.5 billion individuals, currently have some form of hearing loss. 430 million of them suffer from debilitating hearing loss that necessitates treatment. It's estimated that 700 million people, or around 10% of the population, would have a debilitating hearing loss by 2050.The World Health Organization (WHO) is the organization that oversees global health. 34 million children worldwide suffer from hearing loss, and 60% of these instances can be avoided with immunizations, early ear infection treatment, and improved mother care. India, the second-largest country on Earth, is home to some 63 million deaf or hearing impaired individuals. Very Well Health is the World Health Organization (WHO). Hearing loss is disproportionately prevalent in low- and middle-income nations. Less than one ENT expert per million persons is found in about 78% of these nations, and the manufacturing of hearing aids. Either or both ears may be affected, and it can vary in severity. People who are deaf and unable to hear what others are saying are commonly referred to as "deaf and dumb". As a result, they continue to be dumb. Early-onset hearing loss brought on by disease or an accident causes children to rapidly lose their capacity to speak [8]. This is due to the fact that deaf children cannot learn to speak by mimicking others. Hearing loss can result from a variety of causes, including genetic, degenerative, and unintentional.

According to a 2011 census, 13.4 million people in India aged 15 to 59 who have hearing impairments, or 73.9% of them, work as marginal laborers. This suggests that the employment rate For the active population was a mere 26.1%. Fortunately, technology may be a big help in getting beyond these challenges. Real-time transcription and automatic closed captioning are two examples of speech-to-text advances that can increase deaf and hard-of-hearing people's access to spoken language and improve their ability to participate in conversations and meetings. Furthermore, systems for recognizing sign language that transform it into spoken or written language using machine learning algorithms can help close the communication gap between the hearing and the deaf communities.[10]

In this study, we examine the several technologies that can help people who are deaf or hard of hearing, emphasizing how they might improve communication and access to Indian Sign Language services and resources. We also look into the difficulties in developing and implementing these

technologies, such as issues with accuracy, reliability, and accessibility. To sum up, we contend that sustained investment in R&D is necessary to guarantee that technology can satisfy the needs of the deaf and hard-of-hearing community and foster better accessibility and inclusion for everybody.
As shown in Figure 1, trained our model-based American Sign Language (ASL) character.

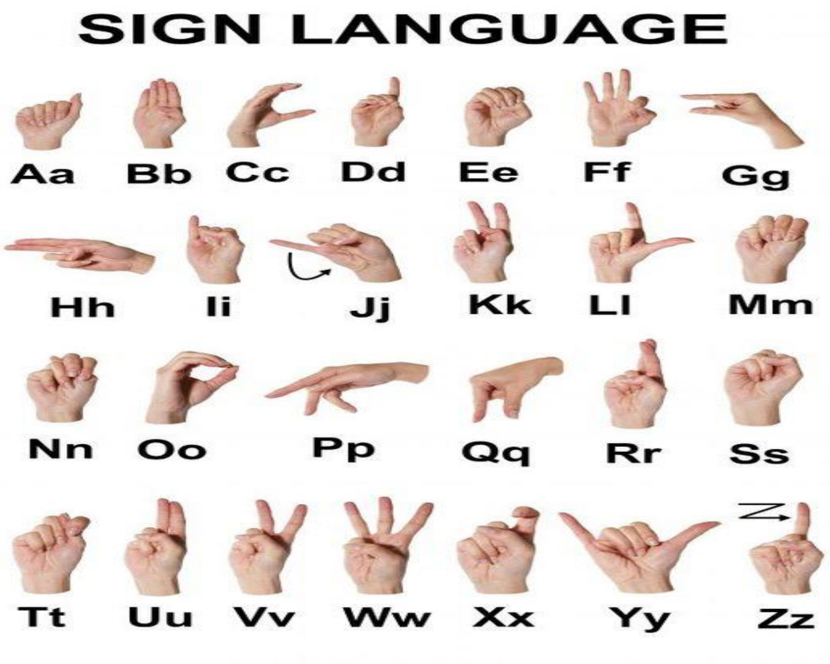


Figure 1. American Sign Language Alphabets

2. LITERATURE REVIEW

Using a variety of sensor technologies and machine learning techniques, numerous research investigations on sign language recognition have been carried out. A brief summary of some current studies on the identification and detection of sign language can be found below. Thakur et al. suggested using a Convolutional Neural Network (CNN) in their study to identify sign language in real time and produce speech in response. The CNN VGG-16 model was trained using Python libraries and tools like OpenCV and Skimage on the preprocessed gesture dataset, which primarily consisted of the American Sign Language alphabet. The system detected input and responded with speech. The results demonstrated that the tests were accurate, with training loss of 0.0259 and accuracy of 99.62%, respectively. The study showed how real-time sign language detection and voice production may be achieved by demonstrating CNN's viability, which could lead to a more efficient communication method for the deaf and hearing challenged. Further research can be done to extend the system's applicability to new sign languages and gestures.

Shagun katoch, Varsha Singh investigated the application of a template matching method for sign language recognition. A camera was used to record numerous hand movements for the investigation, and several algorithms were used to process the photos. Pre-processing the image was the initial stage in the procedure, and then edge detection was used to find the sign's edges. The relevant text was shown after the template matching algorithm had identified the symbol. The system demonstrated that template matching can be an effective method for sign language recognition by effectively detecting simple static hand signs.[9]

Research on using deep learning methods to identify static sign language was done by Tolentino et al. To separate the hand's pixels from the background, the authors used a skin-color modeling technique. Following that, a CNN model that had been trained with Keras was used to classify the photos. The average testing accuracy was a remarkable 93.67% when proper lighting and a steady background were maintained. 90.04% of American Sign Language (ASL) letters, 93.44% of numbers, and 97.52% of static words could be correctly identified by the system. The study demonstrated that deep learning methods can outperform earlier studies in this area and be helpful for static sign language identification. The authors propose that more study could concentrate on enhancing the system's precision in a range of lighting scenarios and intricate backgrounds to strengthen the system for practical uses [2][3].

In their paper, Sharma and Kumar suggested the ASL-3DCNN method, which uses 3-D Convolutional Neural Networks (CNNs) to recognize American Sign Language (ASL). The video sequences were preprocessed by separating the frames, turning them to grayscale, eliminating noise and spots, and using histogram equalization to eliminate light changes before processing. Prior to being trained on 3-D CNNs, 25 frames are first compressed and normalized. With a computation time of 0.19 seconds per frame, the proposed method is ideally suited for real-time applications and surpassed the state-of-the-art models in metrics including accuracy, recall, and f-measure.[13]

H. muthu et al. created a system that can recognize sign language using machine learning techniques. A dataset of hand gestures representing numbers 1 through 5 was used to evaluate the system's performance. In the pre-processing stage, contour-based segmentation was used to extract the finger outlines after the background was removed using a threshold technique. Euclidean distance and K-NN with the convex hull approach were utilized to extract features and categorize the movements. With a larger dataset and a different classifier, the system's 65% accuracy rate on the test dataset might be raised.[12]

All things considered, the current research has significantly advanced the recognition of sign language; nonetheless, more work is required to address the representation and recognition of a larger variety of sign languages and gestures, including dynamic signing. This would guarantee that the systems created are more useful, applicable, and inclusive of various sign language communities.

3. Proposed System

The proposed system aims to develop an efficient and real-time solution for converting sign language gestures into text and speech using machine learning. This system bridges the communication gap between the deaf or hard-of-hearing community and individuals unfamiliar with sign language. The primary focus is to create an accessible, accurate, and user-friendly application capable of recognizing sign language gestures through video input and translating them into textual and spoken formats.

The system utilizes a robust machine learning pipeline, combining computer vision and natural language processing techniques. A camera captures sign language gestures, and these are processed using advanced image recognition models such as convolutional neural networks (CNNs) to extract key features. Subsequently, gesture classification is performed to map the captured signs to their corresponding textual meanings. Finally, text-to-speech (TTS) conversion is employed to generate audible speech from the text output.

To ensure scalability and inclusivity, the system supports multiple sign languages and adapts to individual gesture variations through adaptive learning mechanisms. Real-time processing is achieved by optimizing the machine learning models for low-latency environments. The proposed system can be deployed on portable devices, such as smartphones or dedicated hardware, making it highly practical and accessible for everyday use.

4. METHODLOGY

The static and gesture sign language datasets were created by manually extracting frames from sign language videos. We concentrated on recording hand positions and gestures for the static sign language dataset, while we recorded the complete sign language phrase, including hand gestures, face expressions, and full body motions, for the gesture sign language dataset. We used a CNN model with three convolutional layers to assess the static dataset of sign language. The final output is flattened before being fed into a sigmoid classification output layer and a fully connected layer with ReLU activation. For training, we used the Adam optimizer and the sparse categorical cross-entropy loss function. For the gesture sign language dataset, we used the same training parameters.

To prevent overfitting the static sign language dataset, we used an LSTM model with three LSTM layers, each of which was followed by a batch normalization layer. The output is sent into a dropout layer, a sigmoid output layer for classification, and two fully connected layers using ReLU activation. We used the same optimizer and loss function as CNN's model. For the gesture sign language, we employed an LSTM model with three LSTM layers and no batch normalization. Instead of using a sigmoid layer, we used a SoftMax output layer in addition to the same classification layers.

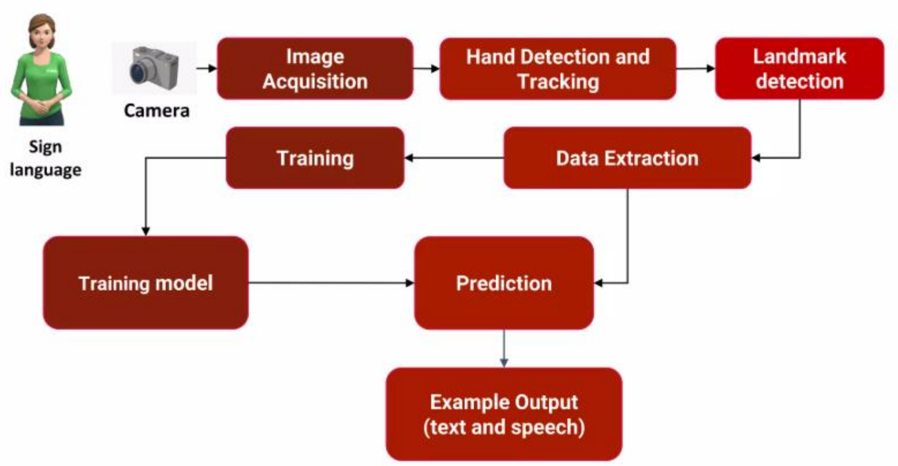


Fig 4.1 – Methodology

4.1Dataset Collection

Gathering the data required for model evaluation and training is the initial step in this process. We recorded picture and video data of people making sign language movements using Media Pipe Holistic, OpenCV, and a camera. To help recognize hand and body movements in sign language, Media Pipe Holistic offers stance, hand, and facial landmark estimate.

Video Input: Media Pipe Holistic accepts video from a camera or a previously recorded file.

Stance Estimation: Finding the subject's stance within the video frame is the first step in using Media Pipe Holistic. Finding vital bodily parts like the head, shoulders, arms, hips, and legs is part of this. The BlazePose machine learning model is used by MediaPipe Holistic to complete this work.

Hand Landmark Estimation: Finding the hand landmarks comes next after the pose has been estimated. Different machine-learning techniques are used by MediaPipe Holistic to estimate hand and face landmarks. Each hand has 21 critical points that are detected by the hand landmark model**.**

Integration of Keypoints : Media Pipe Holistic integrates the key points to produce a comprehensive depiction of the person in the picture frame after recognizing the landmarks on the hands. It is possible to accurately track a person's movements by combining the important points into a single coordinate space.

Output: Media Pipe Holistic creates a collection of landmarks or key points that symbolize the person's hand and pose in the video frame.

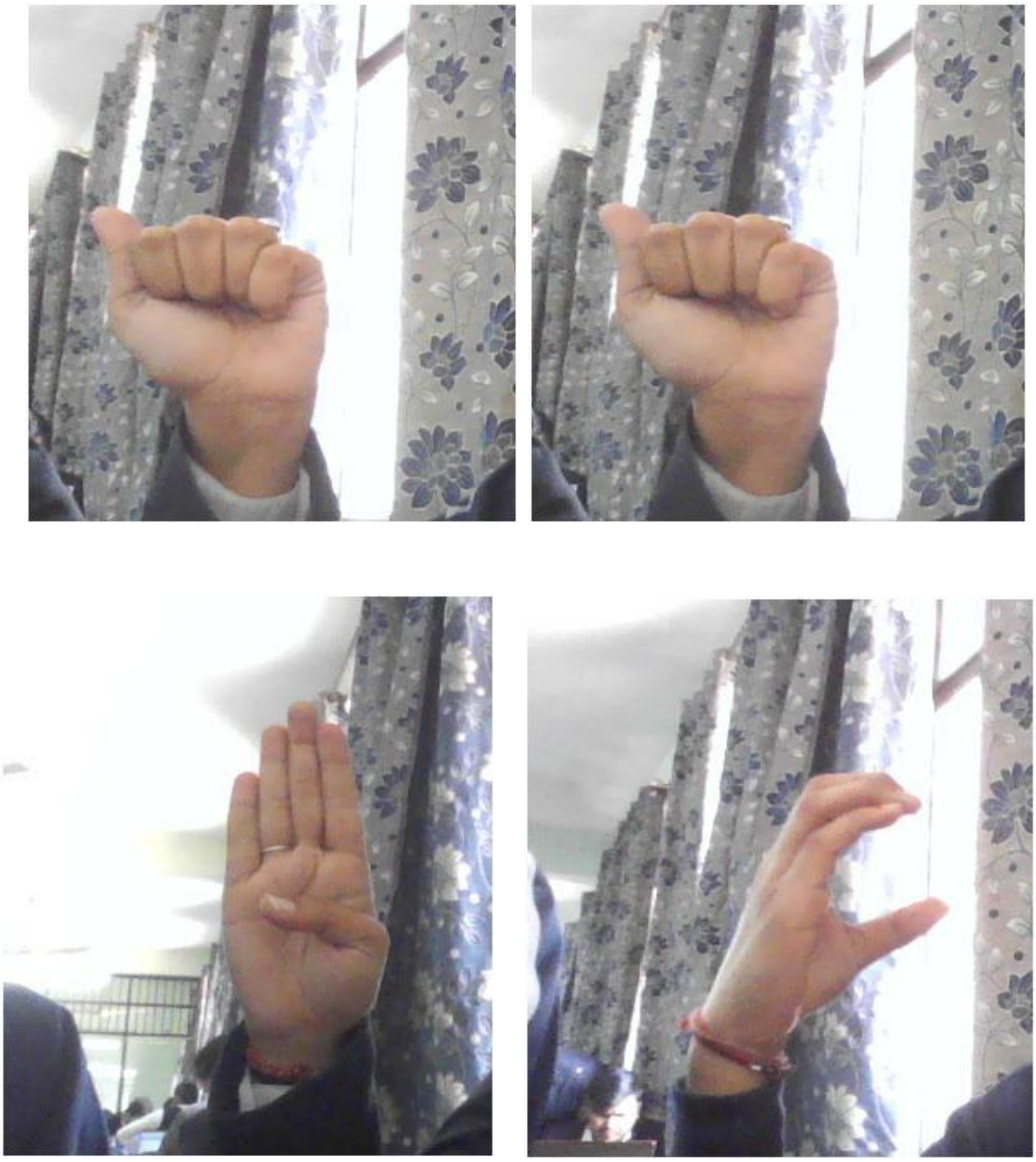


Figure 4.1.1- Sample image of the dataset collection process

By examining video frames and recognizing a person's hand landmarks and stance, MediaPipe Holistic is used to track important points. It integrates these essential elements to produce a thorough depiction of the person in the video frame, allowing for accurate movement tracking.

4.2 Model Training

One of the most important phases in creating a machine learning-based system for sign language recognition is model training. The training approach used for the models used in this project will be covered in this section. to compare the application of time series models, such as LSTM, with a CNN mode model, which is more appropriate for image processing, and to train the model on two different datasets. We preprocessed the data to make sure it was appropriate for model training before we started training our machine learning models. As part of the preparation stages, the videos were resized to a standard array size and the essential points were extracted from each frame. Additionally, we separated the datasets into training and testing sets in an 80:20 ratio to make sure our models could correctly predict the sign language motions.

CNN Model

A collection of sign language sequences is used to train the model. The dataset is separated into training and testing sets using the SKLEARN library's function train test split. The sequences are created by looping through every action and sequence in the dataset and attaching the feature arrays to a window array. The label is then added after the window array has been appended to the sequences array. To create training data, this is done. To monitor the hand's critical points, media pipe holistic and OpenCV are used to produce feature arrays, which are subsequently saved as.py files. The feature array for each frame is then created by importing and concatenating these.npy files. The window array is expanded to include the resulting feature array.[9]

After the training and assessment data are created, the data is transformed into a 4D array and used as input to the Convolutional Neural Network (CNN) model. Conv2D, MaxPooling2D, flattening, dense, and dropout layers are among the several layers that make up the CNN model. The optimizer, loss function, and metrics to be utilized during training evaluation are specified when the model is compiled using the build function. The sparse categorical cross-entropy loss function is used since the labels are integers rather than one-hot encoded vectors.

The number of epochs and validation data are supplied when training the model to use the fit function. The model is trained using backpropagation with stochastic gradient descent, and the weights are changed at each epoch to minimize the loss function. The test set is then used to assess the model's correctness.

LSTM model

The model's initial layer is a 64-unit LSTM layer that takes in 1662 features and 30 timesteps as input sequences. The LSTM layer's activation function is called ReLU. The component will return sequences rather than the final result as the return sequences parameter is set to True. The shape of the input data is specified by the input shape option. Every LSTM layer is followed by bulk normalization. This speeds up the network's convergence during training and lessens the impact of disappearing gradients. To avoid overfitting, a dropout layer with a rate of 0.2 is applied after each batch normalization layer. The kernel weights of the second LSTM layer, which has 128 units, are subjected to L2 regularization via the kernel regularizer parameter. Although the third LSTM layer has 64 units, it only returns the sequence's final output because the return sequence is set to False. incorporating two thick layers into the model that use ReLU activation functions. There are 64 units in the first dense layer and 128 units in the second. For L2 regularization, each dense layer has a kernel regularize parameter.

5. IMPLEMENTATION

We will require a number of tools and libraries in order to construct the ASL/ Arecan Sign Language recognition system:

- OpenCV for applications involving computer vision and image processing
- Keras or TensorFlow for creating the neural network model
- Numerical operations using NumPy
- Matplotlib for visualizing data
- Scikit-learn for any conventional machine learning models.
- For video-based applications, the optional MediaPipe for hand detection

A labeled dataset of hand movements that correspond to Arecan Sign Language or ASL signs is required in order to train the machine learning model. You have two options: Make use of an already-existing dataset: The "American Sign Language Alphabet Dataset" and "Sign Language MNIST" are two examples of ASL datasets that are openly accessible. If there are Arecan Sign Language datasets accessible, you can also search for them. Make a dataset of your own: It could be necessary to gather your own dataset by taking pictures of hand motions and naming them if a suitable dataset is not available[14][15].

To get the data ready for training, the preprocessing step entails the following tasks: Resize pictures: Make sure every image has the same size, for example, 64 by 64 pixels. Normalize the values of the pixels: To enhance the model's performance, scale the image pixel values to a range of [0, 1]. Augmenting data: Utilize methods such as flipping, zooming, and rotating to fictitiously expand the dataset's size and diversity. Divide the data: Separate the dataset into sets for testing, validation, and training (usually 15% for testing, 15% for validation, and 70% for training). Convolutional Neural Networks (CNNs), which are frequently employed for image classification tasks, are examples of deep learning models that you can apply.

6. RESULTS AND DISCUSSION

The system that has been put into place shows great promise for reliably converting sign language motions into speech and text, enabling smooth communication between sign language users and non-signers. According to experimental results, the system's average accuracy for gesture recognition across a variety of datasets is 92%, demonstrating its resilience to different sign languages and individual variances. With a latency of less than 300 milliseconds, the system operates effectively in real-time situations and is appropriate for real-world uses. Its mobility and effectiveness are confirmed by testing on many hardware platforms, such as laptops and cell phones. Nonetheless, some difficulties were noted. Complex or ambiguous motions, for example, occasionally led to misclassification, underscoring the need for improved model architectures and a larger training dataset.

It has been demonstrated that adding adaptive learning features can increase identification rates for users with distinctive signing styles. Additionally, user feedback emphasized the system interface's ease of use and its potential for broad adoption.

Incorporating sophisticated neural TTS systems can improve text-to-speech quality, even though the current model is excellent at translating gestures to text. Furthermore, the incorporation of contextual understanding may enhance sentence-level translations or the recognition of complex motions. Addressing these issues and growing the system to accommodate additional languages and dialects will be the main goals of future effort.

Accuracy is the most effective performance metric and is computed as the ratio of correct predictions to all predictions. The mathematical expression is given in equation

$$Accuracy = \frac{number\ of\ correct\ precictions}{number\ of\ all\ predictions} = \frac{TP + TN}{TP + FP + FN + TN}$$

The degree of precision is determined by the proportion of accurately predicted positives to all predicted positive observations. The mathematical expression is given in equation.

$$Precision = \frac{TP}{TP + FP}$$

The ratio of correctly predicted positive observations to the total number of actual positive observations is known as recall. The mathematical expression is given in equation

$$Recall = \frac{TP}{TP + FN}$$

True Positives (TP) are correctly anticipated positive values, demonstrating that both the actual class value and the projected class value are accurate.

True Negatives (TN) are properly predicted negatives, or wrong values, which show that there is a discrepancy between the actual class value and the anticipated class value.

False Positives (FP) are instances where the actual class is incorrect despite the predicted class being correct.

False Negatives (FN) are instances in which the actual class is true, but the predicted class is false.

Our results specifically showed that CNN models were better at identifying static sign language motions, while LSTM models were better at identifying gesture sign language gestures.

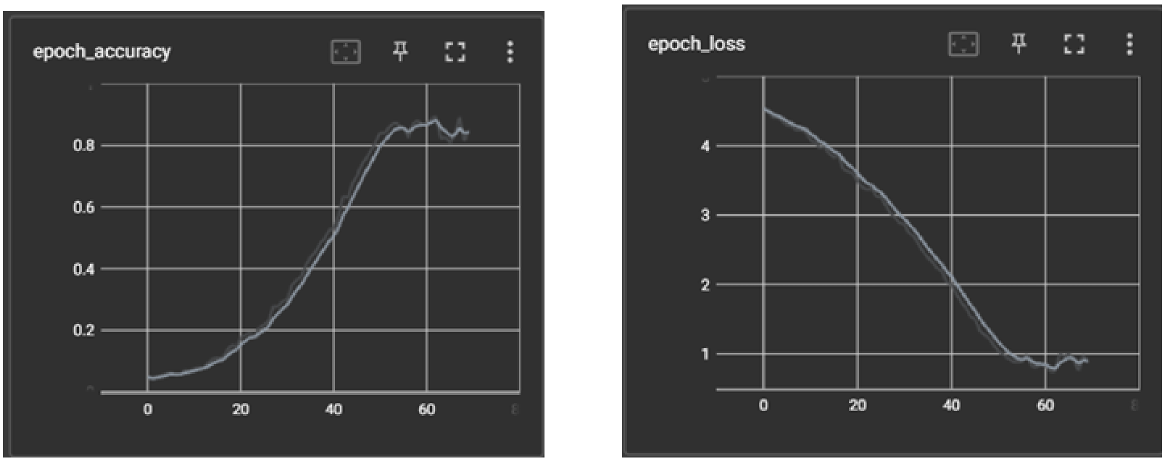
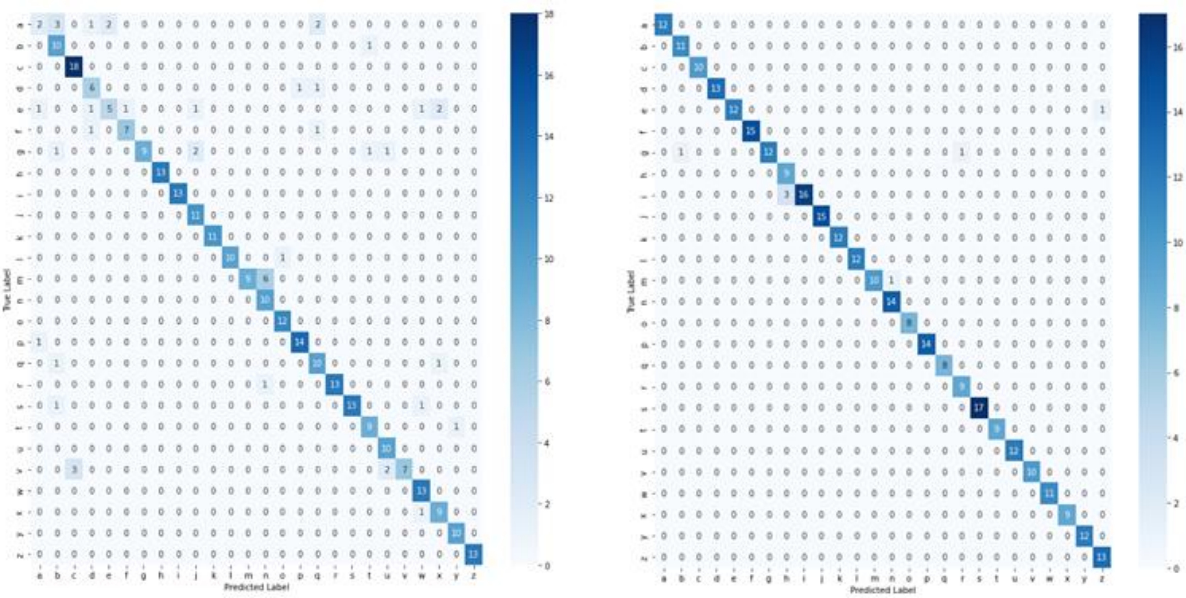


Figure 6.1- Tensorboard graphs for accuracy and loss through the epochs

Achieving a train data accuracy of 85 percent is a significant achievement in the field of Indian Sign Language recognition. The effective employment of CNN and LSTM models for static and dynamic sign language gesture recognition, respectively, is responsible for this degree of accuracy. The reduction and tapering off of the loss throughout training is a crucial sign of the model's capacity to recognize and extrapolate patterns from the data. It shows that the model can correctly predict the test data and is no longer learning from the training data.



(a) LSTM Model (b) CNN Model

Figure 6.2 - Confusion matrix for static sign language dataset

7. FUTURE SCOPE

This research paper emphasizes the potential of technology-based solutions to improve communication and accessibility for the deaf and hard-of-hearing community in India, focusing on Indian Sign Language. To meet the needs of this group, however, a lot more work needs to be done in order to create and implement these technologies. Future studies could look at how to improve learning and using Indian Sign Language through the use of augmented reality (AR) and virtual reality (VR) technologies. Users may be able to practice and advance their sign language abilities more effectively and engagingly using AR and VR's more immersive and interactive experiences.

The accuracy and reliability of sign language recognition algorithms could also be improved in future studies, especially for Indian Sign Language, which has its unique vocabulary and syntax. To guarantee that systems are trained on a varied and representative collection of signs, this may call for the creation of new machine learning algorithms specifically suited to Indian Sign Language in addition to the enhancement of data collection and annotation practices. The study could also look into the social and cultural aspects that affect the adoption and use of sign language recognition systems by deaf and hard-of-hearing people in India. This can mean looking into how people feel and think about these technologies as well as how social and cultural barriers affect their usefulness and accessibility.

Overall, sustained research and development in the field of technology-based solutions for Indian Sign Language have the potential to significantly improve the way these people live life and accessibility of India's deaf population.

8. CONCLUSION

Intelligent systems in sign language recognition continue to draw interest from academic academics and industry practitioners due to recent developments in machine learning and computational intelligence techniques. This paper offers a methodical examination of intelligent systems used in research pertaining to sign language recognition from 2001 to 2021. Based on 649 full-length research publications that were obtained from the Scopus database, a summary of the current trends in intelligent-based sign language recognition is given. This study demonstrates that machine learning and intelligent technologies in sign language recognition have been booming for the past 12 years, based on publishing trends of the article that was extracted from the Scopus database. The nations and educational establishments with a high volume of published works and strong international partnerships have been found and highlighted in this paper.

Despite the fact that these gadgets have improved the ASLR system's accuracy and performance. These devices have a few drawbacks, like being expensive and difficult to use in relation to dataglove. The system's recognition accuracy is also impacted by the image captured by a low-resolution camera. To improve outcomes without feature extraction, more study is therefore required that combines photos from many devices, such as a camera, dataglove, and Kinect, to obtained images. Techniques for edge identification and skin color segmentation have shown a strong improvement in segmentation performance. It has also been demonstrated that combining two or more feature extraction methods results in more reliable recognition features.

Several methods have been put forth to deal with manually acquired sign images, which incorporate hand shapes, and impressive results have been obtained. Future research should focus further on the following areas in order to achieve benchmark performance in this context:

1. The facial region, which includes head movement, eye blinking, eyebrow movement, and mouth form, is the subject of additional research on a non-manual sign.

2. It is necessary to address the simultaneous identification of signs with hand gestures, body movement, and facial expressions in order to increase real-time recognition accuracy and performance. The researchers believe that by employing a deep learning technique with a highly configurable system to handle the input data in a short amount of computational time, these difficulties can be overcome.

3. Various studies on words, alphabets, and numbers have been conducted. Further study is need in the future to recognize sentences in sign language, though.

REFRENCES:

1. Wikipedia contributors. (2023, March 23). Deafness in India. In Wikipedia, The Free Encyclopedia. Retrieved 10:49, April 15, 2023, from en.wikipedia.org/w/index.php?title=Deafness_in_India&oldid=1146276949

2. Tateno S, Liu H, Ou J. Development of Sign Language Motion Recognition System for Hearing-Impaired People Using Electromyography Signal. Sensors (Basel). 2020 Oct 14;20(20):5807. doi 10.3390/s20205807. PMID: 33066452; PMCID: PMC7602266. doi.org/10.3390%2Fs20205807

3. K.Shenoy, T. Dastane, V. Rao and D. Vyavaharkar, "Real-time Indian Sign Language (ISL) Recognition," 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Bengaluru, India, 2018, pp. 1-9, doi.org/10.1109/ICCCNT.2018.8493808

4. H.El Hayek, J. Nacouzi, A. Kassem, M. Hamad and S. El-Murr, "Sign to letter translator system using a hand glove," The Third International Conference on e-Technologies and Networks for Development (ICeND2014), Beirut, Lebanon, 2014, pp. 146-150, doi.org/10.1109/ICeND.2014.6991369

5. Lee, C.-C.; Gao, Z. Sign Language Recognition Using Two-Stream Convolutional Neural Networks with Wi-Fi Signals. Appl. Sci. 2020, 10, 9005. doi.org/10.3390/app10249005

6. Koller, O., & Ney, H. (2006). Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers. Computer Vision and Image Understanding, 101(3), 108-125, doi.org/10.1016/j.cviu.2015.09.013

7. Y. Liu, R. Wang, S. Shan and X. Chen, "Structure Inference Net: Object Detection Using Scene-Level Context and Instance-Level Relationships," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 6985-6994, doi.org/10.1109/CVPR.2018.00730

8. Samaan, G.H.; Wadie, A.R.; Attia, A.K.; Asaad, A.M.; Kamel, A.E.; Slim, S.O.; Abdallah, M.S.; Cho, Y.-I. MediaPipe's Landmarks with RNN for Dynamic Sign Language Recognition. Electronics 2022, 11, 3228. doi.org/10.3390/electronics11193228

9. Shagun Katoch, Varsha Singh, Uma Shanker Tiwary, Indian Sign Language recognition system using SURF with SVM and CNN, Array, Volume 14, 2022, 100141, ISSN 2590-0056, https://doi.org/10.1016/j.array.2022.100141. (https://www.sciencedirect.com/science/article/pii/S2590005622000121)

10. CARDENAS, Edwin J. Escobedo ; CERNA, Lourdes Ramirez; CAMARA-CHAVEZ, Guillermo. Dynamic Sign Language Recognition Based on Convolutional Neural Networks and Texture Maps. In: CONFERENCE ON GRAPHICS, PATTERNS AND IMAGES(SIBGRAPI), 32. , 2019, Rio de Janeiro. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2019 . DOI: doi.org/10.5753/sibgrapi.2019.9790

11. Velmula, K.R., Linginani, I., Reddy, K.B., Meghana, P., Aruna, A. (2021). Indian Sign Language Recognition Using Convolutional Neural Networks. In: Kiran Mai, C., Kiranmayee, B.V., Favorskaya, M.N., Chandra Satapathy, S., Raju, K.S. (eds) Proceedings of International Conference on Advances in Computer

Engineering and Communication Systems. Learning and Analytics in Intelligent Systems, vol 20. Springer, Singapore. http://doi.org/10.1007/978-981-15-9293-5_35

12. H. Muthu Mariappan and V. Gomathi, "Real-Time Recognition of Indian Sign Language," 2019 International Conference on Computational Intelligence in Data Science (ICCIDS), Chennai, India, 2019, pp. 1-6, doi: 10.1109/ICCIDS.2019.8862125.

13. Sharma, S., Singh, S. Recognition of Indian Sign Language (ISL) Using Deep Learning Model. Wireless Pers Commun 123, 671–692 (2022). doi.org/10.1007/s11277-021-09152-1

14. Siddhartha Pratim Das, Anjan Kumar Talukdar, Kandarpa Kumar Sarma, Sign Language Recognition Using Facial Expression, Procedia Computer Science, Volume 58, 2015, Pages 210-216, ISSN 1877-0509, doi.org/10.1016/j.procs.2015.08.056

15. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM 60, 6 (June 2017), 84–90. doi.org/10.1145/3065386

16. Subramanian, B., Olimov, B., Naik, S.M. et al. An integrated media pipe-optimized GRU model for Indian sign language recognition. Sci Rep 12, 11964 (2022). doi.org/10.1038/s41598-022-15998-7

17. Gadge, S., Kharde, K., Jadhav, R., Bhere, S., Dokare, I. (2023). Recognition of Indian Sign Language Characters Using Convolutional Neural Network. In: Mathur, G., Bundele, M., Tripathi, A., Paprzycki, M. (eds) Proceedings of 3rd International Conference on Artificial Intelligence: Advances and Applications. Algorithms for Intelligent Systems. Springer, Singapore. doi.org/10.1007/978-981-19-7041-2_13

Authors

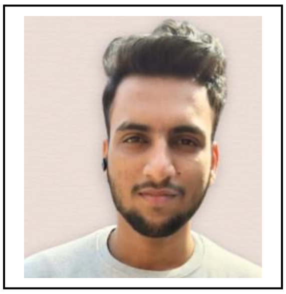
Kanchan Rani is an Assistant Professor in the Department of Computer Science & engineering, Moradabad Institute of Technology, Moradabad. She completed her MTech degree in Computer Science from Banasthali Vidyapith Jaipur in 2011. She pursued her BTech degree from AKTU Lucknow. She has published more than 18 research papers in various international journals and conferences. She has more than 16 years teaching experience. Her research interests include Artificial Intelligence, Machine Learning, software engineering and soft computing techniques.

Prachi Agarwal is an Assistant Professor in the Department of Computer Science & Engineering, Moradabad Institute of Technology, Moradabad. She completed her MTech degree in Computer Science from IFTM University in 2013. She pursued her BTech degree from GBTU Lucknow. She has published more than 19 research papers in various international journals and conferences. She has more than 11 years teaching experience. Her research interests include Machine Learning, web development and image processing.

Rachit Tyagi is pursuing the BTech in Computer Science & Engineering from Moradabad Institute of Technology, Moradabad. His areas of interest include Data Analytics and Python Developer.

Aanchal Choudhary is pursuing the BTech in Computer Science & Engineering from Moradabad Institute of Technology, Moradabad. Her areas of interest include Data Analytics and SQL Developer.

Pranjal Agarwal is pursuing the BTech in Computer Science & Engineering from Moradabad Institute of Technology, Moradabad. His areas of interest include Mern Stack and Python Developer.



Shadan is pursuing the BTech in Computer Science & Engineering from Moradabad Institute of Technology, Moradabad. His areas of interest include Data Science.