IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

Sign2Text & Text2Sign: Bridging Communication Barriers

[1] Pushplata Dubey, [2] Vishwadharini M, [3] Vijaya Vittal Desai, [4] Avirath G S,[5] Vinod V Tallur

[1] Assistant Professor, [2] Student, [3]Student, [4] Student, [5] Student

Department of Computer Science and Engineering,

Cambridge Institute of Technology, Bangalore, India

Abstract: This paper presents a potential solution to break down communication barriers and promote inclusivity across various social and professional settings, bridging the gap between individuals with and without hearing impairments. The "Sign Language to Speech Conversion" and "Speech to Sign Language Conversion" initiative strives to create a real-time system that can translate Sign2Text & Text2Sign. The goal is to facilitate seamless two-way communication between individuals with hearing impairments and those without. The system will leverage advanced technologies CNN and RNNs to accurately recognize a wide range of gestures from sign language inputs. Furthermore, NLP models facilitate seamless translation of text. A key aspect is the real-time function of the system, minimizing delays in the translation process to enable instantaneous communication between ISL users and those relying on spoken language.

Index Terms **- Spatial/temporal relationships, CNN, RNN, NLP.**

I. Introduction

Facilitating effective communication in a diverse and inclusive society is of utmost importance, ensuring equal opportunities and accessibility for hearing and speech impairments. Sign language (SL) serves as a vital medium for hard-of-hearing communities, enabling the expression of thoughts, emotions, and ideas. However, the communication barrier between sign language users and those relying on spoken language remains a significant challenge, often hindering seamless interaction and mutual understanding. The rapid advancement of technology has paved the way for innovative solutions to bridge this communication gap. One promising area of research and development focuses on the conversion between SL and text/speech, aiming to create real-time systems that enable smooth and natural communication between signers and non-signers.

This research paper explores the cutting-edge approaches, challenges, and potential impact of Sign2Text and Text2Sign conversion systems. It explores the application of CNN for gesture recognition and RNN for sequence modeling, which are crucial components in accurately interpreting and generating sign language representations. Similarly, the key steps involved in the conversion to SL include linguistic analysis, content simplification, gesture selection, and video processing techniques aimed at achieving precise and streamlined interpretation. By presenting a comprehensive overview of this conversion pipeline, the research paper aims to contribute towards further advanced and reliable translation systems, ultimately facilitating seamless interaction among SL users and those who rely on written text.

II. LITERATURE SURVEY

There are multiple techniques employed for sign-to-text conversion. Some approaches involve [1] tokenization methods like Recurrent Neural Network-Hidden Markov Model hybrids and attention-based encoders and decoders. However, these models encounter difficulties with regional sign variations or signs specific to certain cultural contexts. Another method [2] classifies hand poses using the K-Nearest Neighbours algorithm, while gesture classification employs Hidden Markov Models with motion and intermediate hand pose sequences as inputs. A limitation of implementing object detection techniques is the demand for a wide variety of annotated hand samples to enable the detection of hands in nearly any position. Real-time sign language interpretation has evolved using image processing frameworks such as OpenCV in [3] which capture the entire signing duration. Gesture detection is then performed using CNN models consisting of multiple layers. Some works focus specifically on finger spelling translation. An alternative approach [4] utilizes OpenCV for contour feature extraction to recognize hand gestures, with a pre-recorded audio clip played upon successful recognition. Improved accuracy can potentially be achieved by employing separate 3D CNN instances for learning fine-grained hand shape features and coarse-grained global body configuration features as done in [5]. Further enhancements involve spatial-temporal fusion [6] using an ST-LSTM fusion attention network, video feature extraction via a dual-stream CNN, and an attention-based Bi-LSTM for correlating video and text, albeit with longer training times. The process of converting speech to sign language involves unique techniques. One approach [7] is to accept audio and text as input, and then match it with a database of sign language videos. If a match is found, the corresponding sign movements are displayed based on the grammar rules of the target sign language. If no match is found, the system can go through additional steps such as tokenization and lemmatization. The core of such a system is natural language processing, which provides capabilities like tokenization, parsing, lemmatization, and part-of-speech tagging. Another interesting approach for real-time speech-to-sign language conversion was the "Speech to Sign Language Interpreter System (SSLIS)" described in [8]. This system was designed to translate spoken English into video representations of ASL in a live, interactive mode. Futermore the core translation functionality, the SSLIS system included several other noteworthy features to enhance its capabilities and usability. For the speech recognition component, the system utilized the Sphinx 3.5 engine. Importantly, the translation process did not strictly adhere to ASL syntax; instead, it employed the manual Signed English (SE) system, which is more closely parallel to English grammar.

III. DATASET

SL is a crucial means of communication for individuals who are deaf or have hearing impairments. It has its unique syntactic framework and relies on gestures and body movements. Sign language involves manual gestures performed with hand poses and non-manual features expressed through eye, mouth, and gaze movements.

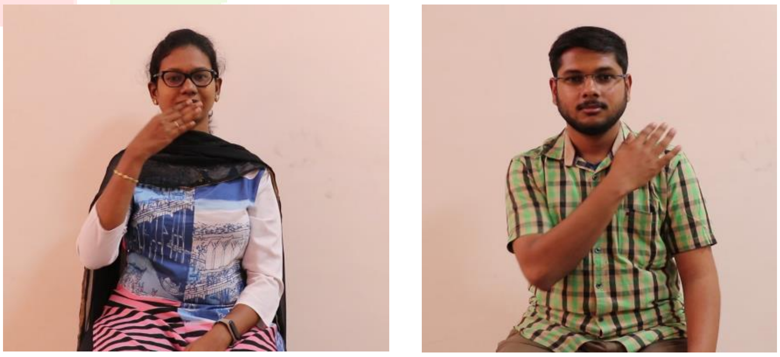


Figure 1: Sample ISL gesture image of “Thank” and “Sorry”

In this work, we have utilized the Indian Sign Language Dataset. Unlike some other datasets, the Indian sign language dataset includes gestures that involve the entire upper body. To capture this, we recorded videos of various sign language users during conferences and community events related to Indian Sign Language (ISL). The corpus consists of a diverse collection of fully annotated videos recorded from various sources, covering a large vocabulary of signs. The dataset includes gestures at both the sentence and word levels. This corpus was created to address the challenges faced in Sign Language Recognition and Translation (SLRT) and aims to improve translation and recognition performance significantly. The videos are annotated with corresponding spoken language sentences to provide a clear understanding of the corpus data.

IV. Methodology

4.1 Sign Language to Speech Conversion

The initiative aims to create a real-time system that converts sign language into text and speech. The objective is to enable effortless communication between individuals with hearing impairments and those without. The proposed solution utilizes cutting-edge technologies like CNN and RNN, which are deep learning algorithms. These algorithms enhance the system's capability to precisely identify a diverse collection of gestures, facilitating accurate sign language recognition and conversion.

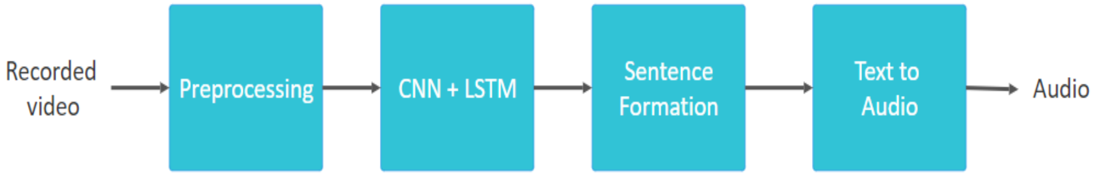


Figure 2: SL to Speech conversion methodology

Real-world data often contains noise and may contain inconsistencies, gaps, and missing values, and might be in a format incompatible with direct utilization in machine learning models. Data preprocessing is a pivotal step for cleaning and preparing the data to make it appropriate for these models. This process also helps increase the precision and efficacy of ML algorithms. With respect to this project, the sign language video data needs to undergo preprocessing to prepare it for the ML model.

The different preprocessing methods employed for sign language recognition and conversion in this project are as follows

4.1.1 Keyframe extraction

Video processing can be described as a computationally intensive task. Even a short 3-second video may consist of over 100 frames and can be resource-intensive to memory and processing power. Additionally, many video snapshots contain redundant information. Processing every single frame to predict a gesture is impractical. To address these challenges, a technique called keyframe extraction is employed. Instead of processing all frames, only a selected set of keyframes is extracted and processed, ensuring that no crucial information is lost. In a video, certain frames represent the rest state of the person recording, where no gesture is being performed. These frames are not applicable for further modeling and analysis. Moreover, the existence of such irrelevant frames can introduce ambiguity during Gesture identification and categorization

To eliminate frames representing the rest position, a method based on Root Mean Square Error (RMSE) is used. By calculating the RMSE between consecutive frames, frames with minimal differences (corresponding to the rest position) can be identified and removed. This process separates the frames representing actual gestures from the rest position frames.

4.1.2 Denoising

Denoising is a crucial process in signal and Image Processing that aims to remove unwanted noise or disturbances from the data. Noise can manifest as random variations, artifacts, or other undesirable distortions, which can degrade the level and comprehensibility of the underlying information. The primary goal of denoising is to optimize the signal while minimizing the effect of noise, striking a balance between preserving the desired information and eliminating the undesirable components.

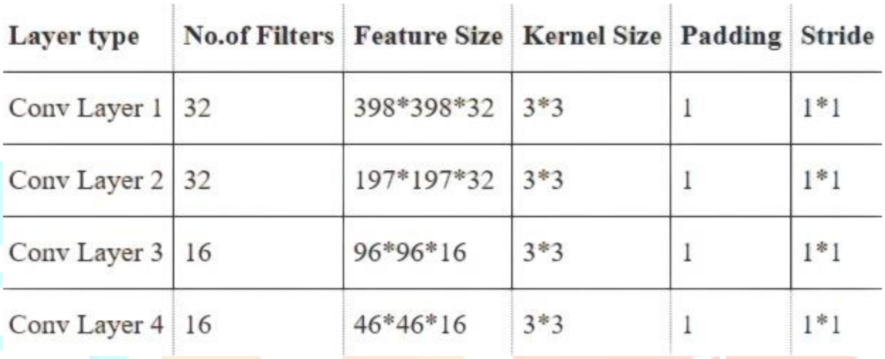
One common denoising technique used in image processing is Gaussian blur. This method applies a smoothing or blurring effect to the image, effectively reducing the impact of high-frequency noise while preserving the overall structure and low-frequency components. By convolving the image with a Gaussian kernel, the algorithm calculates a weighted mean of the surrounding pixels, effectively smoothing out the noise while retaining the essential features of the image. The denoising process is vital for improving the integrity and validity of data, enabling more accurate analysis, processing, and interpretation in various domains.

4.1.3 Training Model

The signer's video, containing a set of continuous gestures, is processed using the RMSE to separate individual gestures. The key frames representing each gesture are then inputted into the trained CNN layers for classification.

The classification process begins by flattening the output from the convolutional layers, converting the three-dimensional array of features into a vector that can be processed by the fully connected layers. The convolutional filters are trained to ascertain the occurrence of specific features or patterns in the input images. Through a set of convolutional layers, the filters learn to derive these features, which are typically represented as smaller matrices compared to the original image dimensions. The fully connected layers, in combination with dropout regularization, are repeated until the feature vector is reduced to output class values. Finally, a softmax function is applied as the last layer, assigning a probability to each gesture class. The class with the top probability is subsequently selected as the output.

Table 1: CNN Architecture



Layer type	No.of Filters	Feature Size	Kernel Size	Padding	Stride
Conv Layer 1	32	398*398*32	3*3	1	1*1
Conv Layer 2	32	197*197*32	3*3	1	1*1
Conv Layer 3	16	96*96*16	3*3	1	1*1
Conv Layer 4	16	46*46*16	3*3	1	1*1

In addition to the CNN, RNN is employed to capture temporal relationships between gestures. RNNs have a "memory" that allows them to consider past insights when making decisions for the current input. This memory enables the RNN to understand the sequential nature of SL gestures. The CNN captures spatial information from individual frames, while the RNN-LSTM (Long Short-Term Memory) processes this information sequentially, understanding the temporal relationships between signs. By combining these two neural network architectures, the system can generate meaningful phrases or sentences in the target spoken language, translating the SL gestures accurately.

4.1.4 Text-to-Speech

The process of transforming text into speech involves utilizing a Text-to-Speech (TTS) system or software. To transforming text into speech, the user typically provides the desired text as input, and the TTS system synthesizes it into audible speech. Many popular platforms and programming languages offer libraries or APIs that facilitate text-to-speech functionality, simplifying the implementation process.

In our project, we have employed the gTTS (Google Text-to-Speech) library, this involves the Python library and command-line interface tool that interfaces with Google Translates text-to-speech API. After processing the input text, the library generates the corresponding audio output. This audio file is then saved and played back in real time, allowing the user to hear the spoken version of the text.

4.2 Speech to SL Conversion

The speech-to-sign conversion process begins by receiving speech input provided by the user through a built-in or external microphone. To perform this operation, the system utilizes a JavaScript library called Speech Recognition.

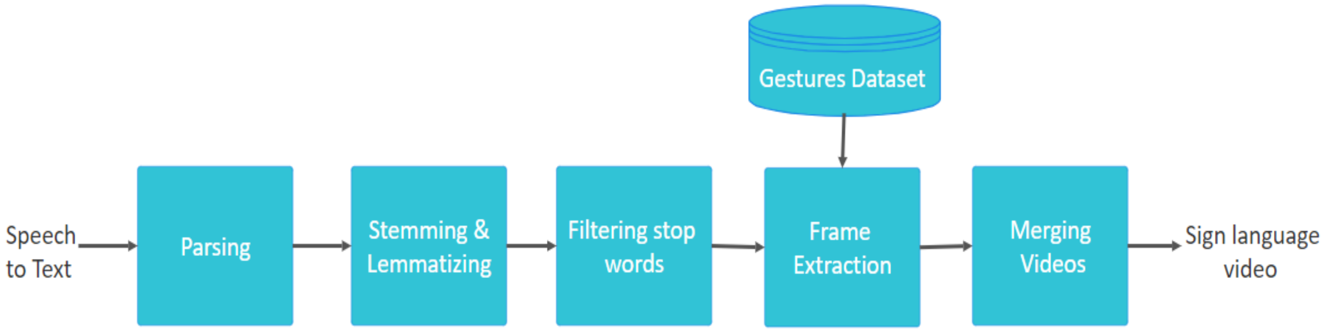


Figure 3: Speech to SL conversion methodology

The various steps used for speech recognition and conversion as per this project are as follows

4.2.1 Parsing

Once the speech input is received, the next step is parsing. In this step, the received text undergoes tokenization, this constitutes the method of breaking down a string into smaller units, such as words, sub words, or characters. These smaller components are known as tokens and serve as the building blocks for natural language processing.

After tokenization, the Stanford Parser is employed to create a structured representation of the text. The parser performs part-of-speech tagging, which assigns grammatical categories (e.g., noun, verb, adjective) to each token. Additionally, it generates a context-free grammar (CFG) representation of the parse tree and produces a dependency representation, which captures the syntactic relationships between the words in the sentence. With the parse tree generated for the input text, a new parse tree is initialized to represent the corresponding sign language syntax. This new parse tree is constructed based on grammatical rules and structure of the target sign language. The process involves mapping the parsed input text to the appropriate sign language representations, ensuring that the final output accurately conveys the intended meaning in the sign language format.

4.2.2 Stemming and Lemmatizing

In Indian Sign Language (ISL), words are typically used in their root form, without suffixes or gerunds. To align with this convention, the system employs stemming, a process that reduces words to their root or base form by removing affixes or suffixes. The Natural Language Toolkit (NLTK) library, a popular tool for natural language processing in Python, provides stemming algorithms that the system utilizes to convert tokens or words to their root forms.

Additionally, the system performs lemmatization, which is the process of grouping together the inflected forms of a term and aligning them with a common base form, known as the lemma. The NLTK package offers the WordNet Lemmatizer, which uses the WordNet corpus as well as the part-of-speech information of every token for determining the correct lemma or primitive form of the word. The lemmatization step helps Enhance the precision of the model by ensuring that words are represented in their standardized root forms, aligning with the conventions of ISL.

4.2.3 Filtering Stop Words

Indian Sign Language (ISL) syntax is concise and does not include stop words. Sentences in ISL are formed using key terms, distinct from function words like auxiliary verbs (e.g., 'am', 'are', 'is', 'be'), prepositions (e.g., 'to', 'for'), and articles (e.g., 'the', 'a', 'an') are omitted.

4.2.4 Frame Extraction

ISL representations do not include inflections, such as gerunds, suffixes, or other word forms like plurals or different tenses. Only the root form of each word is used. The generated ISL sentence is free from conjunctions, articles, and linking verbs, making it more concise and aligned with the conventions of ISL. Once the root words free from stop words are extracted the corresponding frame is extracted from the gesture dataset.

4.2.5 Merging Videos

To generate sign language videos for each word in the sentence, the system utilizes a web-based Indian Sign Language (ISL) dictionary. To automate the method of retrieving these videos, the system employs Seleniuma web automation platform enabling programmatic control of web browsers. Once the individual sign language videos are obtained from the dictionary, they are merged or concatenated into a single video file. This final video is a series of SL videos that represents the spoken sentence in ISL. The concatenation process is facilitated by MoviePy, a Python package designed for video editing and manipulation. MoviePy provides the necessary functionality to combine multiple video clips into a single continuous video stream. After merging the individual sign language videos, the resulting video is presented as the final output, allowing users to view the SL representation of the spoken sentence. By leveraging web automation tools like Selenium and video editing libraries like MoviePy, the system can automate. The method of retrieving and combining sign language videos, enabling a seamless translation from verbal language to SL format.

V. Results and Discussion

The effectiveness of the proposed hybrid CNN-LSTM Evaluation of the model involved a range of metrics. The key results are as follows. The Precision was found to be 0.98, Recall was 0.98, F1-Score was 0.97 and the Accuracy was as high as 98% for the model This analysis offers a perspective on the classification capabilities of the hybrid CNN-LSTM architecture. The high precision and recall indicate the model's ability to accurately identify the target classes. However, the relatively lower F1 score suggests there may be room for improvement in balancing the model's precision and recall. The overall accuracy of 90% demonstrates the model's strong performance on the evaluation dataset.

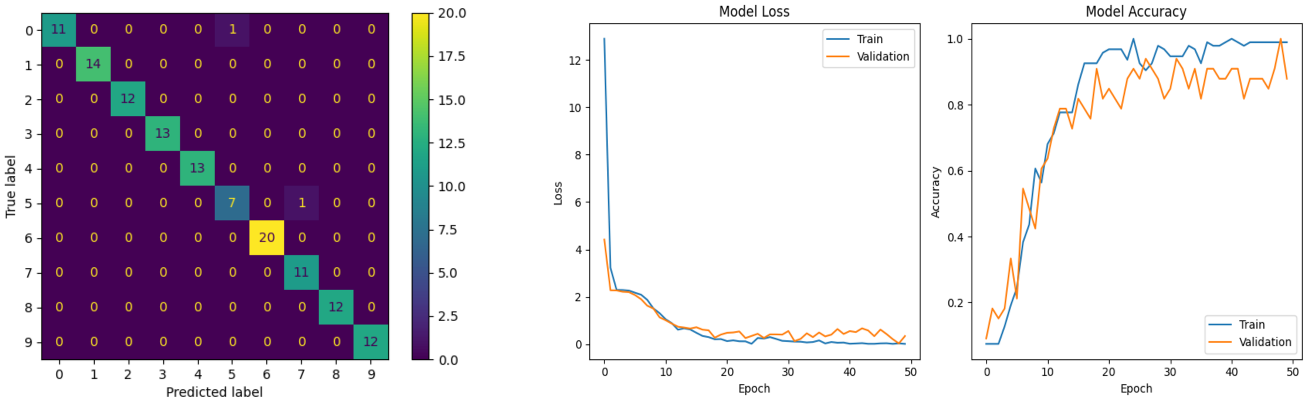


Figure 4: Confusion matrix, model loss, and accuracy curve for the CNN model

The speech to SL conversion system achieved an impressive Precision rate for 94.67%. This means that for a given set of test samples. The system effectively translates 94.67% of the spoken input into their respective SL representations. This level of accuracy represents a significant enhancement compared to earlier methods of speech-to-sign language translation. Traditional methods often struggled to reach accuracy levels above 80-85%. The superior performance of this system can be attributed to its core NLP-driven architecture.

VI. CONCLUSION

In conclusion, the proposed real-time sign language translation system offers significant advancements in facilitating communication and inclusivity between individuals with and without hearing impairments. By employing Advanced methodologies such as CNNs and RNNs to analyze spatial and temporal patterns in sign language, the system accurately translates between sign language and text/speech, In addition to from speech into sign language. This comprehensive approach, combining spatial and sequential information, outperforms traditional algorithms and enables seamless two-way communication. Furthermore, the system's real-time capability minimizes translation delays, enhancing its usability and effectiveness across various social and professional contexts.

Additionally, the system's ability to extract and display gestures from text further augments its utility. By parsing speech input, performing part-of-speech tagging, generating parse trees, and retrieving relevant frames from a gesture dataset based on root words, the system accurately translates Verbal language into SL gestures. This feature augments, not only communication accuracy but also provides a seamless transition between spoken and sign language, making the system superior to other algorithms in terms of functionality and usability.

VII. ACKNOWLEDGMENT

We extend our heartfelt gratitude to Mrs. Pushplata Dubey, Assistant Professor in the Dept of CSE at CITech, for her invaluable guidance and impressive technical insights that significantly contributed to the successful completion of our project. Additionally, we wish to convey our deep appreciation to our friends and teachers who provided assistance in various technical aspects, enriching our project with their expertise and feedback. Lastly, we are grateful for our parents for their unwavering support and encouragement throughout this journey, serving as a constant source of strength and motivation.

REFERENCES

[1] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney and R. Bowden, "Neural Sign Language Translation," 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 2018, pp. 7784-7793, doi: 10.1109/CVPR.2018.00812

[2] 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: 10.1109/ICCCNT.2018.8493808

[3] Ankit Ojha, Ayush Pandey, Shubham Maurya, Abhishek Thakur, Dr. Dayananda P, 2020, Sign Language to Text and Speech Translation in Real Time Using Convolutional Neural Network, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) NCAIT – 2020 (Volume 8 – Issue 15)

[4] Kaliyamoorthi, Manikandan & Patidar, Ayush & Walia, Pallav & Barman Roy, Aneek. (2018). Hand Gesture Detection and Conversion to Speech and Text.

[5] M. Al-Hammadi et al., "Deep Learning-Based Approach for Sign Language Gesture Recognition With Efficient Hand Gesture Representation," in IEEE Access, vol. 8, pp. 192527-192542, 2020, doi: 10.1109/ACCESS.2020.3032140.

[6] Q. Xiao, X. Chang, X. Zhang and X. Liu, "Multi-Information Spatial–Temporal LSTM Fusion Continuous Sign Language Neural Machine Translation," in IEEE Access, vol. 8, pp. 216718-216728, 2020, doi: 10.1109/ACCESS.2020.3039539. Jayanthi P, Ponsy R K Sathia Bhama, B Madhubalasr, “Sign Language Recognition using Deep CNN with Normalised Keyframe Extraction and Prediction using LSTM”, 2023

[7] Sharma, Purushottam & Tulsian, Devesh & Verma, Chaman & Sharma, Pratibha & Nancy, Nancy. (2022). Translating Speech to Indian Sign Language Using Natural Language Processing. Future Internet. 14. 253. 10.3390/fi14090253.

[8] Khalil, Khalid & El-Darymli, Khalid & Khalifa, Othman & Enemosah, Hassan. (2006). Speech to Sign Language Interpreter System (SSLIS).R. Cui, H. Liu, and C. Zhang, "A deep neural framework for continuous sign language recognition by iterative training", 2019.

[9] B., Natarajan & Elangovan, Rajalakshmi & R., Elakkiya & Kotecha, Ketan & Abraham, Ajith & Gabralla, Lubna & Subramaniyaswamy, V. (2022). Development of an End-to-End Deep Learning Framework for Sign Language Recognition, Translation, and Video Generation. IEEE Access. PP. 1-1. 10.1109/ACCESS.2022.3210543.

[10] Chavan, Shruti & Yu, Xinrui & Saniie, Jafar. (2021). Convolutional Neural Network Hand Gesture Recognition for American Sign Language. 188-192. 10.1109/EIT51626.2021.9491897.

[11] R. Ranjan, A. Pandey, S. Gupta, A. Kumar, and P. Gupta, "Real-Time Hand Gesture Recognition for Sign Language Interpretation Using Deep Learning," 2020 International Conference on Communication and Signal Processing (ICCSP), Chennai, India, 2020, pp. 100-104, doi: 10.1109/ICCSP48905.2020.9116486.

[12] Y. Wang, Y. Wu, Y. Yang, and M. Hong, "Sign Language Recognition Using 3D Convolutional Neural Networks," 2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA), Xi'an, China, 2019, pp. 854-859, doi: 10.1109/ICIEA.2019.8833873.

[13] S. Panwar, P. Kumar, and S. Jain, "Real-Time Hand Gesture Recognition for Sign Language Translation using Convolutional Neural Network," 2020 International Conference on Computing, Power and Communication Technologies (GUCON), Greater Noida, India, 2020, pp. 547-552, doi: 10.1109/GUCON47633.2020.9164635.

[14] A. K. Tripathy, A. Sahoo, S. K. Tripathy, and A. Dash, "Sign Language Recognition Using Deep Learning: A Survey," 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN), Mangalore, India, 2021, pp. 255-260, doi: 10.1109/ICACCCN52307.2021.9791096.

[15] S. K. Sahu, S. Sahoo, P. K. Behera, and A. K. Nayak, "Hand Gesture Recognition for Sign Language Translation Using Convolutional Neural Network," 2021 6th International Conference on Computing, Communication and Security (ICCCS), Pune, India, 2021, pp. 1-5, doi: 10.1109/ICCCS50989.2021.9530242.