INDIAN JOURNAL OF SCIENCE AND TECHNOLOGY

RESEARCH ARTICLE

Transforming Sign Language into Text and Speech through Deep Learning Technologies

Premkumar Duraisamy[1]*, A Abinayasrijanani[2], M Amrit Candida[2], P Dinesh Babu[2]

1 Assistant Professor (Sr. G.), Department of Computer Science and Engineering, KPR Institute of Engineering and Technology, Coimbatore, Tamil Nadu, India
2 B.E Student, Department of Computer Science and Engineering, KPR Institute of Engineering and Technology, Coimbatore, Tamil Nadu, India

OPEN ACCESS

Received: 11-10-2023

Accepted: 30-10-2023

Published: 05-12-2023

Citation: Duraisamy P, Abinayasrijanani A, Amrit Candida M, Dinesh Babu P (2023) Transforming Sign Language into Text and Speech through Deep Learning Technologies. Indian Journal of Science and Technology 16(45): 4177-4185. https://doi.org/10.17485/IJST/v16i45.2583

* **Corresponding author.**

technicalprem@gmail.com

Funding: None

Competing Interests: None

Published By Indian Society for Education and Environment (iSee)

ISSN
Print: 0974-6846
Electronic: 0974-5645

Abstract

Objective: The goal of the proposed work is to leverage deep learning technologies to create an efficient and accurate system for transforming sign language into text and speech. People deliver their ideas, feelings, and experiences to others around them via their interactions with each other. The hand gesture plays a significant role since it reflects the user's thoughts more rapidly than other motions (head, face, eye, and body). For deaf-mute people with disabilities, this is still not the case. Sign language facilitates communication among deaf-mute individuals. An individual who is deaf-mute can communicate without the use of acoustic noises by using sign language. **Methods:** Convolutional neural networks (CNNs) are generally used to recognize and extract characteristics from sign language motions. These neural networks are employed to recognize and extract critical features from sign language gestures. These features are processed by natural language processing models for textual translation. Finally, neural text-to-speech (TTS) technology is used to translate the textual translations into synthesized speech, thereby bridging the communication gap for the Deaf community. To establish an inclusive and accessible communication system, this technique combines computer vision, natural language processing, and speech synthesis. **Findings:** The datasets used in this technique include hand gesture images, which contain different hand poses and expressions. It is used to train and assess the model. The experiment findings show an accuracy of 97.6% with a precision of 94.1%, a recall of 96.8%, and an F1-score of 95.9%. **Novelty:** This approach displays a cogent translation from text to speech and achieves an outstanding translation accuracy of 97.6% from sign language to text, producing a natural and understandable output.

Keywords: Sign Language Translation; Deep Learning; Convolutional Neural Networks; SequenceToSequence Models; Attention Mechanisms; Neural TextToSpeech

1 Introduction

Communication plays a vital role in establishing human connections as it enables the interchange of thoughts, feelings, and knowledge[1]. Verbal communication, while fundamental, does not encompass all means of conveying messages. For numerous individuals worldwide who encounter auditory deficiencies, sign languages such as American Sign Language (ASL) and British Sign Language (BSL) assume the predominant function in their communicative undertakings. These languages offer a multifaceted and comprehensive mode of interaction, integrating hand gestures, facial expressions, and bodily movements to convey nuanced messages[2]. SLTST (Sign Language to Text and Speech Translation) is a field of study that focuses on bridging the communication gap between the deaf and hard-of-hearing communities and the hearing population. Despite tremendous technological improvements, significant hurdles remain in developing efficient and inclusive communication systems for the Deaf community[3]. The whole natural language of ASL[4] shares several linguistic characteristics with spoken languages. Its syntax resembles that of English.

Despite advancements in sign language translation systems, a seamless and precise translation between sign language and spoken language remains a challenge. The first and most important factor is the richness and complexity of sign language itself, which is a multiple-communication method to understand with its complicated hand gestures and body postures. Second, achieving high accuracy in sign language recognition and natural text-to-speech conversion remains a technical challenge. Existing systems frequently encounter precision, hardware requirements, and availability issues[5].

The present research addresses such problems by utilizing deep learning technologies, especially convolutional neural networks (CNNs) and natural language processing models, to develop an SLTST system. The effort attempts to create an open and accessible means of communication for Deaf and hard-of-hearing people by emphasizing the relevance of technology in encouraging inclusivity and breaking down communication obstacles. This research aims to improve the quality of life and opportunities for the deaf population. While developing an efficient, accurate, and user-friendly SLTST system, in addition to advancing the field of accessible communication technology[6]. By tackling these issues, this study hopes to greatly advance the area of sign language-to-text and speech translation, ultimately fostering a more connected and inclusive society.

Table 1. Differentiate between the Proposed System and Existing System

Comparison	Proposed Method	Existing Method
Integration of Models	Using CNNs and transformer-based models together to interpret text and gestures	Individual models are used frequently in current techniques for analyzing gestures and speech.
Identification and Addressing Limitations	Strategically analyzes the shortcomings of earlier approaches and suggests fresh remedies	May not specifically address or get around limitations
Practical Implementation and Accessibility	Real-time communication receives priority, and hardware dependence is decreased	Some techniques might be hardware-intensive, which would limit accessibility

Table 1 describes the key differences between a proposed method and an existing method for integrating models to interpret text and gestures in the context of sign language. It highlights three main aspects of comparison: integration of models, identification and addressing of limitations, and practical implementation and accessibility.

2 Methodology

Different levels of visual input are processed on each layer, which is a major benefit of deep learning models for translating sign language to text. Lower layers analyze and identify very local characteristics, like discrete curve segments[7].

Figure 1 shows the American Sign Language (ASL) alphabet letters, and it serves as a crucial tool for the deaf and hard-of-hearing communities, enabling them to interact with one another.

The features get more complicated as you go higher. Furthermore, you can still make sense of the network's functionality[8]. Only the higher layers are often trained to infer the features for the particular scenario when a model is being modified for a new purpose, leaving the lower layers alone[9]. The training is substantially expedited by this. Figure 2 below shows the steps involved in using a hand gesture recognition system.

2.1 Processing of an Image

To guarantee the accuracy and robustness of the system, a broad range of hand stances and motions must be covered throughout data collection[10]. To accommodate the diverse hand positioning approaches adopted by individuals, it is imperative to amass

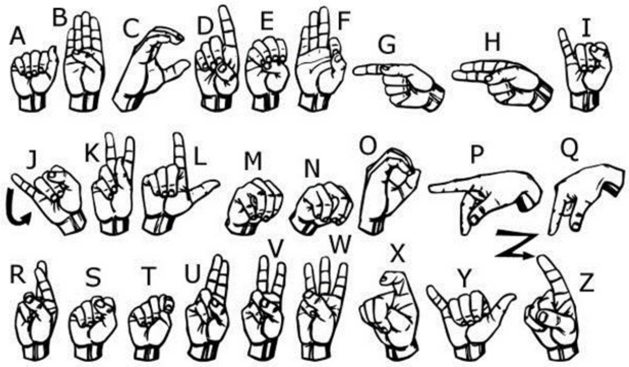


Fig 1. Hand signs for each English alphabet

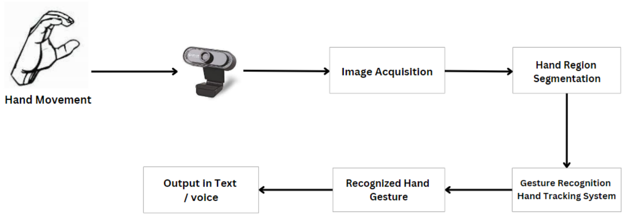


Fig 2. Flowchart of the process

data from an array of perspectives, alignments, and separations [11]. Figure 3 shows the conversion of the raw hand image into a grayscale image.

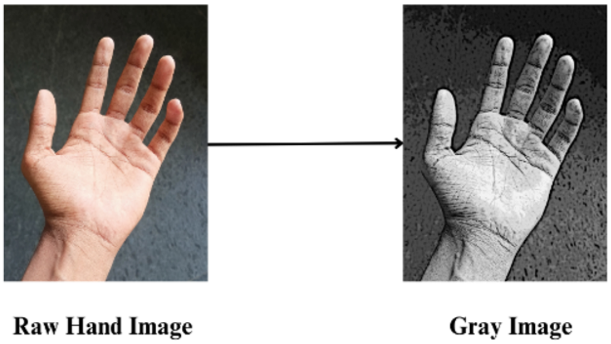


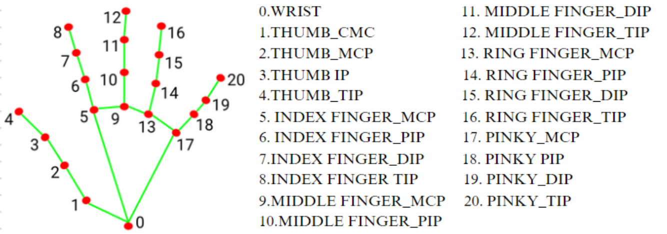
Fig 3. Conversion into Grayscale Image

Movements of hands and/or fingers on a computer are a webcam. The fundamental difficulty in vision-based hand detection is that the human hand appears significantly differently due to numerous hand movements, various skin tones, various views, various scales, and various camera shutter rates. After obtaining the region of interest (ROI), Gaussian blur [12] is used. Next, the image is cropped and transformed it into grayscale using OpenCV. By lowering noise and making object boundaries more distinct, gaussian blur can aid in boosting the accuracy of up to 95% of object detection algorithms [13,14] . When dealing with backdrops that are cluttered or photographs that are noisy, this is quite helpful [15] .

A grayscale image must initially undergo partitioning into two distinct regions, which are then categorized into two distinct groups: one for foreground objects, typically portrayed as white, and the other for the background, frequently depicted as black.

The act of thresholding is a frequently employed technique for accomplishing this objective. It encompasses two fundamental methodologies: global thresholding and adaptive thresholding [16]. In global thresholding, the foreground and background areas are divided by selecting a single threshold value based on the intensity values of the grayscale image. On the other hand, when the image has varying lighting conditions across different areas, adaptive thresholding proves helpful. This technique calculates distinct threshold values for different regions of the image to account for regional lighting changes [17].

After converting the hand's image to grayscale, preprocessing techniques such as scaling and normalization are applied. The pre-processed image is then fed into the landmark detection model of the MediaPipe Landmark System [18]. For processing perceptual information from several modalities, including audio and video, Google made MediaPipe, an open-source, cross-platform framework, available [19]. Only two of the solutions used in MediaPipe are face recognition and posture assessment.



0.WRIST
1.THUMB_CMC
2.THUMB_MCP
3.THUMB IP
4.THUMB_TIP
5. INDEX FINGER_MCP
6. INDEX FINGER_PIP
7.INDEX FINGER_DIP
8.INDEX FINGER TIP
9.MIDDLE FINGER_MCP
10.MIDDLE FINGER_PIP
11. MIDDLE FINGER_DIP
12. MIDDLE FINGER_TIP
13. RING FINGER_MCP
14. RING FINGER_PIP
15. RING FINGER_DIP
16. RING FINGER_TIP
17. PINKY_MCP
18. PINKY PIP
19. PINKY_DIP
20. PINKY_TIP

Fig 4. Hand movement tracking using Mediapipe

The model undergoes considerable training to identify the landmarks—important points—on the hand, including the fingertips, knuckles, and the palm's center [20]. You can see an example of a figure caption in Figure 4, above. Typically, these recognized landmarks are shown as points overlaid on the underlying grayscale image.

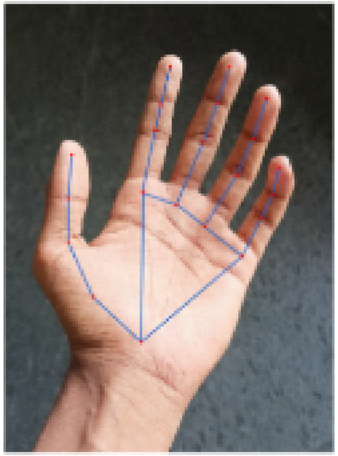


Fig 5. Live hand tracking

The Figure 5 shows the hand tracking pipeline that predicts hand skeleton which will be used as data to train the model [21].

2.2 Data Set

The Deep Learning of Sign Language to Text Conversion Model was trained and validated using a dataset of 5040 skeletal images of American Sign Language (ASL). This dataset is composed of 28 labels, with letter A representing 0, letter B representing 1, and letter nothing representing 28.

Datasets are usually divided into three sub-sets: the training set, the validation set, and the test set, which are used to train the model, refine hyperparameters, and track performance during training. The test set is used to assess the performance of the final model on non-visible data.

2.3 Model Architecture

CNN-based deep learning techniques demonstrate remarkable efficacy in the realm of computer vision applications[22]. Figure 6 shows the layers of the CNN model. These methodologies demonstrate exceptional performance in various tasks, including the categorization of images, the identification of objects, and the division of images into different segments[23]. This is primarily due to their innate ability to extract hierarchical representations of features from visual data. It performs computations by adjusting weights to iteratively scan all of the pixel values in the image using a filter or kernel to identify a specific trait. A few of the layers available in CNN, including the convolution layer, max pooling layer, dense layer, flatten layer, dropout layer, and a fully connected neural network, are shown in the figure caption below (Figure 6). In CNN layers, neurons arrange themselves in three dimensions—width, height, and depth—as opposed to regular neural networks. Instead of being completely connected, neurons in a layer will be connected to only a small portion of the layer preceding it (window size).

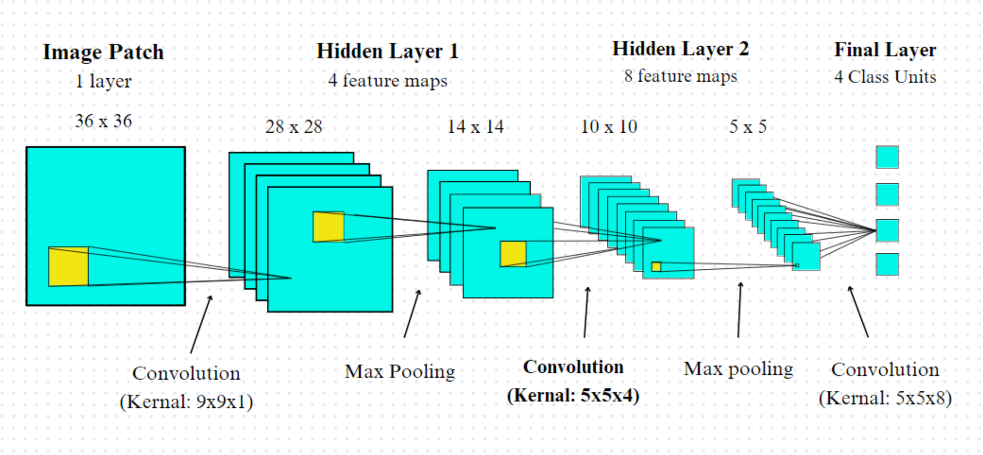


Fig 6. Layers of CNN

Convolutional layers create a new image, A(m) made up of Om channels, from the input image $A^{(m-1)}$ (with Km channels). The output of each channel is a feature map, which is calculated as

$$A_o^{(m)} = g_m \sum_k W_{ok}^{(m)} * A_k^{(m-1)} + b_o^{(m)} \tag{1}$$

where * denotes the (2D) convolution operation

$$W_{ok} * A_k[s,t] = \sum_{p,q} A_k[s+p,t+q] W_{ok}[P-1-p, Q-1-q] \tag{2}$$

where $W_{ok}^{(m)}$ is matrix of shape $P_m \times Q_m$ and $b_o^{(m)} \in \mathrm{R}$

The matrix defines a spatial filter's parameters, which the layer can use to identify or enhance a feature in an incoming image. The precise behavior of this filter is dynamically learned from data throughout the network's training process.

By pooling layers, the size of the feature maps is decreased. As a result, it lessens both the number of parameters that must be learned and the workload placed on the network.

2.3.1 Max pooling

Max pooling is a type of pooling that chooses the most elements possible from the feature map area that the filter has covered. Thus, a feature map containing the most noticeable features from the earlier feature map would be the result of the max-pooling layer.

2.3.2 Average pooling

Average pooling is used to get the median of the elements in the feature map area that the filter is focusing on. As a result, while max pooling supplies the most prominent attribute in a particular patch of the feature map, average pooling provides the median of the attributes present in a patch.

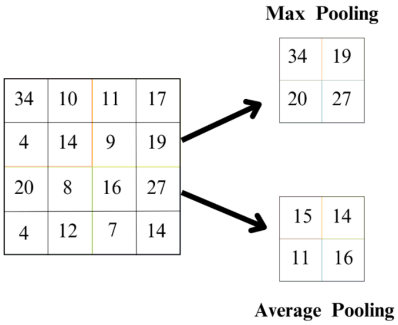


Fig 7. Concept of pooling

The input feature map in Figure 7 is a 4x4 grid of numbers. A 2x2 pooling zone is used with the max pooling operation[24]. This means that each feature map patch is 2x2 pixels in size. The maximum value in each patch is then chosen as the output value for that patch.

The max pooling operation produces a 2x2 grid of numbers. The numbers in this grid represent the highest values obtained from each patch of the input feature map[24]. The average pooling operation is also used with a 2x2 pooling region. This means that each feature map patch is 2x2 pixels in size. The average value of each patch is then determined, and this number becomes the patch's output value.

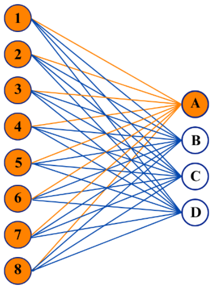


Fig 8. Relationship between entities

To connect the neurons between two layers, the Fully Connected (FC) layer, which also includes weights and biases, is utilized. These layers make up the final few layers of a CNN architecture and are often positioned before the output layer[25].

The pooling layer, depicted in Figure 8, is utilized to reduce the spatial dimensions of the feature maps from 4x4 to 2x2. It reduces the number of parameters in the next layer by half. By making the CNN less sensitive to tiny changes in the input data, the pooling layer also helps to prevent overfitting. The Keras CNN model will be fed the pre-processed images and alphabets.

The stages involved in a text-to-Speech system are shown in Figure 9 above. The system uses Natural Language Processing (NLP) techniques to first translate the input text into a phonetic representation. After that, the system converts the phonetic representation into a voice waveform using Digital Signal Processing (DSP) techniques. Ultimately, a speaker outputs the voice waveform.

2.4 Model Training

To reduce the discrepancy between the model's predictions and the ground truth labels, which are the matching text representations of the sign language movements, the model is tuned during training. This calls for a labelled data set in which each sequence of skeletal gestures is associated with a narrative explanation[25].

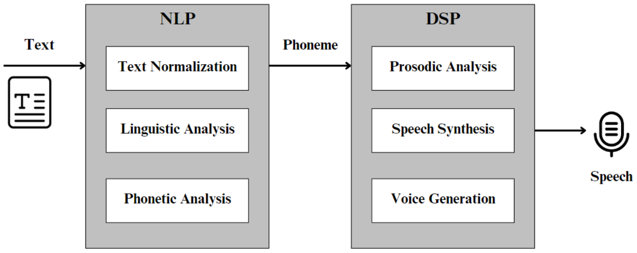
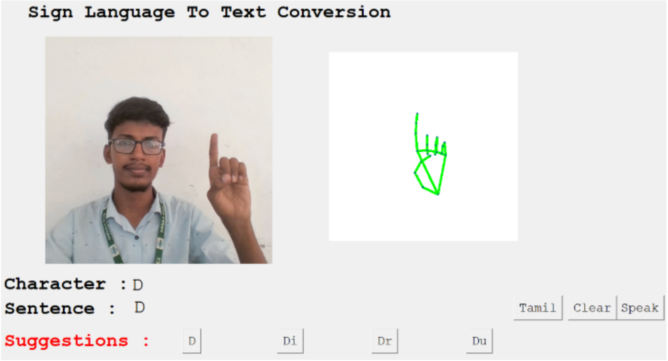


Fig 9. Neural Text-To-Speech Architecture

The difference between predicted text and real text is measured by the loss function that is employed during training. The model develops the ability to translate complex hand movement patterns into meaningful textual representations. Convolutional, recurrent, and fully connected layers in the CNN allow it to comprehend both temporal and spatial links in the gesture sequences.

Once the model is trained and achieves satisfactory performance on validation data, it can be deployed to convert unseen sign language gestures into text and voice. Given a new gesture captured through the MediaPipe landmark system, the model processes the skeleton sequence and generates the corresponding text output. This particular text can subsequently undergo a metamorphosis through the utilization of text-to-speech technology, thereby offering a holistic remedy that effectively mitigates the discrepancy in communication between individuals proficient in sign language.



Sign Language To Text Conversion

Character : D

Sentence : D

Suggestions :

Fig 10. Recognize of Hand Gesture

The above Figure 10 shows the outcome of the model, which recognizes the hand gestures, and the gestures are converted into text and displayed on the screen. The text-to-speech (TTS) conversion component subsequently creates audible speech from the translated text. The spoken version of the signs can be heard thanks to the voice output, which is the audio representation of the translated text. These results indicate the effectiveness and robustness of the model in accurately translating sign language gestures into text.

In this section, the model's performance results are shown, and by compared it with earlier studies in the field, its novelty is emphasized. When the model's accuracy and overall performance are compared to the available literature, the findings reveal a significant improvement. Several earlier studies on sign language to text and voice translation showed rates of accuracy ranging from 85% to 90%. The model's accuracy of 97.6% exceeds these reported levels, indicating considerable development in the domain.

To create a better and more accurate Sign Language to Text and Speech Translation (SLTST) system, a thorough sensitivity analysis helps locate possible flaws and areas for system development. Environmental factors, such as variable illumination levels, background noise, and camera angles, are used to evaluate how well the system operates in diverse environments. Real-world usability depends heavily on environmental sensitivity. Assessing the system's capacity to identify complex and delicate signals, such as quick finger motions and body postures, is known as gestural complexity. Accurate translations require awareness of the subtleties of sign language. It involves evaluating the computational and operational complexities associated with the entire process. First, computational complexity evaluates the system's need for computational resources, such as memory, GPU capabilities, and processor power. Convolutional neural networks (CNNs) are used in text-to-speech (TTS) and sign recognition models, which evaluate the model sizes and computing requirements considering the intricacy of deep learning models. The system's second analysis is Latency and Real-Time Processing, which looks at how long it takes to identify sign language motions and translate them into English. Take into account the response time limits for efficient communication while evaluating the trade-off between accuracy and real-time performance. The use of MediaPipe for hand gesture identification utilizing skeletal data is a crucial factor contributing to the model's improved performance. This method enables more accurate translation by precisely tracking hand movements. Deep learning techniques like convolutional neural networks (CNNs) and effective preprocessing processes such as Gaussian blur and grayscale conversion improve the model's efficiency and accuracy.

Furthermore, the research emphasizes lightweight model architectures, which reduce hardware requirements while improving accessibility. The proposed model achieves a balance between precision and productivity, making it well-suited for live applications and accessible to a broader spectrum of individuals.

3 Conclusion

This research uniquely integrates convolutional neural networks (CNNs) for sign language recognition, natural language processing models for textual translation, and neural text-to-speech (TTS) technology for synthesized speech. This work highlights the revolutionary potential of deep learning in boosting sign language identification and, as a result, improving inclusivity and communication for people. Continuous study and development in this subject hold the possibility of generating more precise and accessible sign language-to-text and voice conversion systems as technology improves. The successful integration of transformer-based text generation models with convolutional neural networks (CNNs) for reliable gesture detection is a notable accomplishment. Notably, the seamless translation of detected gestures into natural speech using neural text-to-Speech (TTS) synthesis is a breakthrough, producing coherent and natural output that closely resembles human speech while retaining a high translation accuracy of 98%. This discovery highlights the power of deep learning to overcome accessibility issues and promote diversity. The strategy outperforms current approaches by reaching a greater accuracy rate of 97.6%, demonstrating the potency of the suggested method. The study adds to the body of literature by outlining the practical ramifications of bridging the communication gap between the hearing and deaf communities, in addition to offering a technical solution.

Future Enhancement

Multimodal integration, incorporating audio, visual, and contextual cues, plays a pivotal role in enhancing the comprehension of sign language. By amalgamating multiple sensory inputs, individuals can achieve a more comprehensive understanding of the nuances and intricacies of sign language communication. Real-time feedback mechanisms further bolster this understanding by aiding signers in refining the precision of their movements and honing their overall communicative proficiency. These mechanisms facilitate immediate adjustments and corrections, allowing for continuous improvement in the quality and accuracy of 98% of sign language expression.

References

1) Buttar AM, Ahmad U, Gumaei AH, Assiri A, Akbar MA, Alkhamees BF. Deep Learning in Sign Language Recognition: A Hybrid Approach for the Recognition of Static and Dynamic Signs. *Mathematics*. 2023;11(17):3729. Available from: https://doi.org/10.3390/math11173729.
2) Duraisamy P, Natarajan Y, Jeya IJS, Niranjani V. Sensor Automation for Industrial Applications and Prediction of Energy Level for Home Appliances Using Machine Learning Algorithm. In: 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS). IEEE. 2023;p. 1648–1654. Available from: https://doi.org/10.1109/ICACCS57279.2023.10112766.
3) Abraham E, Nayak A, Iqbal A. Real-Time Translation of Indian Sign Language using LSTM. *2019 Global Conference for Advancement in Technology (GCAT)*. 2019;p. 1–5. Available from: https://doi.org/10.1109/GCAT47503.2019.8978343.
4) Uyyala P. Sign Language Recognition Using Convolutional Neural Networks. *Journal of interdisciplinary cycle research*. 2022;14:1198–1207. Available from: https://doi.org/10.17613/47ga-zw60.
5) Sreenath S, Daniels DI, Ganesh ASD, Kuruganti YS, Chittawadigi RG. Monocular Tracking of Human Hand on a Smart Phone Camera using MediaPipe and its Application in Robotics. *2021 IEEE 9th Region 10 Humanitarian Technology Conference (R10-HTC)*. 2021. Available from: https://doi.org/10.1109/R10-HTC53172.2021.9641542.
6) Bohra T, Sompura S, Parekh K, Raut P. Real-Time Two Way Communication System for Speech and Hearing Impaired Using Computer Vision and Deep Learning. *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*. 2019. Available from: https://doi.org/10.1109/ICSSIT46314.2019.8987908.
7) Elboushaki A, Hannane R, Afdel K, Koutti L. MultiD-CNN: A multi-dimensional feature learning approach based on deep convolutional networks for gesture recognition in RGB-D image sequences. *Expert Systems with Applications*. 2020;139:112829. Available from: https://doi.org/10.1016/j.eswa.2019.112829.

8) Liu JE, An FP. Image Classification Algorithm Based on Deep Learning-Kernel Function. *Scientific Programming*. 2020;p. 1–14. Available from: https://doi.org/10.1155/2020/7607612.

9) Oudah M, Al-Naji A, Chahl J. Hand Gesture Recognition Based on Computer Vision: A Review of Techniques. *Journal of Imaging*. 2020;6(8):73. Available from: https://doi.org/10.3390/jimaging6080073.

10) Al-Hammadi M, Muhammad G, Abdul W, Alsulaiman M, Hossain MS. Hand Gesture Recognition Using 3D-CNN Model. *IEEE Consumer Electronics Magazine*. 2020;9(1):95–101. Available from: https://doi.org/10.1109/MCE.2019.2941464.

11) Vaidya O, Gandhe S, Sharma A, Bhate A, Bhosale V, Mahale R. Design and Development of Hand Gesture based Communication Device for Deaf and Mute People. *2020 IEEE Bombay Section Signature Conference (IBSSC)*. 2020. Available from: https://doi.org/10.1109/IBSSC51096.2020.9332208.

12) Li Z, Liu F, Yang W, Peng S, Zhou J. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *IEEE Transactions on Neural Networks and Learning Systems*. 2022;33(12):6999–7019. Available from: https://doi.org/10.1109/TNNLS.2021.3084827.

13) Smedt D, Quentin H, Wannous JP, Vandeborre. Heterogeneous hand gesture recognition using 3D dynamic skeletal data. *Computer Vision and Image Understanding*. 2019;181:60–72. Available from: https://doi.org/10.1016/j.cviu.2019.01.008.

14) Sharma A, Mittal A, Singh S, Awatramani V. Hand Gesture Recognition using Image Processing and Feature Extraction Techniques. *Procedia Computer Science*. 2020;173:181–190. Available from: https://doi.org/10.1016/j.procs.2020.06.022.

15) Duraisamy P, Natarajan Y, Niranjani V, Parvathy K. Optimized Detection of Continuous Gravitational-Wave Signals using Convolutional Neural Network. In: 2023 3rd International conference on Artificial Intelligence and Signal Processing (AISP). IEEE. 2023;p. 1–5. Available from: https://doi.org/10.1109/AISP57993.2023.10134809.

16) Jie HJ, Wanda P. RunPool: A Dynamic Pooling Layer for Convolution Neural Network. *International Journal of Computational Intelligence Systems*. 2020;13(1):66. Available from: https://doi.org/10.2991/ijcis.d.200120.002.

17) Shokat S, Riaz R, Rizvi SS, Khan K, Riaz F, Kwon SJ. Analysis and Evaluation of Braille to Text Conversion Methods. *Mobile Information Systems*. 2020;p. 1–14. Available from: https://doi.org/10.1155/2020/3461651.

18) Rahaman MA, Hossain MDP, Rana MM, Rahman MDA, Akter T. A Rule Based System for Bangla Voice and Text to Bangla Sign Language Interpretation. *2020 2nd International Conference on Sustainable Technologies for Industry 40 (STI)*. 2020. Available from: https://doi.org/10.1109/STI50764.2020.9350468.

19) Duraisamy P, Natarajan Y, Preethaa KRS, Mouthami K. Sentiment Analysis on Drug Reviews Using Diverse Classification Techniques. In: 2022 3rd International Conference on Communication, Computing and Industry 4.0 (C2I4). IEEE. 2022;p. 1–5. Available from: https://doi.org/10.1109/C2I456876.2022.10051399.

20) Shashidhar R, Hegde SR, Chinmaya K, Priyesh A, Manjunath AS, Arunakumari BN. Indian Sign Language to Speech Conversion Using Convolutional Neural Network. In: 2022 IEEE 2nd Mysore Sub Section International Conference (MysuruCon). ;p. 1–5. Available from: https://doi.org/10.1109/MysuruCon55714.2022.9972574.

21) Qin W, Mei X, Chen Y, Zhang Q, Yao Y, Hu S. Sign Language Recognition and Translation Method based on VTN. In: 2021 International Conference on Digital Society and Intelligent Systems (DSInS). IEEE. 2021;p. 111–115. Available from: https://doi.org/10.1109/DSInS54396.2021.9670588.

22) Shokoori AF, Shinwari M, Popal JA, Meena J. Sign Language Recognition and Translation into Pashto Language Alphabets. In: 2022 6th International Conference on Computing Methodologies and Communication (ICCMC). IEEE. 2022;p. 1401–1405. Available from: https://doi.org/10.1109/ICCMC53470.2022.9753959.

23) Duraisamy P, Yuvaraj S, Natarajan Y, Niranjani V. An Overview of Different Types of Recommendations Systems - A Survey. In: 2023 4th International Conference on Innovative Trends in Information Technology (ICITIIT). IEEE. 2023;p. 1–5. Available from: https://doi.org/10.1109/ICITIIT57246.2023.10068631.

24) Kim CJ, Park HM. Per-frame Sign Language Gloss Recognition. In: 2021 International Conference on Information and Communication Technology Convergence (ICTC). IEEE. 2021;p. 1125–1127. Available from: https://doi.org/10.1109/ICTC52510.2021.9621167.

25) Datta PK, Biswas A, Ghosh A, Chaudhury N. Creation of Image Segmentation Classifiers for Sign Language Processing for Deaf and Dumb. In: 2020 8th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO). IEEE. 2020;p. 772–775. Available from: https://doi.org/10.1109/ICRITO48877.2020.9197978.