Advancements in Vision-Based Sign Language Recognition: A Comprehensive Review

Dinushika Chithrani, Adithya Rajapakshe, Dhanuka Jayasinghe, Umaya Balagalla,
Bhathiya Pilanawithana, Uditha Wijewardhana, Udaya Wijenayake
Faculty of Engineering, University of Sri Jayewardenepura, Sri Lanka
e-mail: dinushika219@gmail.com, hiruni.rajapaksha@gmail.com, dsankalpa247@gmail.com, umayabalagalla@sjp.ac.lk,
bpilanawithana@sjp.ac.lk, uditha@sjp.ac.lk, udayaw@sjp.ac.lk

***Abstract*—As nearly 70 million individuals suffer from disabling hearing loss, sign language serves as an important means of communication. Unfortunately, the lack of proficiency in sign language among the general population hinders meaningful interactions with those who rely on it. This paper presents an extensive analysis of the cutting-edge methodologies in sign language translation, with the ultimate goal of facilitating effective communication between sign language users and the broader community. In addition to reviewing state-of-the-art approaches, this work also investigates into the challenges and limitations faced by gesture recognition research. Overall, it is expected that the study may provide readers and researchers with a guide for future research and creation in the field of sign language recognition.**

***Index Terms*—Sign Language, Gesture recognition, Computer Vision, Machine Learning, Image Processing**

I. Introduction

Communication plays a vital role in the human experience, serving as a fundamental and efficient means of expressing thoughts, emotions, and viewpoints. Nevertheless, a significant portion of the global population lacks this ability. Many individuals suffer from hearing loss, speech impairments, or both. These conditions are among the most prevalent disabilities worldwide. Consequently, there is a pressing need to eliminate the communication barriers that have a huge impact on the lives and social interactions of deaf-mute individuals [1].

According to the World Federation of the Deaf, deaf and mute people use over 300 sign languages worldwide to communicate among themselves [2]. To overcome the communication barriers between the hearing impaired and normal humans, Sign Language Recognition (SLR) serves as a key method. Different methods and algorithms are developed to identify signs and understand their meanings using collaborative research areas involving pattern matching, computer vision, natural language processing, and linguistics.

Sign language involves using various body parts, namely fingers, hands, arms, head, body and facial expressions to convey information [3]. Hand gestures rely on five key parameters: hand shape, palm orientation, movement, location, and expression/non-manual signals [4]. Accurate sign language communication requires all these five parameters to be performed correctly. Two forms of hand gestures, static and dynamic, are used in sign language. Static gestures only include hand poses, while dynamic gestures include hand movements [5].

Recognizing hand gestures can be accomplished through a vision-based or sensor-based approach. Vision-based methods involve capturing images or video of hand gestures using a camera, while sensor-based methods use sensor instruments to capture the motion, position, and velocity of the hand.

This paper intends to focus on reviewing the latest developments in Vision-based Sign Language Recognition. The rest of the paper is organized as follows: Section 1 describes the identified problem. Section 2 discusses the techniques used in vision-based sign language recognition, covering data acquisition, pre-processing, sign detection and feature extraction, and recognition/classification methods, which are shown in figure 1. Image acquisition is the first stage in this process of acquiring sign images or video frames. The second stage is pre-processing to eliminate unwanted noise, enhance image quality, and segment the meaningful regions. Sign detection and feature extraction are techniques to obtain the sign gestures from the image and transform the input raw data into numerical features. The recognition and classification step describes the process of identifying the sign using different techniques [1]. In section 3, the findings of previous research are discussed and summarized.

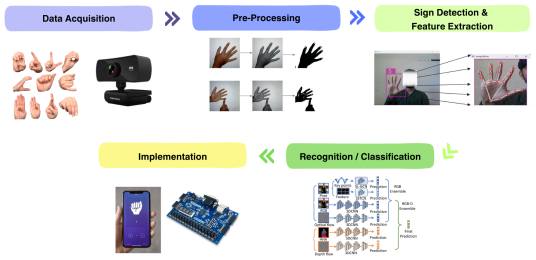

Fig. 1. Vision-based gesture recognition stages

II. Problem Statement

Despite the importance of sign language recognition systems, there is a lack of an updated Literature Survey. The need for a review is to identify, categorize, and analyze existing

research in sign language recognition. Although there has been a substantial increase in research papers related to sign language recognition over the past decade, there is a need to summarize all the techniques used. Due to the importance of sign language recognition in the deaf and speaking disabled community, the need for a review is also to identify research gaps and future trends. Even though there is a significant improvement in the recognition accuracy of proposed modes in sign language recognition, some challenges still need to be addressed.

III. Literature review

The strategy we have followed for this Literature Review includes the process of gesture recognition, which can be categorized into a few stages in general. Based on these stages, the review of multiple research papers is discussed in this section.

A. Data acquisition

In the context of vision-based gesture recognition, the collected data consists of frames of videos. These systems take input using different image-capturing devices such as regular video cameras, webcams, stereo cameras, thermal cameras, or more advanced technologies such as Kinect and LMC. Researchers capture videos and convert them into frames, while some only capture images for static sign recognition.

Webcams are one of the commonly used techniques to capture images and videos. In [6], Hettiarachchi and Meegama process 200 frames and consider the final frame. Similarly, research [7] used a webcam, and they were able to capture 30 fps with a 0.7-megapixel resolution, and [8] used a 15-fps frame rate. Since some researchers have implemented mobile applications, they have used the phone camera to capture data [9]. In research, [10] Wimalaratne and Fernando used a Kinect XBOX 360 device to capture some additional information as the depth of an image. To get more information, research [11] has used a CMOS image sensor camera. In some cases, [12], [13] used digital cameras to input high-resolution images. Rishan et al. [14] used a Leap Motion Controller to capture the input and get the 3-axis information of the hand gesture.

After the data-capturing stage [15], Dhanawansa and Rajakaruna used a method to identify the important frames from the captured frame sequence. They used a 3-frame difference method to extract signs from the frame sequence. It puts three adjacent frames as a group and subtracts both adjacent frames to identify the hand poses from the video.

B. Image Pre-Processing

Image pre-processing stages are conducted to enhance the overall system's performance by modifying the input data to the model.

Since the input data of a system may have varying heights, hand lengths, and other differences, it's essential to normalize the data properly to ensure that the final output remains unaffected by these factors. The input images were converted into a fixed size in [9], [16], [17], [11] and [18]. In some research articles, [10], [19] images were normalized at the pre-processing stage. Background subtraction is also a main method used in this step. In research [10], they used a filtering method to remove background and unwanted skeleton points.

Skin colour segmentation is mostly performed in RGB, YCbCr, HSV and HSI colour spaces. It is the process of isolating and delineating an image's portion corresponding to the human skin. In research [15], they used skin segmentation and subsequent filtering to isolate the critical regions of the skin within the field of view using the YCbCr colour scheme. The features of the resulting binary mask were preserved through the application of morphological transforms. These morphological operations include techniques such as erosion, dilation, opening and closing. The median blue filtering method was applied to get an effective smoothened edge of the mask. Finally, larger contours were extracted, eliminating the background noise.

In research [12] and [20], hand segmentation was done using a thresholding method to convert RGB images into grayscale. Subsequently, Morphological filtering was done to repair and smooth the segmented image. Similar to [15], Peiris [7] used a skin detection method where they used YCbCr as the colour scheme of choice to detect skin rather than RGB. They created a mask to identify the outlines of the hand in the next stage. Gupta et al. [11] used an illumination compensation method using RGB mean values to get good quality images to ignore the light source changes with changes in the environment conditions. Following with skin colour segmentation to analyze the skin colour pixels of the hand in YCbCr colour space using the Otsu method. Next, morphological filtering was used to out the noisy pixels.

Extracted images are converted into grayscale images in research [21]. They used a thresholding method to perform edge detection correctly with dilation and erosion operations to make the detected edges sharper. Next, they extract the largest contour and consider it as the person and the rest of the image as the background. During skin segmentation, the parts, including the face and neck, can also be detected with the hands. Therefore, in [8], Athira et al. first used a face detection and elimination method. They have also used hand segmentation and noise removal techniques afterwards. Data augmentation techniques were used in the pre-processing stage of the research [22] to increase the amount of data input to the system.

C. Sign Detection and Feature Extraction

Sign detection helps to identify the signs since it specifies movements and poses made by a person's hands, body, or face within a defined Region of Interest (ROI). Feature extraction refers to converting raw data into numerical values while preserving the information in the original dataset.

Both [23] and [19] adopt a similar approach, using MediaPipe for real-time gesture detection. MediaPipe Holistic is utilized to detect and track key points, allowing to recognize hand and body movements in sign language gestures. In [10], Pumudu and Prasad developed a gesture pre-processing

module to extract the feature points. Data extracted from each frame formed a "feature frame", representing the X, Y, and Z coordinates of selected joint positions of the skeleton.

In [15], Dhanawansa and Rajakaruna utilized a combination of hand detection and tracking algorithms, binary masking, and template matching to identify the gestures in the image effectively. Dissanayake et al. [21] performed edge detection using a thresholding method. The accuracy of identifying the edges of sign language gestures is guaranteed by this method.

Kuo et al. [20] employed an algorithm based on RGB values in the image frame for hand gesture detection. The height and width of the white region (hand) and centroid of the region were determined here. In [13], images are being compressed from 4096-pixel values into a 16-feature vector. It uses the distance between the image's centre and the maximum location of each quadrant of the edge image calculated from the pre-processed image.

In [11], four different shape-based features are calculated: Area of hand, Perimeter of hand, Thumb detection, Radial profile and angular position. In [12], Guerrero et al. also used a shape-based feature extraction approach. Here, the shape of the hand from the binary image of the hand (background removed) is transformed into 2 vectors, which contain the number of pixels per row and column of the object. From these vectors, 2 new fixed-size vectors are formed and concatenated to form a single feature vector for feeding into the MLP Neural Network.

In [9], Dahanayaka et al. focused on achieving real-time object detection. They accomplished this by developing a Single-Shot Detector (SSD) VGG16 model designed for swift object detection. The architecture of the model adds six auxiliary convolutional layers to improve object detection performance and uses the VGG16 network to extract feature maps. In [6], they have used 2D CNN starting with a 128 x 128 image containing three colour channels. The convolutional layer transforms it into a 126 x 126 image with three colour channels. Then, they used a progressively larger filter size while adding multiple convolutional layers to enhance feature extraction.

In [24], Yanqiu Liao et al. used a B3D ResNet model for obtaining short-term spatio-temporal features. Here, video sequences are transmitted into the B3D ResNet model after localizing the hand position in the video frames to obtain the features. In [18], Junfu et al. used a 3D-ResNet to obtain fixed-length features. The input video frames representing the combination of different sign gestures are converted into a set of ordered video clips using a sliding window on the image sequence before sending it to the 3D-ResNet model.

In [8], a feature vector of dimension 6 is extracted from the frames related to each dynamic frame identified after performing key frame extraction (based on hand position and shape changes) and co-articulation elimination (based on the motion trajectory and acceleration). The feature vector consists of Hand shape, Trajectory length, average speed, number of significant curves, the number of points of minima, and palm orientation. For the static gestures, a feature vector consists of only hand shape, average speed, and palm orientation is extracted.

D. Recognition / Classification

In this stage, the label of the input gesture is predicted by sending the extracted features from the input through the trained model. Basic algorithms and different machine learning techniques, like Neural Networks, Dynamic Time Wrapping (DTW), Support Vector Machines (SVMs), and Nearest Neighbors, are used for the recognition/classification process.

1) Neural Networks:

When considering sign language translation, neural networks like ANN, CNN, MLP, and LSTM are used. In CNN, there are different layers: Convolutional Layers with ReLU, Pooling Layers, Flatten Layers, Fully connected (FC) layers, Softmax, and Output layer.

Zhang et al. [17] adopted a simple Convolutional Neural Network (CNN) with two Convolutional layers, two fully connected layers and ReLU as the activation function. Classification of the gesture takes place from the fully connected layers where each input static gesture is classified individually, and a single word output is generated. In the proposed method, the first fully connected layer has 180 neurons, and the last layer has 6 neurons, equal to the number of classes. This method has achieved an accuracy of 86.3% for the selected static word signs.

Guerrero et al. [12] employed an Artificial Neural Network (ANN) with fully connected layers (MLP) for the classification of static hand gestures with an accuracy of 98.15%. Here, each gesture is classified as letters individually using the shape based features of the hand extracted from the feature extraction stage. The proposed network comprises an input layer with 120 neurons, a hidden layer with 60 neurons and an output layer with 23 neurons.

Dissanayake et al. [21] also used CNNs for developing their static and dynamic sign classifiers. In this method, the captured video frames are first separated into static and dynamic signs with a numbered sequence considering the transition time taken to change from one sign to another. After that, the separated frames are fed into the CNN-based static and dynamic classifiers for identifying each gesture to output a complete sentence. For the static sign classifier, an accuracy of 97% and for the dynamic sign classifier, an accuracy of 95% is achieved. Here, the dynamic sign classification model consists of 4 hidden layers with an input shape of 61440 arrays and an output shape of 8 because here, they only considered 8 dynamic signs with 8 video classes.

In [15], images are input to a CNN model. The prediction depends on 3 consecutive frames of a sign, and 2/3 majority is considered. The predicted output sign sequence is compared with a pre-defined sequence to recognize dynamic signs.

Similarly, [6], [16], [7], and [9] also used CNN-based approaches for classification where image-based inputs of the gestures are used. In [16], De Silva et al. have only used simple static word signs, where each gesture is recognized

individually and generated results for each word separately. [9] and [6] also followed a similar approach to De Silva et al. for letters. In [7], they have recognized sentences using finger spelling signing. Their program goes through i number of iterations until the input image data is sent to the neural network and generates the results letter by letter, forming words and sentences. When considering the accuracies, both [7], [9] achieved an accuracy of 95% where [6] and [16] achieved accuracy of 91.23% and 98.61% respectively.

Smitha et al. [13] adopted a co-simulation neural network. This inputs the image-converted feature vector and then compares it with the feature vectors of a training set of gestures. Since the features extracted from the image to be used for recognition were 16, the input layer has 16 neurons. The proposed system is designed to recognize static hand gestures where each gesture is classified individually, letter by letter.

Moving out of using basic models, Liao [24] presented a B3D ResNet model, which mainly includes 17 convolutions layers, two Bidirectional-LSTM layers, one fully connected layer, and one soft-max layer for dynamic sign language recognition. The classification occurs as the processed short-term spatio-temporal feature sequences from the B3D ResNet model are sent to the Bidirectional-LSTM layers. (consequently, an intermediate score is obtained corresponding to each action). Next, the soft-max layer classifies the video sequence label and recognizes dynamic sign language gestures. As the Bi-directional LSTM unit integrates information from the future and the past, it predicts each chunk in the video sequence. In this method. Each gesture is classified individually, and a sentence-based output is given with an accuracy of 89.8% on the DEVISIGN_D dataset and 86.9% on SLR_Dataset. When considering LSTM Neural Networks, Rathnayake et al. [23] used an LSTM Neural Network to classify static and dynamic gestures. Here, the key points extracted from MediaPipe Holistic technology are taken as the input to the network. The proposed LSTM performed with an accuracy of 80% for static and 77% for dynamic gestures. A complete sentence was provided as the output using a transformer after detecting grammatical errors.

The model proposed by Junfu et al. [18] has employed a combination of a Connectionist Temporal Classification (CTC), LSTM aligned with a soft-DTW constraint. Here, the fixed-length features obtained from the previous 3D-ResNet is first given to the bidirectional LSTM(BLSTM). After that the output of the BLSTM encoder is sent to an attention-aware LSTM decoder and a CTC decoder to arrange labelled signs in a more grammatically correct way with the help of soft-dynamic time wrapping algorithm to give the targeted sentence output.

Rishan et al. [14] used a combination of Leap Motion technology with geometric template matching to identify and interpret unique signs. Here, the geometric template matching was based on the $P Point-Cloud Recognizer described by D.Vatavu et al. [25]. In this approach, the input to the model was hand motion data captured by the Leap Motion controller, and grammatically correct sentences are output by considering the history of the signs performed. The NLP unit in the system processes the array of sign gestures and generates meaningful sentences based on the order and combination of signs. It uses regular expressions and the WordNet API for interpreting combined signs and signs with multiple meanings.

2) DTW and Nearest Neighbor:

Fernando et al. [10] used a two-step gesture Identification algorithm where step 01 is based on the Dynamic Time Warping algorithm (DTW) and step 02 is based on the Nearest Neighbor classification. The inputs to the DTW are Real-time coordinate data captured from the Kinect camera and pre-trained sample data. It provides cost function value from the comparison as the input to the nearest neighbour classifier. It selects the gesture name with the minimum cost and provides a word-by-word output with an accuracy of 94.25%.

3) Support Vector Machine(SVM):

The method proposed by Athira et al. [8] used SVM for classification. The SVM training algorithm builds a model that predicts whether a new example falls into one category or another. Here, the feature vectors consisting of Hand shape, Trajectory length, average speed, number of significant curves, the number of points of minima, and palm orientation are used. The proposed method achieved an accuracy of 91% for static gestures and 89% for dynamic gestures.

4) Algorithm based approaches:

Neo et al. [20] employed an algorithm for recognizing numbers from 1 to 5, which inputs the extracted features: height, width and centroid of the hand region. There, a circle with a radius 70% from the farthest distance value was constructed, and active fingers were detected. Radius 70% from the farthest distance is selected based on the experiment conducted on several images that show the selected distance is most likely to intersect with all active fingers. The number of fingers intersecting the circle will be the count represented by the image captured. The number of the transitions accumulated will be deducted by one since the wrist is also detected. The number after the subtraction is considered the number representing the sign gesture.

Gupta et al. [11] also employed an algorithm to identify 10 different hand gestures at a faster rate with an accuracy of 94.40%. Here, the different hand gestures are distinguished based on 4 hand features: area, perimeter, thumb detection and radial profile of hand. In both approaches, a single word is identified at a time individually.

IV. Discussion

This section provides an overview of Vision-based sign language gesture recognition techniques applied in different research papers. Table I summarises the previous work with the published year, static/dynamic data type, the techniques applied for pre-processing, feature detection and classification, and performance.

TABLE I
VISION-BASED GESTURE RECOGNITION SUMMARY

References	Year	Static/Dynamic	Dataset	Pre-Processing	Detection & Extraction	Classification	Implementation	Accuracy
[9]	2021	Static	4 ASL letters “A”, “B”, “C” and “D	Convert images into fixed size	Realtime Single-Shot Detector (SSD) VGG16 model	CNN	Model - Colab and Jupyter Notebooks	95%
[23]	2022	Dynamic	ASL words/sentences	Not mentioned	MediaPipe Holistic	LSTM	Web application	Not mentioned
[10]	2016	Dynamic	15 basic Sinhala word signs	Normalize data and Filtering unwanted noise	Identify skeleton joints points	DTW and K-NN	App, Chatbot	94.20%
[14]	2022	Dynamic	Train itself and give a label	Not mentioned	Geometric template matching technique	ANN	Model	S - 80%, D - 77%
[6]	2020	Static	26 letters	Not mentioned	CNN approach	CNN	Desktop Application	91.23%
[16]	2019	Static	24 hand signs	Convert images into fixed size	Not mentioned	CNN	Model	98.61%
[7]	2019	Static	27 hand signs	Skin detection, Mask to skin tone identification	Not mentioned	CNN	Software application	95%
[15]	2021	Both	10 static signs and dynamic words	Skin segmentation, Morphological transformation, Noise reduction, Contour extraction, Background remove	Binary mask Template matching	CNN	Mobile Application	81.2%
[17]	2018	Static	6 gestures	Resize images and Data augmentation	Not mentioned	CNN	FPGA-based system	86.3%
[20]	2011	Static	1-6 numbers	Image segmentation, Binary image morphology	Algorithm based on RGB values	Algorithm	FPGA (DE2 Board)	Not mentioned
[11]	2012	Static	10 signs	Resize, Illumination compensation, Skin colour segmentation, image filtering	Calculating 4 features	Algorithm	FPGA (Xillinx Virtex2 Pro FPGA board)	94.40%
[13]	2019	Static	ASL alphabet	Converted into feature vector	Algorithm	ANN	FPGA	Not mentioned%
[12]	2014	Static	23 signs of Colombian SL	Hand segmentation and Morphological filtering	Convert the hand shape into vectors	ANN (MLP)	FPGA (DE1 and DE2 boards)	98.15%
[24]	2019	Dynamic	500 signs of Chinese SL signs	Not mentioned	B3D ResNet model	CNN	Model	89.80%
[18]	2015	Dynamic	178 signs of Chinese SL signs	Convert images into fix size	3D-ResNet model	LSTM	Model	Not mentioned%
[22]	2019	Static	29 Swedish alphabet signs	Data augmentation techniques	Not mentioned	CTC + LSTM + soft-DTW	Xilinx FPGA Board	87.1%
[21]	2020	Both	4 static and 8 dynamic signs	Remove background, Convert to grayscale, Edge detection	Thresh-holding method	CNN	Mobile Application	S - 97%, D - 95%
[8]	2022	Both	Indian Sign Language-900 static 700 videos	Face detection and elimination, Hand segmentation and extraction, Noise removal	Convert into feature vector	SVM	Model	S - 91%, D - 89%
[19]	2023	Both	words and ASL alphabets	Frame normalization, Montage and Digitalization	MediaPipe and LabelIMG application	LSTM and YOLOv6	Two Models	LSTM-92%,YOLOv6-96%

-Data acquisition methods are not included in this table, as it is found that many papers lack detailed information on this stage.

The majority of the vision-based studies that were examined rely on a conventional camera or a webcam. Pre-processing techniques are implemented to enhance both accuracy and processing speed. The most commonly applied techniques include filtering methods to remove noises and skin segmentation to identify the hand, reducing the background. This research shows that incorporating skin colour segmentation along with other features, such as edge detection and thresholding, enhances the quality of the segmentation results. Normalizing the image is often used in the pre-processing stage to reduce the computational load. The most used method for sign detection and feature extraction is Media Pipe. When considering classification, CNN is the widely used approach by many research papers where it has shown good accuracy in recognizing static signs. For dynamic signs, the LSTM neural network has shown a higher performance.

V. Conclusion and future work

The application of sign language recognition systems is an emerging and growing trend in society. This paper has covered around 20 research articles on sign language recognition, all of which were published from 2010 to 2023. The objective is to provide a condensed summary of the research focusing on sign languages. It is subsequently organized into various categories, such as data acquisition methods, pre-processing, sign detection and feature extraction, recognition techniques, static versus dynamic signs and accuracy rates.

There are several limitations of this research. Primarily, this research only focuses on vision-based gesture recognition, even though sensor-based methods have already been explored. It's important to note that sign language differs from the spoken language worldwide. This review has considered different types of sign languages. The differences in sign language, including gestures, syntax, and use of body parts, may differ with the language, which could affect the computation.

Despite the fact that research into sign language recognition commenced many years ago, it is still in its early stages. Even though facial expressions and lip movements are equally important in communication, most sign language recognition research papers are based on only hand gesture recognition. Numerous obstacles persist in developing a reliable system, such as dynamic hand detection, database availability, background illumination variations, and high computational cost.

VI. Acknowledgement

This research was supported by the Science and Technology Human Resource Development Project, Ministry of Education, Sri Lanka, funded by the Asian Development Bank (Grant No. STHRD/CRG/R1/SJ/06).

References

[1] I. Adeyanju, O. Bello, and M. Adegboye, "Machine learning methods for sign language recognition: A critical review and analysis," *Intelligent Systems with Applications*, vol. 12, p. 200056, 2021.

[2] J. Murray, "World federation of the deaf. rome, italy." http://wfdeaf.org/our-work/, 2023.

[3] B. K. Triwijoyo, L. Y. R. Karnaen, and A. Adil, "Deep learning approach for sign language recognition," *JITEKI: Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*, vol. 9, no. 1, 2023.

[4] D. S. Ruiz, J. A. Olvera-López, and I. Olmos-Pineda, "Word level sign language recognition via handcrafted features," *IEEE Latin America Transactions*, vol. 21, no. 7, pp. 839–848, 2023.

[5] R. Rastgoo, K. Kiani, and S. Escalera, "Sign language recognition: A deep survey," *Expert Systems with Applications*, vol. 164, p. 113794, 2021.

[6] S. Hettiarachchi and R. Meegam, "Machine learning approach for real time translation of sinhala sign language into text," *University of Sri Jayewardenepura*, 2020.

[7] W. Peiris, "Sinhala sign language to text interpreter based on machine learning," Ph.D. dissertation, 2021.

[8] P. Athira, C. Sruthi, and A. Lijiya, "A signer independent sign language recognition with co-articulation elimination from live videos: an indian scenario," *Journal of King Saud University-Computer and Information Sciences*, vol. 34, no. 3, pp. 771–781, 2022.

[9] D. Dahanayaka, B. Madhusanka, and I. Atthanayake, "A multi-modular approach for sign language and speech recognition for deaf-mute people," *ENGINEER*, vol. 97, p. 1, 2021.

[10] P. Fernando and P. Wimalaratne, "Sign language translation approach to sinhalese language," *GSTF Journal on Computing (JoC)*, vol. 5, pp. 1–9, 2016.

[11] A. Gupta, V. K. Sehrawat, and M. Khosla, "Fpga based real time human hand gesture recognition system," *Procedia Technology*, vol. 6, pp. 98–107, 2012.

[12] J. D. Guerrero-Balaguera and W. J. Pérez-Holguín, "Fpga-based translation system from colombian sign language to text," *Dyna*, vol. 82, no. 189, pp. 172–181, 2015.

[13] R. Smitha, U. S. Kumar, and S. Suresh, "Neural network based sign language recognition using verilog hdl."

[14] R. Rishan, S. Jayalal, and T. Wijayasiriwardhane, "Translation of sri lankan sign language to sinhala text: A leap motion technology-based approach," in *2022 2nd International Conference on Advanced Research in Computing (ICARC)*. IEEE, 2022, pp. 218–223.

[15] I. Dhanawansa and R. Rajakaruna, "Sinhala sign language interpreter optimized for real–time implementation on a mobile device," pp. 422–427, 2021.

[16] S. De Silva *et al.*, "Sign language translator for deaf and speech impaired people using convolutional neural network," 2019.

[17] T. Zhang, W. Zhou, X. Jiang, and Y. Liu, "Fpga-based implementation of hand gesture recognition using convolutional neural network," pp. 133–138, 2018.

[18] J. Pu, W. Zhou, and H. Li, "Iterative alignment network for continuous sign language recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4165–4174.

[19] A. M. Buttar, U. Ahmad, A. H. Gumaei, A. Assiri, M. A. Akbar, and B. F. Alkhamees, "Deep learning in sign language recognition: A hybrid approach for the recognition of static and dynamic signs," *Mathematics*, vol. 11, no. 17, p. 3729, 2023.

[20] K. C. Neo and H. Ibrahim, "Development of sign signal translation system based on altera's fpga de2 board," *International Journal of Human Computer Interaction (IJHCI)*, vol. 2, no. 3, p. 101, 2011.

[21] I. Dissanayake, P. Wickramanayake, M. Mudunkotuwa, and P. Fernando, "Utalk: Sri lankan sign language converter mobile app using image processing and machine learning," in *2020 2nd International Conference on Advancements in Computing (ICAC)*, vol. 1. IEEE, 2020, pp. 31–36.

[22] R. Prieto, "Implementation of an 8-bit dynamic fixed-point convolutional neural network for human sign language recognition on a xilinx fpga board," *Department of Electrical and Information Technology Lund University, dated Mar*, vol. 17, 2019.

[23] R. RKDMP, W. Wijekoon, K. Rajapakse, K. Rasanjalee, and D. De Silva, "Real-time sign language translator," *International Journal of Engineering and Management Research*, vol. 12, no. 6, pp. 117–124, 2022.

[24] Y. Liao, P. Xiong, W. Min, W. Min, and J. Lu, "Dynamic sign language recognition based on video sequence with blstm-3d residual networks," *IEEE Access*, vol. 7, pp. 38 044–38 054, 2019.

[25] R.-D. Vatavu, L. Anthony, and J. O. Wobbrock, "Gestures as point clouds: a $p recognizer for user interface prototypes," 2012, pp. 273–280.