Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, No.7, May : 2024

SIGN LANGUAGE RECOGNITION SYSTEM

Mr. Arun Sharma, Student, Dept. Of CSE(AIML), ABES Engineering College.
Mr. Arun Kumar Yadav, Student, Dept. Of CSE(AIML), ABES Engineering College.
Mr. Anas Ansari, Student, Dept. Of CSE(AIML), ABES Engineering College.
Ms. Parul Phoghat, Assistant Professor, Dept. Of CSE(AIML), ABES Engineering College.

Abstract
This paper introduces a real-time Hand Gesture Recognition model designed for practical applications. Leveraging Google's Tkinter is the standard GUI library for Python, TensorFlow integrated with OpenCV in Python, and a Keras-based feed-forward neural network for classification, the model comprises three integral modules: frame grabbing, hand landmark detection, and classification. Impressively, the model attains a 94.9% accuracy in discerning ten diverse hand gestures, including 27 characters (Alphabets and Blank). A notable achievement of this work lies in the rapid and consistently accurate responsiveness of the hand gesture recognition model, positioning it as particularly suitable for real-time applications. The incorporation of a pre-trained model for feature extraction enhances overall efficiency. The research also emphasizes how well Long Short-Term Memory (LSTM) models work for modeling sequence data and gesture recognition. adding a layer of sophistication to capture temporal dynamics in hand gestures. In summary, the presented approach amalgamates advanced technologies, demonstrating a robust and efficient hand gesture recognition system, so contributing significantly to the field.

Keywords:
Sign Language Recognition, CNN, Data Augmentation, Tensor flow, opencv, keras, python, Tkinter.

I. Introduction

Sign Language, a visual language crucial for the deaf and hard of hearing community in the United States, has become a focal point for communication advancements. Convolutional Neural Networks (CNNs) have emerged as a promising technology for Sign Language recognition, aiming to facilitate interaction between the hearing and deaf communities.

CNNs, well-suited for image and video recognition tasks, are trained on extensive datasets of sign language videos or images. Once trained, these models can classify new, unseen sign language content [1]. The recognition process involves segmenting sign language videos or images into frames, analyzing each frame to discern hand gestures and movements.

Using the Hand Gesture Translation System, several researchers [2]. The CNN model processes these features, generating predictions for the corresponding Sign Language signs. Notably, CNNs excel in their ability to adapt to diverse lighting conditions, backgrounds, and signing styles, making the technology more widely accessible[3].

Deep convolutional neural networks have shown effective in classifying recent recognition tasks. It's been shown that for a range of picture classification tasks, multiple column deep CNN which employ multiply parallel.

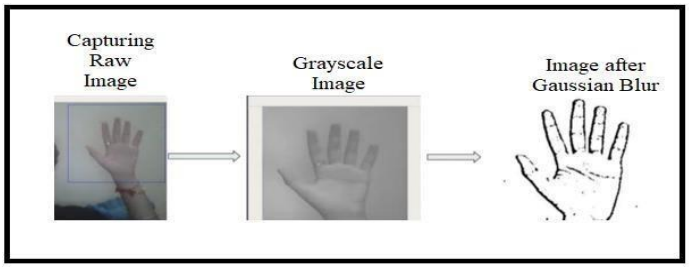

Figure 1: Snapshot of regions

Networks can boost single network recognition rates by as much as 80.0% [4]. The objective of this project is to identify a set of static and dynamic hand gestures while maintaining system accuracy and speed. Gestures that are recognized are used to operate the computer[5]. Recent years have seen notable progress in the recognition of sign language thanks to the combination of deep learning methods, computer vision, and multimodal data processing.

Facilitating communication between those who use sign language and others who don't has grown more and more important in promoting accessibility and inclusivity in a variety of fields, such as communication technology, healthcare, and education.

Drawing on a wide range of academic contributions, this study provides an extensive overview and critique of the state-of-the-art methods in sign language recognition. The advent of deep learning methodologies has revolutionized sign language recognition systems, enabling more robust and accurate interpretations of complex sign gestures. Guo et al [6]. Presented a revolutionary cross-scale aggregation architecture and dimensional global-local shift that performed remarkably well in identifying complex sign motions. Thus, Zhou and associates [7]. Proposed a spatial-temporal multi-cue network that effectively integrates spatial and temporal information for enhanced recognition and translation of sign language.

Furthermore, advancements in network architectures have led to the development of more efficient and scalable models for video-based sign language recognition.

Wang et al [8]. Proposed a skeleton-aware multi-modal framework, which effectively integrates skeletal information with visual cues to achieve robust recognition performance across diverse sign gestures. Moreover, earlier contributions have laid the groundwork for sign language recognition systems, providing foundational insights and methodologies[9]. Transfer learning is used to build a CNN-based classifier for hand shape recognition from a convolutional neural network that has already received extensive training on a sizable dataset.

A technique has been developed to extract hand components from the image while learning and predicting using convolutional neural networks. Therefore, to lessen the chance of overfitting and enhance the gesture classifier's capacity for generalization, it is advised to utilize a successful spatiotemporal data augmentation technique to distort the hand gesture input volumes. Current spatial augmentation techniques are also incorporated into the augmentation process.

II. Related Work

Sign language recognition has attracted a lot of attention lately, particularly in the domains of deep learning and computer vision. In an effort to increase the accuracy, efficiency, and utility of sign language recognition systems, a number of studies have looked into different aspects of these systems. Guo et al. presented a novel method for cross-scale aggregation and dimensional global-local shift in sign language recognition. Their work demonstrates advancements in spatial-temporal feature extraction, contributing to improved accuracy and robustness in sign language recognition systems[10]. Zhou et al. introduced a spatial-temporal multi-cue network tailored for sign language recognition and translation tasks. By leveraging multiple cues and modalities, including visual and temporal information, their model achieves enhanced performance in understanding and translating sign language gestures[11]. The (2+1)D-SLR network, an effective architecture created especially for video sign language recognition, was introduced by Wang et al. Their method efficiently captures the temporal and spatial dynamics of sign language sequences, improving the efficiency and accuracy of recognition. [12].

Lee et al. proposed the Into-TTS system, focusing on prosody control in text-to-speech synthesis. While not directly related to sign language recognition, their work underscores the importance of intonation and expressive cues in linguistic communication, which could inform the design of more nuanced sign language recognition models[13].

Kilic and Karayilan (IEEE) provided insights into the broader landscape of sign language recognition, highlighting key challenges and advancements in the field. Their overview serves as a valuable

resource for researchers and practitioners interested in understanding the historical evolution and current state of sign language recognition technologies[14].
Jiang et al. introduced a skeleton-aware multi-modal sign language recognition approach, integrating skeletal information with visual cues for improved gesture understanding. Their hybrid model demonstrates promising results in recognizing complex sign language gestures across different modalities[15]. Thilagavathi and Pankajakshan presented a sign language recognition system, emphasizing the integration of computer vision techniques and machine learning algorithms for real-time gesture interpretation. Their work contributes to the development of practical sign language communication tools with applications in diverse settings[16].
Ahmad suggested a pairwise conjugate gradient-based technique that is globally convergent stochastic. for adaptive filtering, offering insights into optimization techniques applicable to signal processing tasks, including those relevant to sign language recognition[17].
Mitra and Acharya. conducted a comprehensive survey on gesture recognition methodologies, outlining key approaches and challenges in the field. While not specific to sign language, their survey provides valuable context for understanding the broader landscape of gesture recognition research[18].
Chakraborty et al. (Associate of Computer Management) developed a deep convolutional neural network-based trigger detection system for American Sign Language. Their work demonstrates the feasibility of leveraging deep learning techniques for real-time sign language recognition applications, paving the way for more sophisticated gesture interpretation systems[19]. Overall, the related work in sign language recognition encompasses a wide range of methodologies and approaches, from deep learning architectures to multimodal fusion techniques. By building upon these advancements and addressing existing challenges, researchers continue to push the boundaries
of sign language recognition technology, ultimately striving towards more inclusive and accessible communication system.

III. Methodology

3.1 Workflow

The suggested methodology is based on the utilization of a deep Convolutional Neural Network (CNN) for the recognition of sign language. The decision to use CNNs is grounded in their capability to accurately capture the intricacies of hand signals, particularly when these signals exhibit inherent similarities.
The primary objective is to differentiate between visually similar actions, and CNNs are deemed well-suited for this specific task. Their inherent capability to automatically extract features at multiple hidden layers proves advantageous in efficiently processing raw data, a characteristic highly beneficial for computer vision tasks. Unlike alternative gesture recognition systems in computer vision, CNNs eliminate the need for manual feature engineering. Hence, the use of CNNs becomes imperative in the context of sign language recognition. The trained model is subsequently integrated into a graphical user interface (GUI) based application, enhancing user accessibility. The recommended method's block diagram is depicted in Figure2.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, No.7, May : 2024

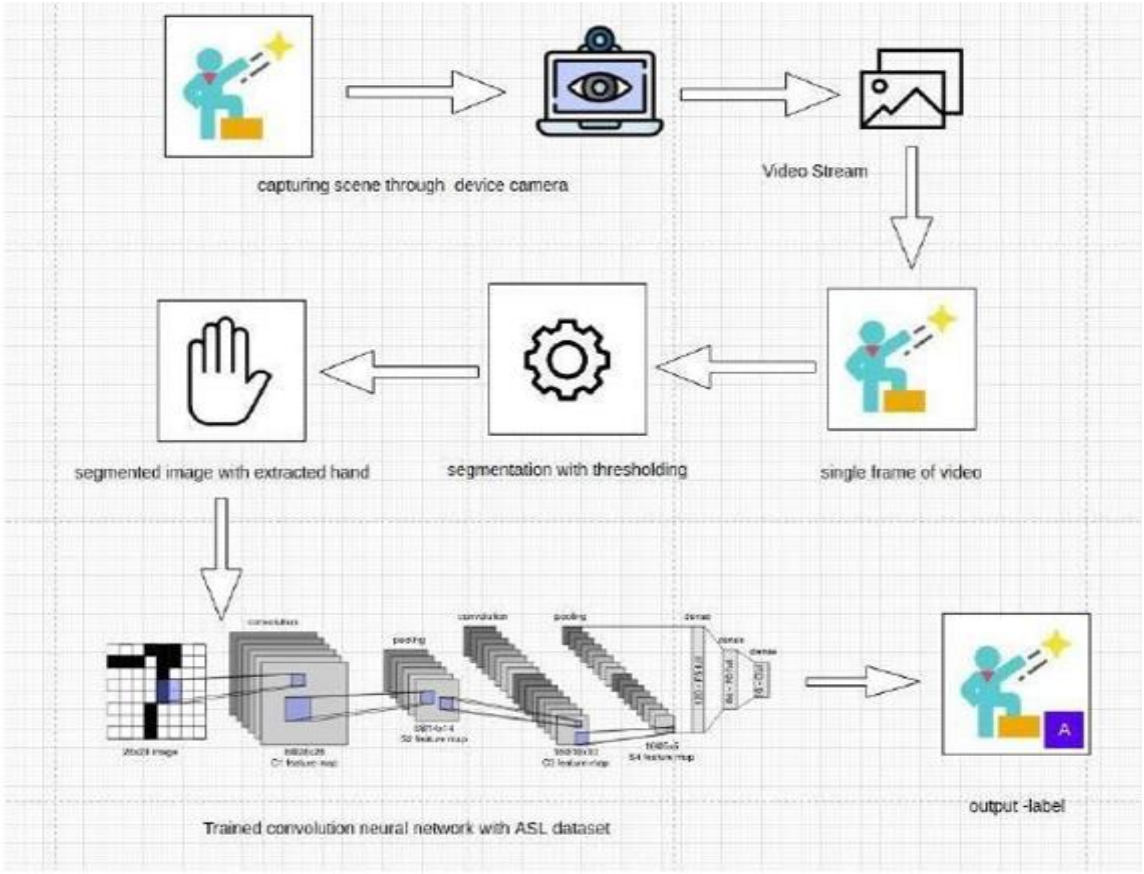

Figure 2: Modular Framework

3.2 Dataset

To train a model, one needs a training set. Gestures are manually generated via webcam and Python programming in order to train a model. To identify a static gesture, only the hand shape is required.

Upon recognizing a static gesture through a trained classifier, the computer receives an instruction. Unlike static gestures, dynamic gestures involve both hand motion and shape changes. To track dynamic hand gestures, the HSV skin color algorithm is utilized to segment the hand area in a frame, and the resulting blob area is refined.

The focus is on detecting and tracking the centroid of the blob in each frame, with the primary goal being to ascertain the coordinates of the traced hand's center.

These coordinates are then employed to match the motion with a specific computer command, facilitating the interaction between hand gestures and computer actions.Using a webcam, 27 hand gesture (26 alphabets and one blank) were recorded for this study in order to assess the model.

Every picture has 40 by 40 pixels. Using a technique known as Gaussian Blur, black and white skin pixels are created by extracting color skin pixels from the color image (Figs. 1 and 3). Rather than combining the picture with a box filter, a Gaussian filter [29] is employed in a Gaussian Blur process.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, No.7, May : 2024

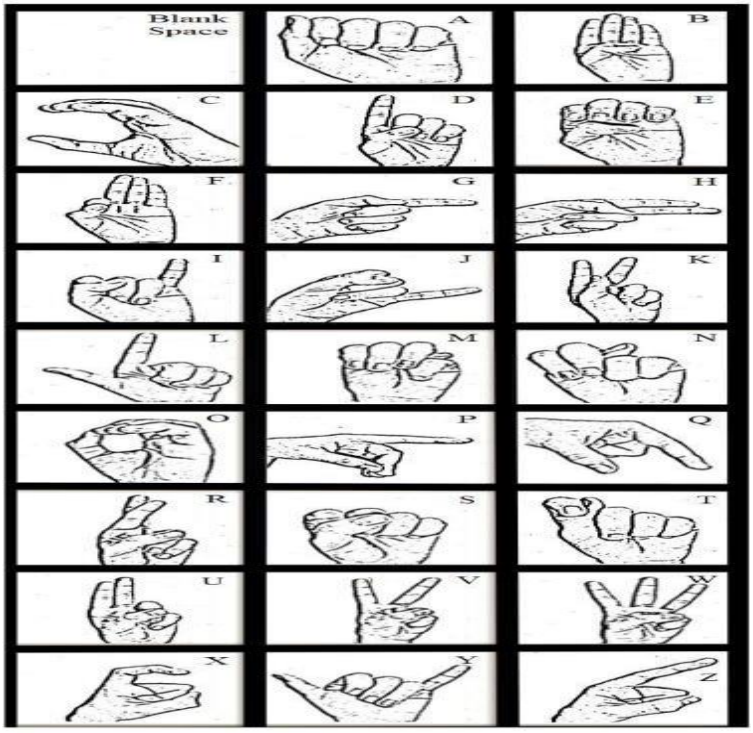

Figure 3: Hand Gestures

There are other components in the frame with the gesture, including empty spaces. Separating unnecessary elements is crucial for improved outcomes. The identified segment is adjusted to eliminate irrelevant hand gestures. The cropping process unfolds in three steps. Firstly, the cropped frame undergoes global thresholding, converting it into a binary (black and white) format. The second stage focuses on extracting the essential element from the frame, effectively isolating it from the background. The corresponding photographs for each hand gesture are arranged into an own folder. Every folder contains a text file with entries for every image. An entry in the text file corresponds to one of the hand gestures observed in (fig. 3).

3.3 Classifier

The Convolution's main goal is to extract features like corners, edges, and colors from the input. As we explore the network further, it starts to identify more intricate components, such as figures, numbers, and even distinct face characteristics. Convolution's main goal is to extract features like corners, edges, and colors—from the input. As we explore the network further, it starts to identify more intricate components, such as figures, numbers, and even distinct face characteristics.

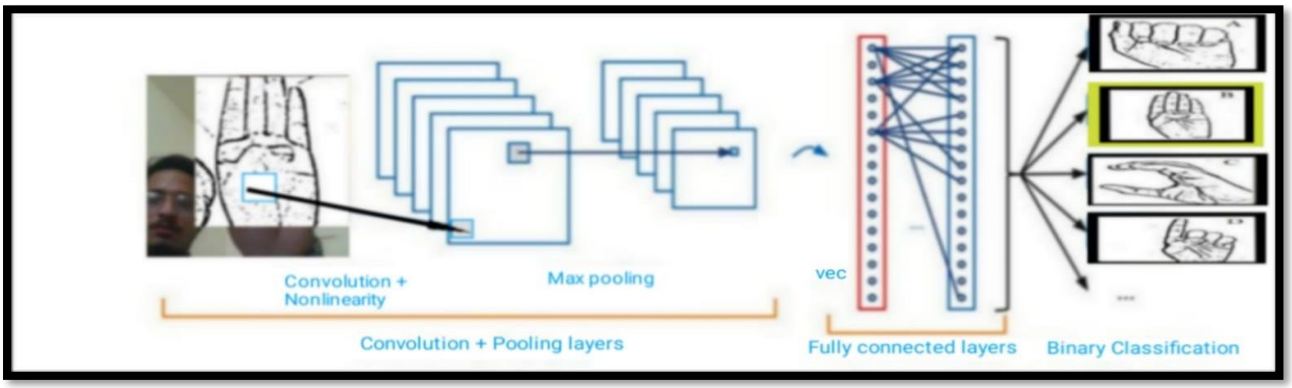

Figure 4: CNN Architecture

A completely connected 27-layer network receives the output of the sixth convolution layer. Except for the last output layer, which comprises a total of 27 neurons—one for each of the 27 hand

movements each layer contains 512 hidden neurons.
In the output layer, apply the sigmoid activation function. The sigmoid function is built using a "S" graph, and any value between 0 and 1 can be used as the input value for x. This is how the sigmoid activation function is calculated:

$$A = 1/1+e^{-x}$$

The Tanh (hyperbolic tangent) activation function, a mathematically modified sigmoid function, is used for the next 35 levels. The function's input values span from -1 to +1, and its representation is as follows:

$$Tanh = e^{x} - e^{-x} / e^{x}$$

In this article, obtaining a larger dataset for every topic would be time-consuming and unfeasible when taking into account real-life applications because users would typically not tolerate hours of data recording for every training. Batch Normalization is utilized to tackle this overfitting issue in greater detail in the ensuing subsections.

3.4 Training

To minimize the cost function of the dataset, adjustments to network parameters are necessary during the training process of a Convolutional Neural Network (CNN). In this project, the root mean square method was utilized for this purpose. Liao et al. employed a dual-channel CNN architecture to concurrently learn segmented depth and RGB images. Similarly, our proposed model followed a comparable approach, simultaneously acquiring knowledge of RGB-D pairs. The training process involved separately training RGB and depth images using the same CNN-based architecture illustrated in Figure 4. Subsequently, the model's performance was assessed in both offline evaluations and real-time scenarios.

3.4.1 Convolutional Layer

A small window size, typically 5 by 5, is employed in the convolutional layer to span the input matrix's depth. This layer consists of filters with set window sizes and learnable parameters. The window is shifted with a stride size, usually 1, in each iteration, and the dot product of the input values at a particular place and the filter entries is computed. A 2-dimensional activation matrix that captures the matrix's response at each spatial place is produced by this iterative approach. In essence, the network gathers filters that become active when particular visual characteristics, like edges with particular orientations or areas of different hues, are present.

3.4.2 Pooling Layer

We use a pooling layer to reduce the size of the activation matrix and, ultimately, the learnable parameters. There are two types of pooling: a. Max Pooling: This method only takes the top four values and utilizes a window size, like a 2*2 window. We continue sliding the window in this manner until, at some point, the activation matrix is half as large as it was at first. c. Average Pooling: In this technique, all values are averaged over a given period of time.

3.4.3 Fully Connected Layer

In the fully connected layer, all inputs are coupled to neurons, while in the convolutional layer, neurons create local connections. A set of neurons in the final layer, the number of which equates to the total number of classes in the final output layer, receives these values once they are extracted from the fully connected layer. The last layer is responsible for predicting the probability that every image will belong

to a distinct class.

IV. Results

This experiment involves capturing hand movements through live video. However, a complication arises when using the program under different lighting conditions. To enhance hand motion identification, the RGB image is converted to an HSV representation (hue, saturation, and value). The subsequent step involves thresholding to reduce background noise.

When determining the accuracy score of the model, both the actual outcome and the predicted value are considered. The accuracy score is computed using the following mathematical form.

$$\text{Accuracy} = (TP + FP) / \text{Total}$$

As stated below, precision is expressed mathematically as a percentage of pertinent outcomes. As a percentage of actual results, it is a true positive.

$$\text{Precision} = TP / (TP+FP)$$

The percentage of all relevant outcomes that the trained model properly classifies is known as recall. It is the percentage of predicted results that are true positives. The following represents the mathematical form for calculating recall.

$$\text{Recall} = TP / (TP+FN)$$

F1-score is the harmonic mean of precision and recall. The formula below represents the F1-score in mathematical form.

$$\text{F1score} = 2*(\text{Precision}*\text{Recall}) / (\text{Precision} + \text{Recall})$$

The accuracy, precision, recall, F1-score, and avg for the training and testing sets are displayed in the following table.

Characters	Precision	Recall	F1-score	Support
0	1.00	1.00	1.00	154
A	1.00	0.96	0.98	156
B	0.80	1.00	0.89	157
C	1.00	1.00	1.00	156
D	1.00	0.91	0.95	155

E	0.97	1.00	0.99	156
F	0.97	0.86	0.91	165
G	0.76	0.97	0.85	153
H	1.00	0.72	0.83	165
I	0.96	0.99	0.97	157
J	0.99	1.00	1.00	166
K	0.99	1.00	0.99	164
L	1.00	1.00	1.00	155
M	0.91	0.95	0.93	165
N	0.76	0.98	0.86	163
O	1.00	1.00	1.00	167
P	0.96	1.00	0.98	155
Q	1.00	0.93	0.96	154
R	0.91	0.97	0.94	157
S	0.99	1.00	0.99	154
T	0.96	0.68	0.80	155
U	0.94	0.82	0.87	153
V	0.99	0.97	0.98	163
W	0.98	0.99	0.98	156
X	0.97	1.00	0.99	153
Y	1.00	0.99	1.00	158
Z	1.00	0.97	0.99	156

Accuracy			0.95	4268
Macro avg	0.96	0.95	0.95	4268
Weighted avg	0.96	0.95	0.95	4268

To get a good outcome, we have to manually modify the HSV value if the illumination in the room changes, which makes things more difficult. Sometimes it might be difficult to distinguish hand motions from different hands due to variances in skin tone.

The ratio of true positives to true negatives is used to calculate accuracy, a performance metric for machine learning classification models. This is done by dividing the total number of positive and negative observations. Stated differently, accuracy represents the probability that, of all the predictions the machine learning model has made, the results will be as expected.

Industrial Engineering Journal
ISSN: 0970-2555
Volume : 53, Issue 5, No.7, May : 2024

Figure 4: Snapshot of Tkinter Application

V. Conclusion

In conclusion, the development and implementation of sign language recognition systems hold immense potential in enhancing accessibility and inclusivity for individuals with hearing impairments. The presented research introduces a real-time hand gesture recognition model integrated with advanced technologies like TensorFlow, OpenCV, and Keras, culminating in a robust and efficient system capable of accurately discerning diverse hand gestures with a notable accuracy of 94.9%.

The efficacy of convolutional neural networks (CNNs) in sign language recognition has been demonstrated, particularly in their ability to adapt to varying lighting conditions, backgrounds, and signing styles. Leveraging CNNs, along with techniques such as data augmentation and transfer learning, contributes to the system's ability to generalize well and maintain high accuracy across different gestures.

Moreover, the incorporation of Long Short-Term Memory (LSTM) models adds a layer of sophistication to capture temporal dynamics in hand gestures, further enhancing the system's capabilities in recognizing sequential sign language expressions.

Despite the achievements, there remain avenues for improvement and future research. Addressing challenges related to lighting conditions, regional variations in sign languages, and user feedback are crucial for enhancing the system's robustness and usability. Additionally, the exploration of natural language processing techniques and the incorporation of multimodal capabilities can contribute to improving translation accuracy and user interaction experience.

References

[1] Neural Computing and Applications, vol. 35, pp. 12481–12493, 2023; Guo, Y., Hou, and W. Li, Z., "Sign language recognition via dimensional global–local shift and cross-scale aggregation".

[2] In IEEE Transactions on Multimedia, vol. 24, pp. 768–779, 2022, H. Zhou, W. Zhou, Y. Zhou, and H. Li published "Spatial-temporal multi-cue network for sign language recognition and translation".

[3] Into-TTS: Intonation Template based Prosody Control System. arXiv 2022, Lee, J.; Lee, J.Y.; Choi, H.; Mun, S.; Park, S.; Bae, J.S.; Kim, C.Ozkan Kilic and Tulay Karayilan, "Sign Language Recognition," IEEE.

[4] Labaka, G.; Perez-de Viñaspre, O.; Núñez-Marcos, A. A survey on computer translation for Sign Language. Syst. Appl. Expert 2022, 213, 118993.

[5] "(2+1)D-SLR: an efficient network for video sign language recognition," F. Wang, Y. Du, G. Wang, Z. Zeng, and L. Zhao, Neural Computing and Applications, vol. 34, pp. 2413–2423, 2022.

[6] "Sign Language Recognition," by Tulay Karayilan and Ozkan Kilic, IEEE.

[7] Skeletons aware multi-modal sign language recognition, S. Jiang, B. Sun, L. Wang, Y. Bai, K. Li, and Y. Fu, in 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops

(CVPRW), pp. 3408–3418, IEEE, 2021.
[8] The Visual Computer, vol. 36, pp. 1233–1246, 2020; J. Imran and B. Raman, "Deep motion templates and extreme learning machine for sign language recognition".
[9] "Selective kernel networks," in 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 510–519, IEEE, X. Li, W. Wang, X. Hu, and J. Yang.
[10] X. Zhang and X. Li, "Dynamic gesture recognition based on MEMP network," Future Internet, vol. 11, no. 4, Article ID 91, 2019.
[11] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural sign language translation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 7784–7793, IEEE, 2018.
[12] "A deep learning approach for analyzing video and skeletal features in sign language recognition," by D. Konstantinidis, K. Dimitropoulos, and P. Daras, appeared in the 2018 IEEE International Conference on Imaging Systems and Techniques (IST), pp. 1–6, IEEE, 2018.
[13] Priyanka C. Pankajakshan and Thilagavathi B., "Sign Language Recognition System," IEEE 2015.
[14] "Sign language recognition using convolutional neural networks," L. Pigou, S. Dieleman, P.-J. Kindermans, and B. Schrauwen, Computer Vision–ECCV 2014 Workshops, pp. 572–578, Springer International Publishing, Zurich, Switzerland, 2014.
[15] A. Ahmad, "A Globally Convergent Stochastic Pairwise Conjugate Gradient-Based Algorithm for Adaptive Filtering," IEEE Signal Processing Letters, vol. 15, pp. 914-917, 2008, doi: 10.1109/LSP.2008.2005437.
[16] S. Mitra and T. Acharya, "Gesture Recognition: A Survey," in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 37, no. 3, pp. 311-324, May 2007. doi: 10.1109/TSMCC.2007.893280.
[17] "Deep Convolutional Neural Network-Based Trigger Detection System for American Sign Language," Associate of Computer Management, N. A. Debasrita Chakraborty, Ashish Ghosh, Deepankar Garg, Jonathan H. Chan.