International Journal of Trend in Scientific Research and Development (IJTSRD)
Special Issue on Smart Innovations in Computer Science and Applications
Available Online: www.ijtsrd.com e-ISSN: 2456 – 6470

Sign Language Recognition System Using a Convolution Neural Network Model

Arshiya Javed Sheikh
Department of Science and Technology, G.H.Raisoni Skill Tech University, Nagpur, India

ABSTRACT
Deaf and hard-of-hearing individuals use sign language to communicate with their peers and others. The process of using computers to recognize the sign language involves recognizing signs through gesture recognition as well as converting signs into text or speech. The signs can be classified as either static or dynamic, with static gesture recognition being easier than dynamic gesture recognition; however, both types of gesture recognition systems are very useful to the human community. This paper describes the methods used to recognize sign language. The different stages of sign language recognition, such as how the data is collected, preprocessed, transformed, and recognized, as well as the results from using these methods, are discussed within this article. Several future research avenues regarding sign language recognition are also presented.

Keywords: sign language recognition; static gesture recognition; dynamic gesture recognition; gesture analysis; face recognition.

1. Introduction

The transfer of information from one location to another and/or from one individual or group to another can be defined as communication. Communication involves three participants: the individual(s) communicating the message, the message itself, and those receiving the message; in order for communication to be considered effective, the message being sent by the individual(s) sending the message must be received in the same way as the sender sent it. There are four types of communication, each with its own level of formality/level of informality: 1) formal/informal, 2) oral communication (face-to-face/distance), 3) written communication, and 4) non-verbal communication. In addition, the channels of communication have been predetermined or otherwise established by the organization or business. Informal communication (conduit communication) refers to the non-standardized channels of communication between individuals within an organization; informal communication occurs without any predetermined formal structure or format. Oral communication (face-to-face/distance) occurs when two individuals are communicating via spoken language (as opposed to written), and the individuals communicating may be located together or situated apart (e.g., through the use of technology such as a telephone or video call). Written communication refers exclusively to the transmission of written messages (e.g., letters, emails, memos, etc.). Non-verbal communication refers to all types of communication that occur without the use of words (e.g., using body language) and may include such things as facial gestures, body posture, and gestures.

The feedback communication is created when a person produces feedback on a product or service provided by a person or business, and the visual communication is produced when a person gets

information from visual sources like boxes, social networks, etc. Active listening is when a person will really listen to what the other person is telling them and understand what the person is trying to say, so they can make the communication much more effective and meaningful. (1) Non-verbal communication is a way to communicate for persons with deafness and/or no speech. Deafness is a condition that has a negative impact on a person's ability to hear, while, in contrast, no speech is a condition that has a negative impact on a person's ability to talk. Without being able to communicate with anyone through the ability to talk or hear, it is often difficult to communicate with other people. Sign languages give people the opportunity to communicate with others without using words. The problem continues to exist as fewer and fewer people learn the sign language; therefore, the persons who are deaf and/or have no speech are able to communicate with each other with sign languages, but it is also difficult for them to communicate with persons who are able to hear and speak because both parties lack knowledge of the sign languages.

A major technological advancement of this nature will allow for the accurate and rapid conversion of American Sign Language (ASL) gestures into the more commonly used English language. To date, much research has been conducted in this area; however, continued research is required for the development of accurate conversion models. Various devices have been utilised in the development of gesture conversion applications, including data gloves and motion capture systems(2); a number of vision-based gesture conversion applications have also been developed(3). A gesture recognition (SLR) system utilizing machine learning algorithms to perform gesture recognition for Indian Sign Language has been created using MATLAB(4). Researchers could build a gesture recognition system capable of recognising one- and two-handed gestures by implementing a combination of two different algorithms (K-Nearest Neighbour Algorithm and Back Propagation Algorithm). Their method has been shown to produce accuracy rates ranging from 93% to 96%; however, it is important to note that it is not a real-time SLR system. The purpose of this paper is to propose a real-time SLR system using TensorFlow Object Detection API and to train this SLR system using an original dataset created through a webcam video capture device. The layout of the remainder of this document is as follows: 1) Section 2 explores existing research works that produce SLR systems; 2) Section 3 discusses the access and generation of data used to train the proposed SLR system; 3) Section 4 describes the proposed methodology for creating and implementing a real-time SLR system.

1.1. Motivation

Human beings are social creatures and need to communicate with each other; however, people who are deaf or hard of hearing often experience significant barriers to communicating with hearing members of society. People who are deaf or hard of hearing generally rely upon sign language as their primary means of communicating with others; however, many hearing people do not know or understand sign language; therefore, there is a large communication gap between people who are deaf or hard of hearing and the hearing community that can have a negative impact on many aspects of daily life including access to education, employment, health care and the ability to engage in social interactions. The recent advances in artificial intelligence and computer vision have made it possible to create systems that can detect hand gestures with a high degree of accuracy and convert them into text or speech. The goal of this project is to create a Sign Language Recognition System that will use cutting-edge artificial intelligence techniques to bridge the communication gap between deaf and non-sign language users. By developing a Sign Language Recognition System, this project will promote greater inclusion, independence, and equal opportunities for individuals with hearing and/or speech impairments.[3,4]

1.2. Contribution

This design's principal contribution lies in the creation of an intelligent system that utilizes image processing and machine learning techniques to recognize or identify signed gestures. The system

uses a camera to capture hand gestures, processes the captured images, and identifies signs by directly classifying them into corresponding symbols, words, or meaningful text.

This design contributes to the field of assistive technology by providing an inexpensive, user-friendly, and efficient method for communicating with other people in real time. In addition to this, it also demonstrates how practical applications of deep learning models— such as convolutional neural networks (CNN)— can be useful for recognizing gestures using computers. Furthermore, this system can be further developed to allow for multiple sign language interpreters, recognizing complete sentences from voice input, or producing voice output through spoken language. Therefore, this system has potential for use across various real-world applications, including educational institutions, public service agencies, and assistive technology devices. [5,6]

2. Related work [7]

Sign languages are a type of visual language that consists of a set of gestures with particular meanings used by Deaf individuals for day-to-day communication (3). Sign language uses hand, face, and body movements as means of communication. There are over 300 different sign languages in use throughout the world (5). Although many sign languages exist, the number of people in the general population who know them is small, which can prevent especially-abled individuals from having the freedom to communicate easily with everyone.

Sign Language Recognition Systems (SLRs) allow for communication using a sign language without the knowledge of the language. Specifically, an SLR recognizes a hand gesture and converts that into a commonly spoken language like English. The area of SLR and general techniques used for machine learning is very broad. Although much progress has been made on SLRs, there are a number of significant issues that still require work. Machine learning techniques permit automated systems to make decisions based on prior experience (i.e., data).

In order to perform bracket algorithms**,** two datasets **are needed- a** training dataset and **a** testing dataset. The training set **trains** the classifier, **while** the **classifier** is **then** tested using the testing set **(**6). **There have been several effective** authors **who** have **created** effective **methods for** data **accommodation as well as** bracket **methodologies (**3)(7). **As per** the data **access methodology**, **previous** work can be **categorized** into two **groups:** direct **measurement methodologies** and vision-**based methodologies** (3). The **measurement-based methods** are **based upon the use of motion-detecting** gloves, **motion capture** systems, or **motion-detecting devices**. The **motion detected from these methods provides a reliable representation** of **the location of the** hands**, fingers**, and other **parts of the body, leading to** the development **of accurate** SLR **methodologies. Vision-based SLR methodologies use discriminative** spatial and temporal **features** from RGB images. **The vast majority of vision-based methodologies first attempt** to **locate** and **track** the **hands** before **detecting brackets for the hand (**3). Hand **detection** is achieved **through** semantic segmentation and skin **color detection and is relatively easy due to** the **ease in identifying skin color** (8)(9). **However**, because the other **parts of the** body**, i.e., the** face and arms, can **easily be misidentified** as hands, the **more** recent hand **detection methodologies** use face **expression** and **face detection as well as** background **detection** to only **identify** the moving **parts** in **the** scene **(**10)(11).

3. Proposed Methodology [8]

3.1. System Architecture

The system architecture of the proposed subscribe Language Recognition system is designed as a modular and sequential channel that processes hand gesture data from acquisition to final textbook affair. Each module performs a specific task and passes its affairs to the coming module, achieving accurate and real- time gesture recognition.

1) Data Acquisition

Data Acquisition is the **very first** step of the **subscribed** Language Recognition **System and is the most tärkeimmät**. In this **step**, **you collect** hand gesture data **as** images or **videos**. **This information** can **either** be **gathered** from **publicly** available sign language **data sets** or **live through the use of** a webcam or camera device. Each **motion** corresponds **with** a specific sign language symbol**, such** as **nouns**, **numbers,** or words.

Usage: The main **use of** data **collection** is **for the creation of a separate labelled** dataset **for all the** different **configurations of hands**, **images,** and lighting conditions. **The better** the quality of **the data set, the better** the system **will be able to correlate** gesture **characteristics with direct motion** and **thus improve** recognition **ability**.

2) Hand Detection and Landmark Extraction

The first part of this process involves detecting the hand area on an image frame from a camera and collecting distinguishable points on the hand, known as "landmarks". The hand detection process will eliminate unnecessary background area and target the area where the hand is located. The landmark generation process will determine the important markers of the hand, including the knuckle joints, the top of the fingers, and the centre of the palm. MediaPipe is utilized for hand landmark generation, creating a total of 21 hand landmarks in real-time. Objective: The purpose of this step is to provide accurate hand structure data to eliminate noise and improve the accuracy of gesture recognition.

3) Data Preprocessing

The preparation of hand-collected data for training a model and conducting a detailed analysis involves data preprocessing, which provides cleaner and more consistent data for the model to work with. There can be many problems with raw data, such as noise, size differences, and differences in lighting of the image, and preprocessing creates uniformity and quality for the data.

Preprocessing steps may include:

- Extracting the area of the image that contains only the hand,
- Resizing all images to be the same,
- Standardizing pixel values,
- Converting the hand landmarks to numbers, and
- Numbering different gesture types.

The primary objective for this stage is to enable the model to learn in a manner that allows for either memorizing or understanding the resulting data and provides a higher level of accuracy for the whole system.

4) Feature Extraction

When extracting features from maybe one image of the assigned person or area, we need to know what to look for so that we can identify the gesture as best as possible amongst all other possible gestures. For example, when trying to extract features from a hand gesture, instead of looking for all of the details of the hand (i.e., fingers), we will look for the critical items (i.e., the position of the fingers to form a hand position or the distance between the fingers) that provide the information needed to show how various parts of the hand are positioned relative to each other in space. Feature extraction is, therefore, a critical precursor to the next step in processing the hand gesture and will provide a means of reducing the volume of data to be processed while also providing the classifier with sufficient data for an accurate gesture classification.

5) Gesture Classification

The extracted features will now be input into a deep learning model to make a classification. One common application of deep learning for pattern recognition in images is using Convolutional Neural Networks (CNN). A CNN uses a set of learned patterns during the training phase to classify previously unseen gestures when presented with new data.

The objective of gesture classification is to accurately identify a sign language gesture being performed in real time.

6) Text Output Generation

The final stage converts the recognised gesture into clear text that can be easily read. The system will display the predicted gesture label on-screen as it happens. The system can also convert this text to speech using a text-to-speech function.

Purpose: This part will assist people in improving the way they communicate and help people who are unable to hear communicate with other people in the world who can hear by converting sign language movements to either text or spoken words, so that they can interact with each other without having to rely on one another's ability to communicate.

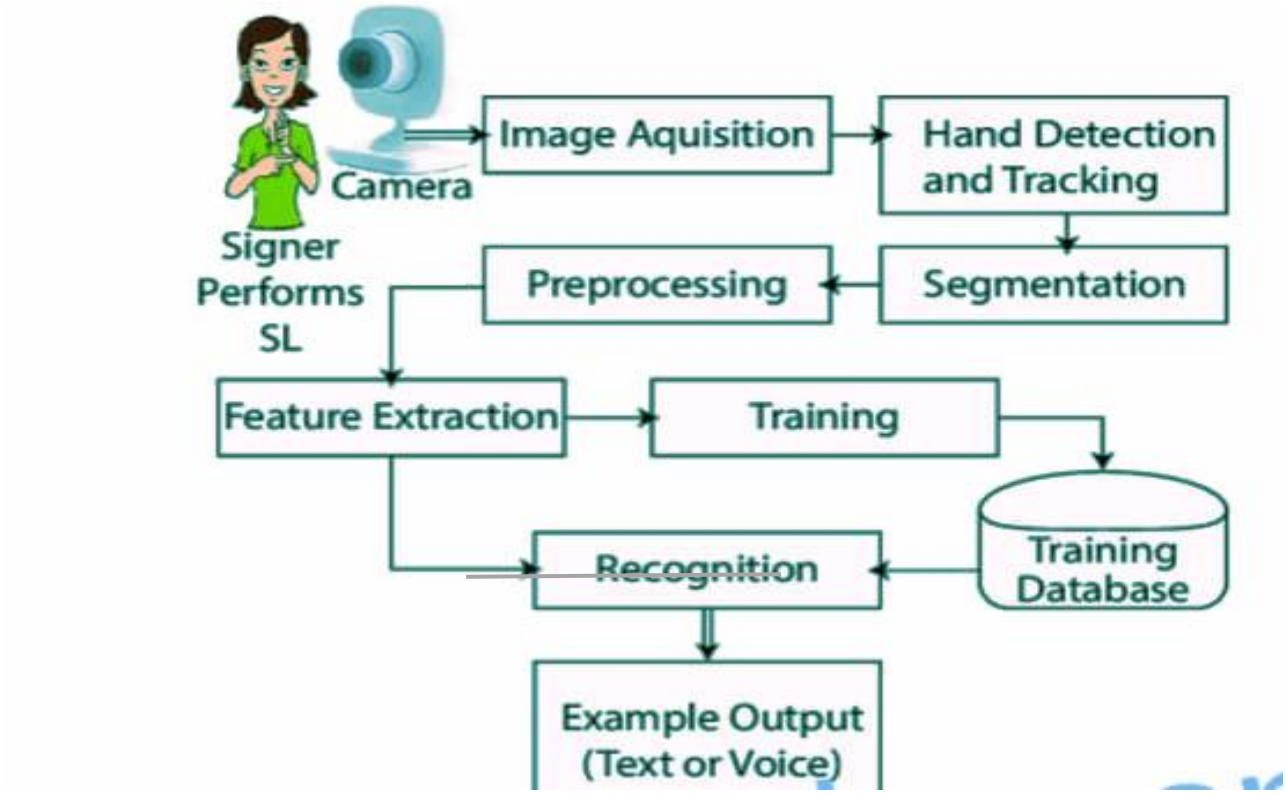

Fig 1. Methodology of the System Architecture.

3.2 Dataset Collection [9]

A publicly accessible data set is utilized for training and evaluating the system. It contains a total of 23 different classifications. The letters of the alphabet are used to classify each of the signs contained in this data set: A, B, C, D, E, F, G, I, K, L, M, N, O, P, Q, R, S, T, U, V, W, X & Z (sign language). The entire data set consists of many different collections or groups of signs that have been photographed and manipulated in order to create images that appear as if they have been photographed in the real world. This data set was taken from Kaggle (https://www.kaggle.com/datasets/soumyakushwaha/indian-sign-language-dataset). These original photographs contain various signs that are indicative of actual sign language expressions, while also containing a variety of different backgrounds that depict noisy, blurry, messy, and colourful images, thus creating an appearance of realistic forms of communication. A small number of images

depicting each of these 23 signs are shown in Fig 2. The complete data set contains the letters of the alphabet - 23 different signs from A to Z - except for the letters H, J, and Y; a total of 702 photographs, each measuring 126x126 pixels, are included in the data set.

1) Image Processing [10]

Before performing any processing on images that will then be used in a machine learning application (e.g., sign language recognition), it is necessary to perform some image preprocessing operations beforehand (e.g., cleaning up and standardising the image files). Just as you do with your data for analysis, image preprocessing is used to modify the image file so that it can be used as input to a computer vision model. Image preprocessing may involve removing extraneous background noise (e.g., trees, grass, etc.) from an image or resizing images to be at a similar scale for improved training. Image preprocessing is an essential function of a machine learning model to maintain the integrity of data that the model uses to learn. Image preprocessing reduces the amount of noise in an image and isolates the most prominent features in the image. As a result, the model learns better and will perform better in recognising sign language due to the improved quality of data that it is trained on

2) Normalization [11]

Images are generally arranged into a grid of pixel units, where the numerical value assigned to each pixel represents the amount of colour and intensity. These numbers may vary according to the format used to save the image. When working with raw data through a CNN, there are difficulties regarding the inconsistency of data and the use of activation functions. A possible solution for handling these issues may be to perform normalisation, in which all pixel values of an image are transformed into a common scale for processing purposes.

A function called toTensor() converts numerical pixel data (values generally ranging from 0-255 for grey-scale images, or from 0-255/img for colour (RGBA) images) into PyTorch tensors. The range of normalised pixel values created by the toTensor() function will have a minimum value of 0 and a maximum value of 1.

3) Resizing

If you use unadjusted images when using the data you have, you can result in a greater computational demand on the CNN and cause difficulties in finding patterns in the image data. To eliminate those issues, we will resize each image in the dataset to be of a uniform size, to be 224 pixels by 224 pixels. The uniform size of all images is selected to be the same as the number required by the architecture to ensure proper use of the pre-trained weights for the model on which it was trained (i.e., the VGG16 model had originally been trained using 224 pixel by 224 pixel-sized images).

Fig. 2. Sample Image Dataset Collection

3.3 Data Preprocessing

The pre-processing phase helps prepare the raw hand gesture data in such a way that gives the deep learning algorithm a high chance of performing well with the data that has been pre-treated for hand gesture recognition; this means that the data leading up to each hand gesture being captured will come from images or video frames.

However, the input data resulting from the images or video frames of a person's hand in a hand gesture may contain noise, data of different sizes, variable lighting conditions, and various distracting things in the background.

Via pre-processing, the video of the hand gesture is divided into individual frames, and the hand region in each video frame is obtained so that all information except the hand is removed from the video frame prior to being passed through the deep learning algorithm for hand gesture recognition.

To isolate the hand in these videos, special techniques are used to obtain only the area of the hand, and therefore all other information, including the background is eliminated from each frame before they, too, are passed through the deep learning algorithm; in this instance, this is often done using the MediaPipe library which easily isolates the hand (its position) and hand keypoints from each of the individual video frames.

After extracting the hand image, it is resized to be the same size as the training image for the neural network to process properly, either 64x64 or 128x128 pixels. The pixel values are then scaled between 0 and 1, for easier working with the numbers, and also help to improve the training of the model overall. If hand landmarks are used, then the keypoints in the images would be converted into numerical information.

Finally, the gesture labels that are associated with the images are converted to numerical values and divided into three datasets, develop specific model performance, prevent the model from being over-fit or learn too much from the training dataset, and allow for quicker and more accurate recognition of gestures.

3.4 Feature Extraction

In the feature-extraction process, key characteristics of a hand are detected from the hand that has been detected and then used for analysis. The three major types of characteristics of the hand include finger joints and tips, palm positioning, and how the fingers move relative to one another. Key characteristics are identified by obtaining their positioning through coordinates (in terms of x and y). The coordinates are adjusted and converted to numerical values, which assist in the elimination of excess data and the simplification of the calculations. A classification model then performs classification on the identification of sign language signs using coordinates that have been extracted from the hand that has been identified.

4. Classification using CNN [12]

The Sign Language Recognition system's most significant stage is the classification of gestures, which is performed as part of classifying the identified features or preprocessed hand images into sign language gestures; specifically, letters or symbols based upon the specific sign language. CNNs are very effective as an example of machine learning for recognizing visual patterns, including shapes, edges, and how parts of an image relate to each other. The key benefit of using a CNN is that it is able to automatically learn important features within the gesture images without requiring pre-manufactured feature sets by people. The CNN is robust against minor variations in: gesture angles, size, and lighting conditions.

CNN Architecture and Layers:

1) Convolution Layer

- The first and most important component of CNNs is their input layer.
- A CNN will utilize multiple filters (or kernels) over the input image data.
- CNNs will determine basic hand gesture features such as edge, curves, and hand shape.
- The CNN will produce a feature map, which is a visual representation of the hand gesture being used.
- The goal of this step in the process is to collect sufficient detail to provide a visual representation of the hand gestures being used.

2) ReLU (Rectified Linear Unit) Activation

The ReLU activation function does the following:

- Introduces non-linearity to the neural network;
- Converts all negative values to zero and keeps positive values as they are;
- Helps the neural network identify complex patterns and perform better than other activation functions;
- Provides faster and more effective training for the neural networks using them.

The ReLU activation function can be used mathematically as follows:

ReLU(x) = Maximum(0, x)

3) Max Pooling Layer

Results in smaller-sized feature maps, which take advantage of transcribed values by taking the highest value from a small-sized area vs picking the highest value in a larger area to reduce computation and memory requirements, as well as making small movements in the location of the data. Purpose: Overall, the intention is to preserve significant features while making it simpler and smaller.

4) Fully Connected (Dense) Layer

- Connects each extracted feature with its respective neuron

- Recognizes the larger associations created by these connections
- Responsible for determining the outcome of events
- Creates a single vector representing all of the feature maps
- Goal: To combine all of the learned features into one final classification output.

5) Softmax Output Layer

- The final layer in CNN takes the output values and converts them into probabilities.
- Each probability will correspond to a different gesture (e.g., Point).
- The final gesture will be the one with the highest probability score.
- Using this method should lead to the most accurate prediction of the final gesture label.

5. Experimental Results [13]

A CNN model was used to evaluate the proposed Sign Language Recognition System, which was developed using a labeled dataset of gestures. Performance was determined by the recognition of static hand gestures, with the system improving steadily throughout the training process. Testing using a webcam in a real-time manner demonstrates that the system is able to recognise gestures quickly and reliably, with minimal delay. Accuracy, efficiency, and effectiveness were all demonstrated within the context of assistive communication application development within a real-time environment.

Fig. 3. Demo Output of the System

6. Conclusion

In this deep learning project, sign language recognition has been achieved via computer vision and convolutional neural networks (CNNs). This means that the goal of this project is to assist in bridging the gap that exists between people who have a hearing disability and those who do not know how to sign by creating an SLR solution using computer vision techniques.

The project structure is laid out in steps that include the following: collecting the data, detecting the hands and identifying the keypoints, preparing the data, extracting features from the data collected, classifying the gestures, and generating text output.

Feature extraction based on hand landmarks through MediaPipe was used to identify and filter the background noise so that the focus was only on the relevant data from and of the hand. This added to the speed of the gesture recognition and improved accuracy of the gesture recognition.

Accordingly, this research has demonstrated that the CNN model is capable of learning and identifying complex hand gestures accurately.

The system performs remarkably well under normal lighting, as well as providing real-time text output, thus making it very useful as an aid for communication. Because of the use of deep learning, the system does not require the addition of extra sensors or specialized hardware, resulting in a simple means of using the system and also providing a cost-efficient solution.

Even though the system has performed very well at recognizing static hand gestures, it does have an identified limitation with regard to its sensitivity to changes in ambient light, and there is no support provided for the recognition of continuous sentences.

The next step in the development of the system for future work will include addressing these limitations by using a time-based approach using LSTM models, enlarging the database used for training purposes, and creating separate models for recognizing different signed languages.

7. Future Scope

- Change Gesture recognition to Dynamic Gesture recognition: Enhance the Gesture recognition system by creating LSTM or GRU models to track entire sequences of gestures and interpret them as fluid.
- Change Single Letter Recognition to Sentence recognition: Enhance the Gesture recognition system to recognize complete words and sentences, and not just a letter.
- Change Single Sign Language Support to Multiple Sign Language Support: Add other sign languages to the Gesture recognition system (ASL, ISL, BSL) to make it accessible to users all over the world.
- Change No Integration of Facial Expressions & Body postures to Include Facial Expression & Body Posture analysis: Analyze a person's facial expressions and body postures along with their gestures to increase the reliability of gesture recognition.
- Change No Speech Output Generation to Add Speech Output: Include a form of text-to-speech that will read the recognized gestures aloud.
- Change Not Available on Mobile or Web to Available on Mobile or Web: The Gesture recognition system should be available via mobile apps or on the web for user convenience.
- Change Low Accuracy in Poor Lighting & Busy Backgrounds to Increased Performance in Poor Lighting & Busy Backgrounds: By using advanced data processing and enhancement techniques, Gesture recognition will have greater performance in both poor lighting and busy/packed backgrounds.
- Change Slow Real-time Performance to Improved Real-time Performance: Improve the responsiveness of the Gesture recognition System by having it operate faster, so as to provide no perceptible delay to users.
- Change Difficult to Use or Interactive to Develop a Simple, User-friendly Interface: Create a simple to operate and user-friendly interface that makes it easy to navigate the Gesture recognition System.
- Change Limited Use for Healthcare/Education to Limited Applications in Education/Healthcare: The Gesture recognition System has applications in both Education and Healthcare facilities.

8. Reference

1. S. Mitra and T. Acharya, “Gesture recognition: A survey,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, vol. 37, no. 3, pp. 311–324, May 2007.

2. C. Valli and C. Lucas, *Linguistics of American Sign Language: An Introduction*, 5th ed. Washington, DC, USA: Gallaudet University Press, 2011.
3. T. Starner and A. Pentland, "Real-time American Sign Language recognition from video using hidden Markov models," in *Proc. Int. Symp. Computer Vision*, 1995, pp. 265–270.
4. P. Kumar, R. K. Aggarwal, and S. Kumar, "Vision-based hand gesture recognition using deep learning," *IEEE Access*, vol. 8, pp. 125678–125689, 2020.
5. Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
6. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2012, pp. 1097–1105.
7. J. Rekha, J. Bhattacharya, and S. Majumder, "Shape, texture and motion features for hand gesture recognition," *International Journal of Computer Applications*, vol. 43, no. 15, pp. 14–19, 2012.
8. V. Bazarevsky et al., "BlazePose: On-device real-time body pose tracking," *Google AI Blog*, 2020.
9. S. Kushwaha, "Indian Sign Language Dataset," *Kaggle*, 2021.
10. G. Bradski and A. Kaehler, *Learning OpenCV: Computer Vision with the OpenCV Library*. Sebastopol, CA, USA: O'Reilly Media, 2008.
11. A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. Advances in Neural Information Processing Systems (NeurIPS)*, 2019.
12. K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2016.
13. D. Powers, "Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation," *Journal of Machine Learning Technologies*, vol. 2, no. 1, pp. 37–63, 2011.
14. K. Cho et al., "Learning phrase representations using RNN encoder-decoder for statistical machine translation," in *Proc. EMNLP*, 2014.
15. Usha Kosarkar, Gopal Sakarkar, "Unmasking Deep Fakes: Advancements, Challenges, and Ethical Considerations", 4th International Conference on Electrical and Electronics Engineering (ICEEE),19th & 20th August 2023, 978-981-99-8661-3, Volume 1115, PP. 249-262, https://doi.org/10.1007/978-981-99-8661-3_19
16. Usha Kosarkar, Gopal Sakarkar, Shilpa Gedam, "Deepfakes, a threat to society", International Journal of Scientific Research in Science and Technology (IJSRST), 13th October 2021, 2395-602X, Volume 9, Issue 6, PP. 1132-1140, https://ijsrst.com/IJSRST219682