2023 Advanced Computing and Communication Technologies for High Performance Applications (ACCTHPA) | 979-8-3503-9844-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/ACCTHPA57160.2023.10083349

# Real-Time Sign Language Recognition System

Sanjukta Sen*, Shreya Narang†, Dr. Gouthaman. P‡
*Final Year B. Tech Student, sanjukta382@gmail.com
†Final Year B. Tech Student, shreya.narang2000@gmail.com
‡Assistant Professor, gouthamp@srmist.edu.in

Department of Networking and Communications (NWC)
School of Computing, Faculty of Engineering and Technology

SRM Institute of Science and Technology, SRM Nagar, Kattankulathur-603203,
Kanchipuram, Chennai, TN, India

***Abstract*—Hearing and speech impairments affect more than 36 million of the world's population. A sign-language recognition system is a crucial step towards the enhancement of communication among visually impaired people. The main idea behind this project is to create a real-time sign language recognition deep learning model which can work on pre-trained image and video dataset and give us the output in real-time. Our system would take the input as an ASL alphabet and save it and then show the string in text or audio format according to the user's choice. It involves accurate extraction of hand gestures using appropriate sensing devices in our model for the smooth communication between the normal and visually-impaired people. This project can serve as a perfect service to the educational industry.**

## I. Introduction

Communication is a fundamental requirement for societal survival. Differently abled people in our society face numerous difficulties in communicating with their surroundings and expressing their feelings on a daily basis. They are unable to communicate with the outside world and face numerous challenges in their daily lives.

Classification of images and machine learning can assist computers to recognize sign language, which can then be interpreted by the mass. This concept employs a pipeline that receives input from a user signing a gesture through a web camera and then generates sign language options for each gesture by extracting various video frames. The Sign Language Recognition deep learning model can work on pre-trained images and video data set and produce the desired real- time output using Python and OpenCV. As the Deep Learning and Neural Networks technology are advancing with every passing day, people are using these technologies to enhance detection system. Image processing is computerized processing of images, which includes collecting, analyzing, processing, and comprehending the results. Computer vision uses a combination of low-level image processing (such as reducing noise and boosting contrast) and higher-level pattern recognition and picture interpretation to identify characteristics in an image. The objective of this project is to develop an app for those who are visually impaired or have other forms of disability in our society. The project's main objective is to look at the best ways to encourage and motivate visually impaired people to move around independently both indoors and outside.

## II. Related Work

The topic, being vast and given the condition that quite a few solutions have been proposed to address the need of sign language detection, needed a lot of study of the previously proposed solutions. Hence, we compared a number of previously published research papers, skimmed through their work to find out their key features and drawbacks so as to address them in our model to build a one-stop-shop for the physically unfortunate section of our society.

### *A. Real Time Sign Language Detection*

In [1], the Sign Language Detection Model captures hand gestures through webcam using OpenCV. After capturing the image, they used pre-trained model SSD Mobile net V2 for recognition of the sign language gesture. They used Convolution Neural Network (CNN) for handling 2D and 3D data as input and for capturing the Temporal and Spatial aspects of an image with the help of appropriate filters.While their major achievements include accurate results under controlled light and intensity, wrong result in very low light is also one of their major drawbacks. Their model makes it simple to incorporate unique gestures, and photographs shot at various angles and frames will increase the model's accuracy.

### *B. Sign Language Recognition Application Systems for Deaf-Mute People: A Review Based on Input- Process-Output*

In [2], they created a sign language recognition system that categorises static signs and alphabets, enabling the system to recognise dynamic movements that appears in a series of photos. They used a number of advanced mechanisms and tools such as HMM, 3D CNN  Microsoft Kinect in order to gain maximum accuracy for their model. The major achievement of their project includes successful recognition of dynamic signals and high accuracy due to use of efficient and vast methods and algorithms. The only drawback is that their model is based on a very small vocabulary of words.

### C. Real Time Conversion of Sign Language to Speech and Prediction of Gestures using Artificial Neural Network

In [3], alongside sign language detection and conversion of the signs into speech, it also aims to predict the need of the mute people. With the help of Artificial Neural Network, [3] has trained the model to predict certain needs of mute person at a particular time of the day. In [3], two layers of algorithm was implemented in which they verified and predicted symbols which are similar to each other. This model also predicts display sentences more appropriately than letter labels. The model uses hardware like Arduino Uno and Flex sensors apart from just the software application for prediction of the mute people's need and for gesture detection.

### D. A Survey on Sign Language Recognition with Efficient Hand Gesture Representation

The method that they developed in [4] takes a fresh approach to hand number gesture recognition by identifying or labelling extensive images of hand parts.

The primary benefit of their solution is the ability to identify the condition of each finger based on the identified hand parts. This makes it simpler to recognize hand gestures that represent numbers. They used Artificial Neural Network, Thinning Algorithm, Neural Network Rules, and Support Vector Machine to create a low-cost, high-performance system. While finding a solution for Number Recognition System besides recognizing alphabets are their major achievement, their model faces a major drawback is capturing gestures in controlled lighting and disturbances in the surrounding.

### E. A Robust Sign Language Recognition System with Multiple Wi-Fi Devices

In [5], their proposed solution includes a Wi-Fi signal-based indoor recognition system that employs two receivers to enhance the performance of recognition. The prediction results from each Wi-Fi receiver are combined in this method to provide the final recognition result. This Wi-Fi-based sign language identification system is more useful in actual situations because to the suggested solution's ability to lower false positives and increase recognition accuracy. The drawback of the model described in paper [5] is that, according to their experimental findings, this system can achieve a higher recognition accuracy of 93.8 percent and a lower mean false positive rate of 1.55 percent when compared to the original implementation, which makes use of a single laptop in the same setting.

## III. System Analysis

### A. Programming Language Used:

Python: The development of data set is carried out in Python. Open CV and Tensor Flow are the modules that Python is used with. The algorithm used in Python is CNN, and the library that was extracted is called NumPy.

### B. Algorithm Used:

Convolution Neural Network: For recognizing signs and gestures, our system uses CNN as its algorithm. To extract features from the frames and forecast hand gestures, a CNN model is used. It is a multiple-layered feed-forward neural network that is primarily utilized for image recognition. This algorithm was chosen because it excels at image classification and pattern recognition tasks and is extremely precise.

### C. Software And Libraries Used:

OpenCV: Using OpenCV, we obtain the live camera feed and create a Region of Interest (ROI) that corresponds to the portion of the region in which hand gestures needs to be detected.

TensorFlow-Lite: TensorFlow Lite is a set of tools that helps developers run their models on mobile, embedded, and edge devices, enabling on-device machine learning.

NumPy: The core Python package for scientific computing is called NumPy. It is a Python library that offers a multidimensional array object. NumPy was utilized for the extraction of shape manipulation, random stimulation, and calculation analysis.

### D. Course Of Action:

Calculating Accumulated Weight: For the purpose of separating the foreground from the background, we calculate the background's accumulated weighted average and then this is removed from the frames which include an object that can be identified as the foreground in front of the background. By computing the cumulative weight for several frames, we determine the accumulated average for the background (in this case, 60 frames).

Calculating Threshold Value: With the aid of cv2, we identify the contours and figure out the value of threshold for each frame. Utilization of segment function, helps in finding the contours and return the highest contours (the object's extreme contours). We can identify whether a hand is there or if any foreground objects have been picked up in the ROI using the contours. Using ROC Curves and Precision-Recall Curves, one can directly determine the best threshold for the CNN. To fine-tune the threshold and determine the ideal value, we use a grid search.

Training CNN TensorFlow Model: To train the model, Convolution Neural Networks are used. We will fix the model in such a way that it trains the batches using the call-backs for ten epochs (the number of epochs may change depending on the user's parameter selection).

Predicting Gesture: A bounding box is created to identify the reference of interest followed by the calculation of the overall average, just as we did when we were constructing the data set. This is done to identify any object in the foreground.

Using the predefined functions, we develop a model for gesture recognition that first recognizes the hand and then detects the sign language.

## IV. Proposed Methodology

### A. Image Acquisition

The camera on a smartphone or laptop is used to record the signer's motions. The camera must be positioned straight at the signer in order to record the front view of their hand motions. The gestures are being recorded as a video sequence, which is then being converted into Image frames.

### B. Data set Creation

Before the recognition system is built, a data set containing all the American Sign Language Alphabets are being fed in the model. It includes all the alphabets between A-Z and all the numb er between 0-9. The data set is built by training the model with probable basic gestures so that when the signer shows any gesture, the train model can quickly find it's meaning from the data set and show the output.

### C. Pre-processing

To fulfil memory needs and control environmental scene circumstances, the content must be pre-processed. It helps avoid local changes due digitization errors and noise that can alter the image information. Various factors like illumination, background, camera location and angle often affect the images dramatically. Pre-processing of the raw video footage is crucial in order to meet the ambient scene conditions. Filtering is the pre-processing block's first and foremost step. The undesired noise from the image scenes is eliminated using a moving average or median filter. The pre-processing block's next important step is background subtraction. The running Gaussian average method is used to create the background subtraction since it is quick and uses less memory than other techniques.

### D. Hand Detection

Detecting the hand is an important step in order to produce the desired output. With the help of OpenCV, the gesture made with the hand can be recognized. The hand movement and shape play the major role in guessing the word or message the signer wants to communicate. For accurate hand detection certain environmental condition like clear background and proper illumination is necessary.

### E. Feature Extraction

The object recognition method requires feature vectors to initiate the acquisition process. Features are divided into 2 categories: movement and outline shape of the hand. The Point of Interest(POI) features of the hands reveal the state of the hand movements. To illustrate the "shape"and "direction of movement," we consider two POIs as shown in Fig.1.
The feature vector is an N-element single-row, single-column matrix. The feature vector must be calculated, which takes time and memory. The feature vector has 55 characteristics, including 40 elements generated from the wavelet transform of the gesture's Fourier-transformed picture, 5 fingertip elements (A, B, C, D, E, 1 - present, 0- absent), 4 motion vector elements, and 6 moving videos (MV) sequence components.

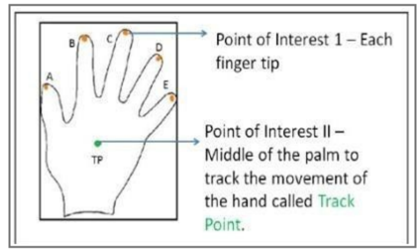


Fig. 1: Point of Interest

### F. Hand Posture Recognition

The hand gesture made by the signer once captured is then processed and matched with the data set to look for the accurate meaning. Since the data set already contains pre-trained data pertaining to the different hand gestures, the output can be displayed only when the hand gesture of the signer matches with the relevant data in the data set.

### G. Display Output as Audio/Text

Once the recognized hand posture finds its relevant meaning from the data set, the meaning of the gesture is then displayed as the outcome. There are 2 methods of displaying the result either through text or audio for easier and effective communication.

## V. System Architecture

### A. Block Diagram

A sign language recognition system block diagram is a diagram in which the stages are represented as blocks and connected by lines that represent the relationship between different blocks. The block diagram begins with image acquisition, where an image from a source such as a camera is captured. This is followed by hand region segmentation, where the shape and movement of the hand are captured, and then hand detection and posture recognition occur. The captured gesture is then classified by comparing it to the database's data set. Once a gesture is identified, it can be converted into audio or text format.

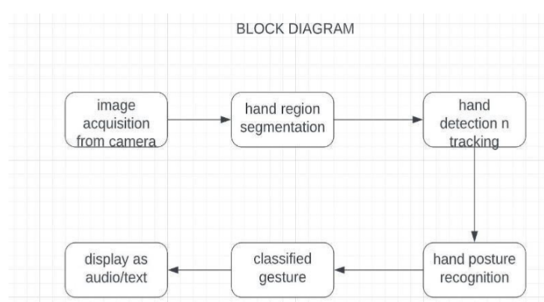


Fig. 2: Block Diagram

### B. Architecture Diagram

An architectural diagram of sign language recognition system is a visual representation that maps the physical elements and the different stages involved in the system. The system begins by capturing the alphabet with a webcam while simultaneously training the data set. The system then starts preprocessing the input image while synchronously can save the data set model. Convolution neural network is the algorithm used in our structure. The character is found in the saved model after the algorithm is applied to the image and the data set, and once the captured alphabet/gesture is detected, it is further converted into the output format, which is audio or text.

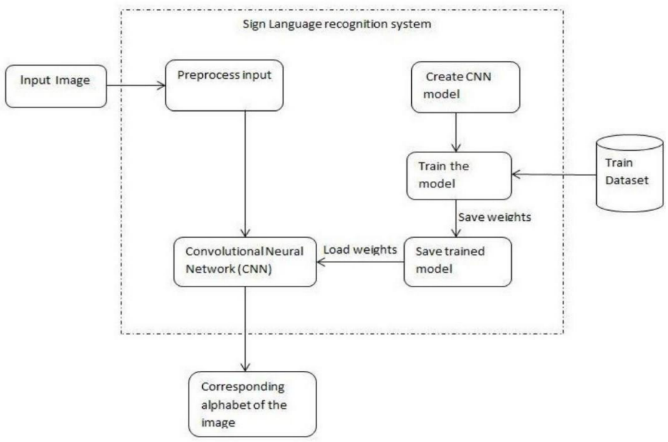


Fig. 3: Architecture Diagram

## VI. Implementation

The dataset is created by training the AI Model with the different alphabets from A to Z in ASL and the numbers from 0 to 9. The image acquisition takes place through the front camera of the computer. After the image is captured using OpenCV, the trained model recognizes the hand gesture,performs feature extraction, and matches the sign shown on the camera with the data in the dataset. As the gesture matches with the data, the output is being displayed.

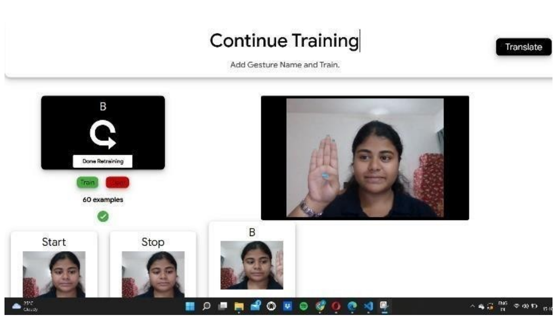


Fig. 4: Training Dataset

## VII. Result and Discussions

A Real-time Sign Language Recognition System has been made using Convolution Neural Networks, OpenCV and Python for the accurate detection of gesture and signs under suitable surroundings and conditions. The model is capable of learning all the 26 alphabets from A to Z, numbers from 0 to 9, and simple words like yes, no, hello, thank you etc. The model can display output in two formats, including text and audio. The text format is accessible to the mute and the audio format is accessible to the blind. It includes a very simple design and algorithms to address a huge problem of the society. Here,the alphabet B has been trained into the data set and when the hand gesture shows the alphabet B, the model immediately captures the sign and produce the desired result with 100 percentage accuracy.


Fig. 5: Successful Output Received

The output received from the experiment helped us to infer that:

- Our algorithm is able to recognize every alphabet from A to Z, with 80-82 percent accuracy.
- Due to disturbances in the background, we are facing loss of accuracy up to 22 percent.

The overall accuracy of the model is around 82.6 percent which is better than most of the pre-existing models in this field. The output can be observed on computer or mobile in form of text or audio. The precise detection of hand gestures in different limitations and difficult environment has been captured and output with high level of accuracy has been generated.

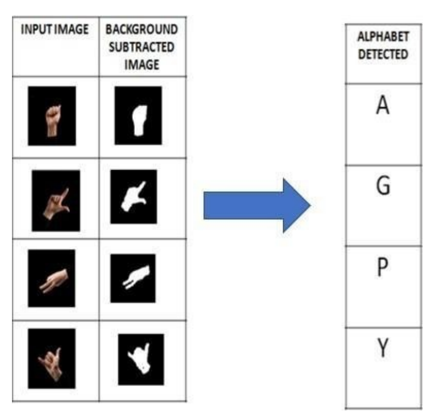


Fig. 6: How sign languages are converted

| Gesture Name | Accuracy |
|---|---|
| Yes | 95% |
| No | 90% |
| Hello | 83.7% |
| Thank You | 86 % |

### A. Performance Analysis

We conducted a tests to check for the performance and the accuracy of our system. This test was performed under different environmental conditions. Hand gestures such as Yes, No and Hello was displayed in front of the camera under unusual conditions. These calculations are automatically generated through python programs. The conditions in which we checked the recognition are: i. Day with proper lightning; ii. Less visible with low quality camera; iii. In a very less illuminating light.

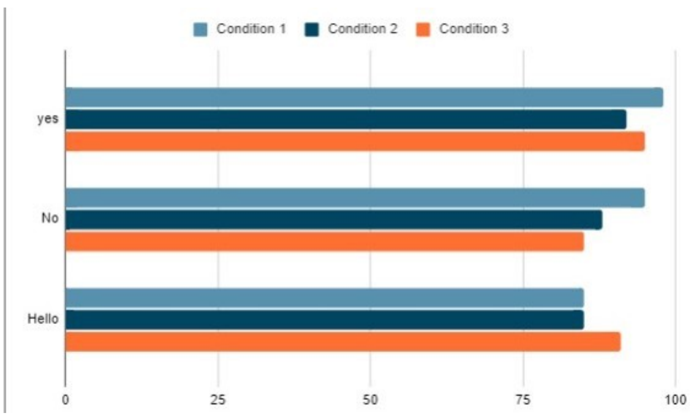


Fig. 7: Accuracy Under Different Environmental Conditions

## VIII. Future Scope:

The dynamic Sign Language Recognition Model has considerable scopes in the future and can be incorporated even for education and business purposes. Some of the future scope for our Real- time Sign Language Recognition Model includes: Training the model for basic signs such as eating, drinking, walking etc. Creating a greater number of classes for numeric sign languages, emotions etc. A chatbot can be incorporated in the application for clarifying user's doubts, assisting them if they experience any issues with the application and also for receiving feedbacks. It can also be used in the education sector to help children with hearing and speech impairments to communicate and learn through sign languages.

The app can used in real-world scenarios like job interviews, meetings where the person can directly contact without needing any professional help. We are also planning to integrate our model with applications like Google Meet, Zoom, etc.

## IX. Conclusion

This project's primary objective is to address the societal problem of deaf and mute individuals by bridging their communication gap with hearing individuals. The basic requirement one needs to use this project is a web cam on a laptop or a mobile phone. OpenCV is being used to capture the hand gesture of the signer and the data is fetched from the trained dataset model to predict the output. From the results, we can conclude that the model which we made shows accuracy up to 82.6 percent under any environmental condition and even uncontrolled lighting which addresses the major drawback in most of the solutions that have been proposed for this problem statement. This software has a rapid processing rate and can generate results in real time. Major difficulties which we faced and observed while making the project was that recognizing the gesture largely depends on the camera quality and proper angle of the hand.

## References


[1] Real Time Sign Language Detection- Aman Pathak, Avinash Kumar, Priyam, Priyanshu Gupta, Gunjan Chu- International Journal for Modern Trends in Science and Technology, 8(01): 32-37, 2022.

[2] Suharjito, R. Anderson, et al. "Sign language recognition application systems for deaf-mute people: a review based on input-process-output." Procedia Comput. Sci 116 (2017): 441-448.

[3] Abraham, Abey, and V. Rohini. "Real time conversion of sign language to speech and prediction of gestures using Artificial Neural Network." Procedia computer science 143 (2018): 587-594.

[4] Al-Hammadi, Muneer, et al. "Deep learning-based approach for sign language gesture recognition with efficient hand gesture representation." IEEE Access 8 (2020): 192527-192542.

[5] Shang, Jiacheng, and Jie Wu. "A robust sign language recognition system with multiple Wi-Fi devices." Proceedings of the Workshop on Mobility in the Evolving Internet Architecture. 2017.

[6] Vij, Sugakshi, and Vivek Kumar Sehgal. "Sign Language Recognition Using Python and OpenCv Project." (2021).

[7] Mekala, Priyanka, et al. "Real-time sign language recognition based on neural network architecture." 2011 IEEE 43rd Southeastern symposium on system theory. IEEE, 2011.

[8] López, Iker Vázquez. Hand Gesture Recognition for Sign Language Transcription. Diss. Boise State University, 2017.

[9] Jain, Sanil, KV Sameer Raja, and Mentor-Prof Amitabha Mukerjee. "Indian sign language character recognition." Indian Institute of Technology, Kanpur Course Project-CS365A (2016).

[10] Goyal, Sakshi, Ishita Sharma, and Shanu Sharma. "Sign language recognition system for deaf and dumb people." International Journal of Engineering Research Technology 2.4 (2013).

[11] Nikam, Ashish S., and Aarti G. Ambekar. "Sign language recognition using image based hand gesture recognition techniques." 2016 online international conference on green engineering and technologies (IC-GET). IEEE, 2016.

[12] Ma, Yongsen, et al. ”SignFi: Sign language recognition using WiFi.” Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 2.1 (2018): 1-21.

[13] Bhagat, Neel Kamal, Y. Vishnusai, and G. N. Rathna. ”Indian sign language gesture recognition using image processing and deep learning.” 2019 Digital Image Computing: Techniques and Applications (DICTA). IEEE, 2019.

[14] Chavan, Shruti, Xinrui Yu, and Jafar Saniie. ”Convolutional neural network hand gesture recognition for American sign language.” 2021 IEEE International Conference on Electro Information Technology (EIT). IEEE, 2021.