

# Hand Gesture Recognition System for Hearing and Speech Impaired People Using ML

G. K. Dayananda\*, I. R. Manjunath, J. G. Sreerama Samartha, M. Vayusutha, D. N. Pratham, Sakshath, A. U. Shashank, Swasthi Prasad Shetty

**Abstract:** Sign language has historically served as a primary form of communication for the deaf community. However, the lack of sign language proficiency among the general population and the limited availability of interpreters create barriers in communication between non-sign language speakers and deaf individuals. To address this issue, a proposed system aims to bridge this communication gap by assisting non-sign language speakers in recognizing and understanding American Sign Language (ASL) gestures. This system has the potential to enhance greatly communication between deaf individuals and the wider society. The process involves enhancing the quality and clarity of hand gestures by passing them through a filter. Subsequently, a classifier analyses the filtered hand gesture and predicts its corresponding meaning or category. This classification step enables non-sign language speakers to interpret accurately the intended message conveyed through these hand gestures. It is important to note that this method specifically focuses on recognizing the 26 letters of the ASL alphabet. Analysis of various datasets reveals that the enhanced network performs better than Faster R-CNN in terms of gesture categorization accuracy. Specifically, when using 175 photos for each move, the upgraded network achieves an 83% recognition accuracy, surpassing the performance of Faster R-CNN in gesture classification. Furthermore, with 600 images for each gesture, the improved network achieves an accuracy of 97. %. This finding suggests that employing a TOF camera to collect images based on depth information effectively reduces the influence from other elements during feature extraction. Overall, this system primarily focuses on recognizing fingerspelling gestures, which represent individual letters in ASL. It is commendable that efforts are being made to utilize neural networks and technology to enhance communication for individuals with disabilities.

**Keywords:** ASL; feature extraction; gesture recognition; hand gesture; image recognition; speech and hearing impaired

## 1 INTRODUCTION

Sign language has historically served as a primary form of communication within the deaf community. However, the limited knowledge of sign language among the general population, as well as the scarcity of interpreters, presents obstacles in bridging the communication gap between non-sign language speakers and deaf individuals. In an effort to address this issue, a proposed system aims to facilitate communication by enabling non-sign language speakers to recognize and comprehend American Sign Language (ASL) gestures [1]. This system has the potential to enhance greatly communication between deaf individuals and the wider society.

The methodology involves subjecting the hand gesture to a filtering process in order to improve its quality and clarity [2]. Subsequently, a classifier analyses the filtered hand gesture and predicts its corresponding meaning or category. This classification step allows non-sign language speakers to interpret accurately the intended message conveyed through the hand gestures. It is important to note that this methodology specifically focuses on recognizing the 26 letters of the ASL alphabet [3].

A visual language, sign language has three main components as shown in Fig. 1:

**Manual Signs:** Manual signs are the core elements of sign language. They involve the use of handshapes, movements, and orientations to convey meaning. Each sign represents a word, concept, or idea. Manual signs can be simple or complex, depending on the language and the specific sign being used.

**Facial Expressions:** Facial expressions play a crucial role in sign language as they provide grammatical and emotional information. They help convey nuances, such as questions, statements, negations, and emotions. Facial

expressions, including eye gaze, brow movement, and mouth patterns, add depth and clarity to the signs being produced.

**Body Language and Postures:** Body language and postures complement the manual signs and facial expressions in sign language. They contribute to the overall meaning and interpretation of the message. Body movements, such as the position of the torso, shoulders, and arms, along with head movements and body orientation, provide additional contextual information and convey the speaker's emphasis, location, and spatial relationships.

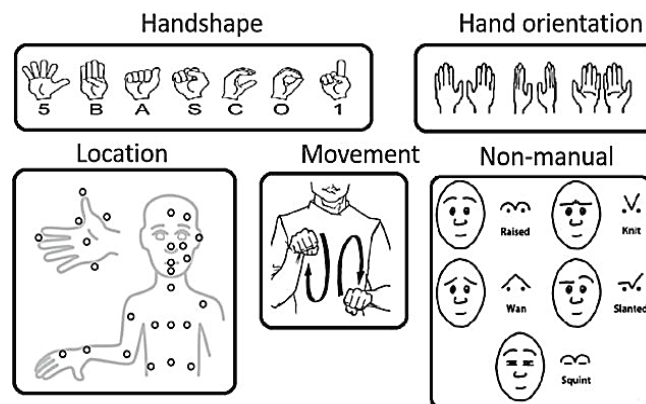


Figure 1 Sign Language major component [2]

Sign language is widely regarded as a vital means of communication and can be used to communicate with others who do not speak sign language, as well as people who have hearing or speech difficulties. Sign language, as opposed to spoken language, communicates meaning through physical gestures. It includes facial emotions, hand gestures, and movements of other body parts such as the eyes, legs, and so on. This study presents a design for recognising and interpreting ASL signs. Social contact, communication differences, education, behavioural issues, mental health

issues, and safety concerns are just a few of the challenges that speech and hearing-impaired people encounter while engaging with hearing-impaired people. A keyboard, mouse, or audio signal can be used to interface with a computer. The former demands constant physical contact, whilst the latter is sensitive to noise and distractions. A gesture is any physical movement done with the hand, eye, or other body part. The best motions for people to use and understand are hand gestures. Fig. 2 shows ASL Sign Language Gesture by hand.

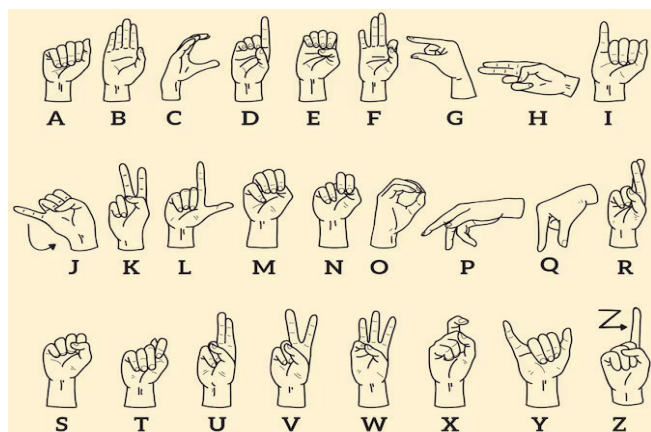


Figure 2 ASL Sign Language Gesture

## 2 LITERATURE SURVEY

In today's technological landscape, it is becoming increasingly common for electronic devices to be controlled without physical touch. This is primarily due to the rise of voice and gesture controls. Voice controls are capable of recognizing spoken words, enabling users to operate devices through verbal commands. On the other hand, gesture controls involve moving one's hand in specific patterns to execute desired tasks on the device [4]. Extensive research has been conducted to develop efficient and rapid hand gesture recognition systems [5]. Numerous experiments have been carried out to identify hand gestures, and these programs are often integrated into other applications, such as device control through hand gestures. Various methods exist for recognizing hand gestures from images.

Algorithms for recognizing hand gestures, such as multivariate fuzzy decision trees [6], Hidden Markov Models (HMMs) [7], dynamic time warping framework [8], latent regression forests [9], support vector machines, and surface electromyograms [10], are utilized to analyse and process these gestures. The captured movements are examined to identify occlusion and interactions involving close fingers, enabling the recognition of the correct gesture while disregarding intermittent gestures. Robust algorithms like HMM are necessary for real-time hand gesture detection that accurately identifies the intended gesture. The effectiveness of categorized gestures is evaluated using the K<sup>th</sup> datasets and sign language alphabets as standard training and validation datasets. Hand gesture recognition plays a crucial role in various applications, including music orchestration, robotics, television control, rehabilitation, and sign language recognition.

This is the initial and most crucial step in the entire sign reformation process [11]. In order to capture photos using a webcam, camera interfacing is necessary. Nowadays, many laptops come with built-in cameras, which simplifies the process of capturing photos for later processing. Morphological operations involve shapes of images [12]. The two primary morphological processes are erosion and dilation, which are employed for noise reduction, element separation, element alignment, and even intensity fluctuation or gap detection in images. While dilation can fill in any gaps left by erosion or add pixels to boundary areas, erosion can reduce noise by widening gaps and reducing image boundaries. Dilation can also be used to link randomly placed pixels and add pixels to the boundaries. In order to construct a contour around the palm and identify finger positions, the convexity hull algorithm [13] is used in contour detection. Adaptive boosting is used in the convexity hull technique to detect hands, and the HAAR classifier algorithm [14] can be used to train classifiers. Segmenting the area of the image that contains the hand is the first step in the convexity hull algorithm. It is necessary to make some assumptions about the hand's characteristics, even if they could vary as the hand travels.

The gesture of the hand is classified based on the maximum score, which represents the minimum Euclidean distance between the input gesture and the dataset image of the corresponding character. The character associated with the input gesture is determined based on this minimum distance. It is important to note that the process described is a basic outline and does not include details of the actual segmentation and feature extraction techniques used. Depending on the complexity of the problem and the dataset, more advanced methods like those that deep learning-based approaches may be used for improved accuracy and efficiency. Additionally, the mention of "detecting human hands against white backgrounds" suggests a potential improvement in image pre-processing to enhance accuracy and speed by using a background subtraction technique to isolate the hand region. This could be beneficial when the images are consistently captured against a consistent background. Finally, the process involves two main steps: training and recognition [15]. During training, the model learns from a labelled dataset of hand gestures. During recognition, the learned model is used to classify new, unseen hand gestures based on their features and the calculated Euclidean distances.

To ensure accurate feature comparison and minimize errors during image processing, the system scales all the captured photographs to the same size. This step is essential when dealing with images captured using different camera resolutions, as it brings uniformity to the dataset and helps in achieving consistent results during recognition. Overall, the proposed system presents a comprehensive approach to assist disabled individuals in expressing themselves through sign language. By leveraging vision-based techniques [16] such as Histograms of Oriented Gradients (HOG) feature extraction and Euclidean distance-based classification, the system can recognize hand gestures and provide meaningful translations for communication.

### 3 SYSTEM REQUIREMENT

The identification of hand gestures has been the subject of extensive research in recent years. We discovered the following are the fundamental steps in hand gesture recognition with the aid of a literature review.

#### 3.1 Relevance of the Work

Individuals who are deaf and unable to speak communicate using hand sign language. However, communicating with individuals who do not understand sign language can be challenging. To address this issue, we are currently developing a program that aims to facilitate communication between deaf individuals and those who do not understand sign language. Our program will detect and translate the hand signs made by the user into text, which will then be displayed on a screen for non-sign language users to read. By implementing this project, we aim to enhance communication capabilities for deaf individuals when interacting with non-sign language users.

##### 3.1.1 Proposed Work

Different methods can be employed to gather data on hand gestures. One approach involves using electromechanical devices to capture accurately the configuration and position of the hand. Alternatively, glove-based methods can be utilized to extract information [17, 18], although these tend to be expensive and not user-friendly. Another option is to rely on vision-based techniques, where a computer camera serves as the input device to observe hand or finger movements. Vision-based methods offer a natural interface between individuals and computers, eliminating the need for additional hardware [19]. These systems, which can be implemented in software and/or hardware, simulate biological vision. By utilizing a vision-based strategy, the need for artificial devices in interaction is resolved, as solely hand movements represent all gestures. The design of the Gesture Recognition System is illustrated in Fig. 3.

To create own dataset using OpenCV [20], can follow these steps:

**Install OpenCV:** Install the OpenCV library on your system by following the installation instructions specific to your operating system.

**Set up your camera:** Connect a camera to computer or use the built-in camera of your device.

**Capture frames:** Use OpenCV to capture frames from the camera. Use the VideoCapture class to access the camera and retrieve frames.

**Apply image processing:** Apply the desired image processing techniques to the captured frames. In this case, Mentioned using a Gaussian blur filter and thresholding. OpenCV provides functions like GaussianBlur and threshold that you can use for these operations.

**Extract features:** After processing the image, extract relevant features from the processed image. The specific features you choose will depend on the problem trying to solve..

**Pass processed image to CNN model:** Extracted the features from the processed image, CNN (Convolutional Neural Network) model for prediction.

**Letter detection and word formation:** Keep track of the predicted letters over consecutive frames. If a letter is detected for more than 50 frames, consider it as a valid detection and store it. Whenever the letter changes or a blank symbol is detected, form a new word and store the previous word. Consider the space between words using the blank symbol.

**Repeat the process:** Continuously capture frames, apply image processing, pass the processed image to the CNN model, and detect and form words until collected sufficient dataset.

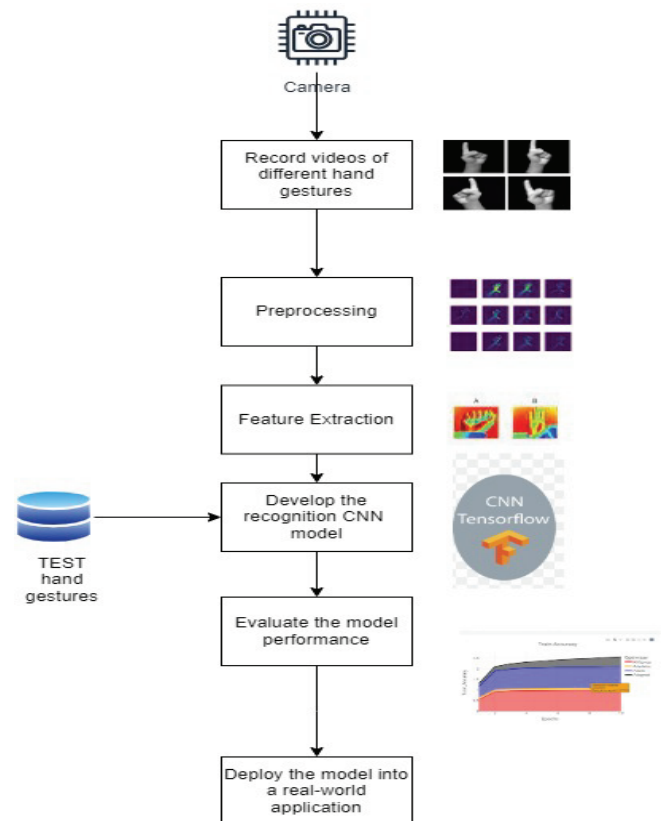


Figure 3 Flow diagram of Gesture Recognition System

### 3.2 Software Requirement

#### 3.2.1 Artificial Neural Network

An Artificial Neural Network (ANN) is a computational model inspired by the structure and functioning of the human brain. It is a type of machine learning algorithm that can learn and adapt from examples, making it suitable for various tasks, including pattern recognition, classification, regression, and decision-making.

#### 3.2.2 Convolutional Neural Networks (CNN) [22]

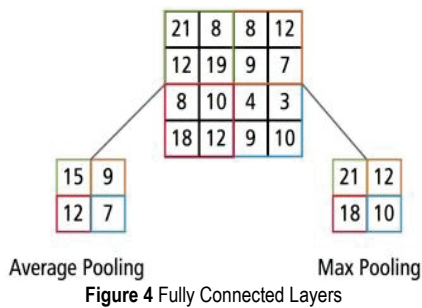
**Convolution Layer:** In this layer, small filters (also called kernels) with learnable parameters are applied to the input image. These filters move across the image to perform

convolutions (dot products) at different spatial positions. The output of this layer is a 2D activation map that represents the response of the filters at each position, capturing specific visual patterns or features like edges or colors.

**Pooling Layer:** After the convolution layer, the pooling layer is used to downsample the activation maps, reducing their spatial dimensions and the number of parameters. Two common types of pooling are max pooling and average pooling. In max pooling, a window (e.g.,  $2 \times 2$ ) slides over the activation map, and only the maximum value within each window is retained, effectively reducing the size of the map by half. In average pooling, the average of all values within the window is taken instead.

**Fully Connected Layers:** After several convolutional and pooling layers, the final stage of the CNN architecture usually involves fully connected layers (Fig. 4). These layers are similar to those in regular neural networks, where each neuron is connected to all neurons from the previous layer. The fully connected layers take the high-level abstract features learned by the earlier layers and combine them to make predictions about the input's class. The number of neurons in the output layer corresponds to the number of classes in the classification task, and the values of the output neurons represent the probabilities of the input belonging to each class.

CNNs have been very successful in computer vision tasks, such as image classification, object detection, and segmentation, because they can effectively learn hierarchical features from raw pixel data. The convolutional and pooling layers help capture local patterns and reduce the spatial dimensions of the data, while the fully connected layers make global decisions based on the extracted features.



### 3.2.3 TensorFlow

TensorFlow is indeed an open-source software library that provides a framework for dataflow programming, especially well-suited for machine learning tasks involving neural networks. Tensor Flow's flexible design allows it to be deployed on different platforms seamlessly. This versatility makes it popular only not for research and development but for production-level machine learning applications. TensorFlow has been a cornerstone in the development and deployment of various machine-learning models, and it has played a significant role in advancing the field of deep learning [22]. Its ease of use, scalability, and compatibility with different hardware have made it a popular choice for researchers and developers alike. To extract different aspects

from an image, a Gaussian blur filter is applied. The resulting image after applying the Gaussian blur creation process is shown below.

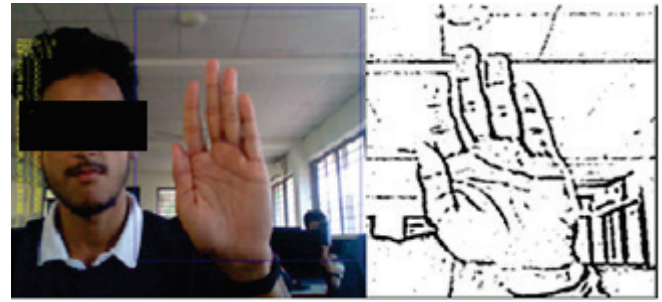


Figure 5 Display for speech to gesture output

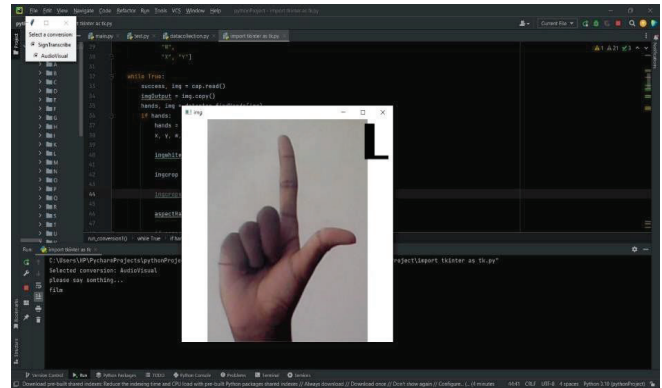


Figure 6 Speech to Gesture output Display

## 4 RESULTS AND DISCUSSION

### 4.1 Data Set Generation

The procedures you described involve using the Open Computer Vision (OpenCV) software to capture images of American Sign Language (ASL) symbols. Create own dataset to train and test a machine learning model for recognizing ASL symbols. This dataset will be used to teach the model to understand and classify different ASL signs. Captured approximately 800 photographs of each ASL symbol for training purposes. Collecting a sufficient number of images for each symbol ensures that the model has enough data to learn and generalize effectively. Additionally, collected about 200 images of each ASL symbol for testing purposes. The testing data is used to evaluate the model's performance and assess its ability to make accurate predictions on new, unseen data. To capture the images, used computer's webcam. This means acquired images in real-time by taking a picture of each frame presented to the webcam. Each frame captured by the webcam contained a specific ASL symbol. To focus the model's attention on the relevant area, defined a Region of Interest (ROI) around the ASL symbol. This ROI was visually indicated by a pink-bordered square. Utilized the OpenCV software for the image capture process. OpenCV is a popular open-source computer vision library that provides various tools and functions for image and video processing tasks. Fig. 7 likely depicts an example of the captured image with the pink-bordered square indicating the ROI around an ASL symbol.

The selection of a suitable filter is also a critical component of any ASL symbol recognition project. The filter that you select will determine the features that are extracted from your images, and the CNN model to identify the symbols will use these features. There are many different filters that you can select and it can be difficult to determine which filter will work best for your project. In your case, you experimented with a variety of filters before settling on the chosen filter.

The accuracy of the model is another important factor to consider. In this case, faced issues with the accuracy of the model in earlier rounds. This was likely because the dataset was not large enough or that the filter was not selecting the correct features. By increasing the input image size and improving the dataset, were able to improve the accuracy of the model.

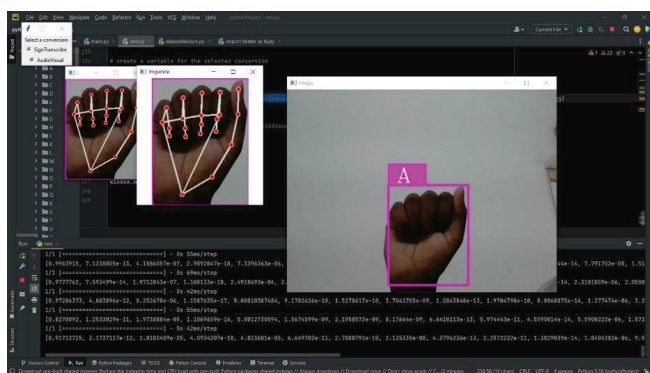


Figure 7 Region of interest (ROI) indicated by a pink-bounded square

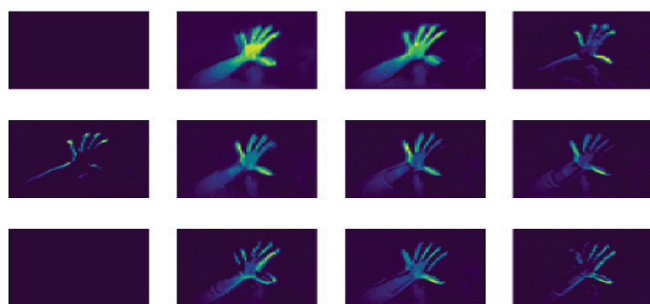


Figure 8 Gaussian blurred image

In order to extract different aspects from our image, we then apply our Gaussian blur filter to it. Fig. 8 displays the image after applying Gaussian blur.

### 4.2 CNN Configuration

The Stochastic Gradient Descent (SGD) algorithm is used to build the model, and a learning rate of 0.002 is used. The categorical cross-entropy function is used to assess the loss because the model is built to handle several classes. Additionally, measures like loss and accuracy are given to efficiently monitor the review process.

This particular arrangement was chosen after experimenting with various blends of nodes and layers. After experimenting with different combinations of nodes and

layers, this specific configuration was chosen. The CNN configuration details can be found in Tab. 1.

Table 1 Details of CNN configuration

Model Content	Details
First Convolution Layer	128 filters of size 3×3, ReLU, input size 72×72
First Max Pooling Layer	Pooling Size 3×3
Second Convolution Layer	128 filters of size 3×3, ReLU
Second Max Pooling layer	Pooling size 3×3
Dropout Layer	Excludes 30% neurons randomly
First Fully connected Layer	512 nodes, ReLU
Dropout Layer	Excludes 30% neurons randomly
Second Fully Connected Layer	512 nodes, ReLU
Dropout Layer	Excludes 30% neurons randomly
Output Layer	12 nodes for 12 classes, SoftMax
Optimization Function	Stochastic Gradient Descent (SGD)
Learning Rate	0.002
Metrics	Loss, Accuracy

### 4.3 Classification Result

This statement is supported by additional performance measures outlined in Tab. 2. As depicted in Fig. 9 and Fig. 10. The tight agreement between training and testing accuracy across each epoch suggests that the use of data augmentation strategies helped the model avoid overfitting.

To evaluate further the model's adaptability, the experiment was repeated using a train-test split of 75-45. Remarkably, the accuracy achieved for this split configuration was 95.2% and 97.5%, indicating the model's ability to perform well even on a larger test set.

Table 2 Details of performance measure

Performance measure	CNN without augmentation	CNN with augmentation
Precision	0.9176	0.9869
Recall	0.9432	0.9834
F-Measure	0.9365	0.9832
Accuracy, %	93.43	97.46

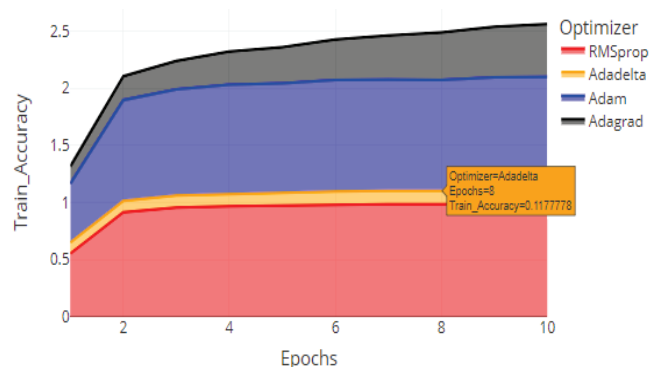
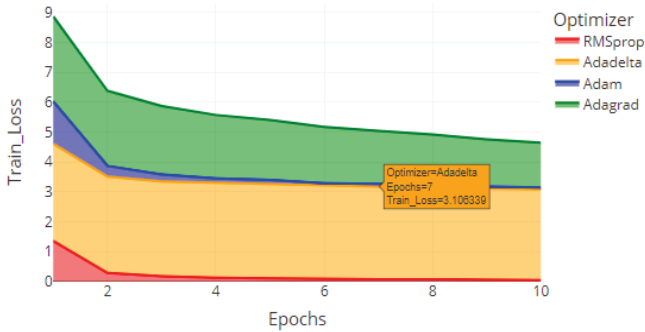


Figure 9 Train Accuracy for various optimizer (Adam and RMSprop shows best accuracies namely 95.2 and 97.5 %)

The training loss values you mentioned for RMSprop (0.071) and Adamgrad (0.0734) as shown in fig 10 indicate that both optimizers are performing similarly in terms of minimizing the loss during training. While there is a slight difference between the two values, it is not significant enough to distinguish easily their performance based solely

on the training loss. When comparing optimizers, it is important to consider multiple factors beyond just the training loss.



**Figure 10** Train Loss for various optimizer (RMSprop 0.071 Adagrad 0.0734 again a close call to distinguish)

Based on the Tab. 3, it appears to contain results for different training epochs, optimizers, activation functions, and evaluation metrics for a specific model. Let us break down each column to understand the information it represents:

**Epochs:** The model underwent two epochs of training.

**RMSprop** was employed as the optimisation algorithm during the training process.

**Activation\_Function:** The layers of the neural network made use of the ReLU activation function.

After two training epochs, the training loss was 0.28154. This shows the typical discrepancy between the training dataset's predicted and actual values. Better performance is indicated by lower values.

**Train\_Accuracy:** After two epochs, the training accuracy was 0.91381. It displays the proportion of samples in the training set that were correctly categorised.

**Train\_Precision:** After two epochs, the training precision was 0.23096. Precision is the percentage of all positively anticipated instances in the training set that were accurately predicted.

**Train\_Recall:** After two epochs, the training recall was 0.99738. Recall quantifies the percentage of positive instances in the training set that were properly predicted out of all the actual positive instances.

**Val\_Loss:** After two epochs, the validation loss was 0.26574. This shows the typical discrepancy between the validation dataset's predicted and actual values. Lower values imply more accurate generalisation.

**Val\_Accuracy:** After two epochs, the validation accuracy was 0.98667. It shows the proportion of samples in the validation set that were correctly categorised.

**Val\_Precision:** After two epochs, the validation precision was 0.99982. Precision is the percentage of all positively predicted cases in the validation set that were accurately predicted.

**Val\_Recall:** After two epochs, the validation recall was 0.043908. Measures the percentage of correctly predicted favourable outcomes.

These values provide insights into the model's performance at the specified training stage. The table provides a snapshot of the model's performance metrics at different training epochs, displaying how the optimizer, activation function, and training progress influence the model's accuracy, loss, precision, and recall values.

**Table 3** Detailed results

Epochs	Optimizer	Activation_Function	Train_Loss	Train_Accuracy	Train_Precision	Train_Recall	Val_Loss	Val_Accuracy	Val_Precision	Val_Recall
2	RMSprop	relu	0.28154	0.91381	0.23096	0.99738	0.043908	0.98667	0.26574	0.99982
7	RMSprop	softsign	0.07608	0.97992	0.25867	0.99937	0.130949	0.97315	0.31932	1
10	RMSprop	softsign	0.06242	0.98294	0.27277	0.99944	0.019199	0.99519	0.4014	0.99963
3	RMSprop	softsign	0.18316	0.94548	0.23709	0.99921	0.064611	0.97593	0.26	1
1	RMSprop	elu	1.41317	0.52857	0.18546	0.89111	0.289114	0.92093	0.23752	0.99926

In scenarios where there are limited datasets, particularly with only 175 images for each gesture, both networks achieve a recognition rate exceeding 96.8%. This observation suggests that using a TOF camera to collect images based on depth information effectively mitigates the interference caused by other factors during feature extraction.

As the number of datasets increases, the data recognition rate also improves. Augmenting the datasets enhances the network's resilience against complex environmental disturbances. A higher recognition rate establishes a solid foundation for subsequent classification tasks.

Based on the information presented in Fig. 9 and Fig. 10 it can be observed that the improved network outperforms

Faster R-CNN in terms of gesture classification accuracy across various datasets. Specifically, the improved network achieves a recognition accuracy of 83% when working with 175 images for each gesture. This accuracy surpasses the recognition performance of Faster R-CNN, indicating the superiority of the improved network in gesture classification.

The improved network demonstrates notable improvements in accuracy compared to the traditional network. When working with 300 images for each gesture, the improved network achieves a recognition accuracy of 94.8%. Remarkably, this accuracy is comparable to the recognition accuracy of the traditional network trained on 100 times the amount of data.

Furthermore, when there are 600 images for each gesture, the improved network achieves an accuracy of 97.5%, slightly surpassing the performance of Faster R-CNN. This demonstrates the effectiveness of the proposed network in achieving higher accuracy, regardless of whether the dataset is small or large.

## 5 CONCLUSION

In this project, a practical ASL (American Sign Language) recognition system has been developed specifically for individuals with hearing and speech impairments. The system utilizes real-time vision technology to identify accurately ASL alphabets. By implementing two layers of algorithms, the system is able to effectively predict and validate symbols that bear close resemblance to each other. Consequently, the system consistently recognizes the symbols as long as they are displayed clearly, without any background noise, and with sufficient illumination.

When there are few datasets available, especially when there are only 175 photos for each move, both networks are able to recognise more than 96.8% of the gestures. This finding implies that the influence from other elements during feature extraction can be efficiently reduced by employing a TOF camera to collect images based on depth information.

Across several datasets, it can be seen that the enhanced network performs better than Faster R-CNN in terms of gesture categorization accuracy. Specifically, using 175 photos for each move, the upgraded network obtains an 83% recognition accuracy. This accuracy outperforms Faster R-CNN's performance in gesture classification, demonstrating the supremacy of the upgraded network.

## 6 REFERENCES

- [1] Corina, D. (1998). Chapter 21 - The Processing of Sign Language: Evidence from Aphasia, Editor(s): Stemmer, B. & Whitaker, H. A. *Handbook of Neurolinguistics*, Academic Press, 313-329. <https://doi.org/10.1016/B978-012666055-5/50025-3>
- [2] Prikhodko, A., Grif, M. & Bakaev, M. (2020). Sign Language Recognition Based on Notations and Neural Networks. [https://doi.org/10.1007/978-3-030-65218-0\\_34](https://doi.org/10.1007/978-3-030-65218-0_34)
- [3] Ghorpade, S. R. & Waghmare, S. K. (2015). A Communication System for Deaf and Dumb People. *International Journal of Science and Research (IJSR)*, 4(9), 2319-7064.
- [4] Haleem, M. S. (2008). Voice controlled automation system. IEEE International Multitopic Conference, Karachi, Pakistan. 508-512. <https://doi.org/10.1109/INMIC.2008.4777791>
- [5] Kallio, S., Kela, J., Mäntyjärvi, J. & Plomp, J. (2006). Visualization of Hand Gestures for Pervasive Computing Environments. *Proceedings of the working conference on advanced visual interfaces*, ACM, Italy, 480-483. <https://doi.org/10.1145/1133265.1133363>
- [6] Zaman, K. R. & Noor, I. (2012). Hand Gesture Recognition: A Literature Review. *International Journal of Artificial Intelligence & Applications (IJAA)*, 3, 161-174. <https://doi.org/10.5121/ijaa.2012.3412>
- [7] Jeon, M.-J., Yang, S.-E. & Bien, Z. (2009). User adaptive hand gesture recognition using multivariate fuzzy decision tree and fuzzy garbage model. *IEEE International Conference on Fuzzy Systems*, Jeju, Korea (South), 474-479. <https://doi.org/10.1109/FUZZY.2009.5277156>
- [8] Wilson, A. D. & Bobick, A. F. (1999). Parametric hidden Markov models for gesture recognition. *IEEE transactions on pattern analysis and machine intelligence*, 21(9), 884-900. <https://doi.org/10.1109/34.790429>
- [9] Reyes, M., Dominguez, G. & Escalera, S. (2011, November). Featureweighting in dynamic timewarping for gesture recognition in depth data. In *2011 IEEE international conference on computer vision workshops (ICCV Workshops)*, 1182-1188. <https://doi.org/10.1109/ICCVW.2011.6130384>
- [10] Tang, D., Jin Chang, H., Tejani, A. & Kim, T. K. (2014). Latent regression forest: Structured estimation of 3D articulated hand posture. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3786-3793. <https://doi.org/10.1109/CVPR.2014.490>
- [11] Rahman, M. H. & Afrin, J. (2013). Hand gesture recognition using multiclass support vector machine. *International Journal of Computer Applications*, 74(1), 39-43. <https://doi.org/10.5120/12852-9367>
- [12] Chen, X. & Wang, Z. J. (2013). Pattern recognition of number gestures based on a wireless surface EMG system. *Biomedical Signal Processing and Control*, 8(2), 184-192. <https://doi.org/10.1016/j.bspc.2012.08.005>
- [13] Chudasama, D., Patel, T., Joshi, S. & Prajapati, G. I. (2015). Image segmentation using morphological operations. *International Journal of Computer Applications*, 117(18). <https://doi.org/10.5120/20654-3197>
- [14] Madan, A. (2021). Face recognition using Haar cascade classifier. *Int. J. Mod. Trends Sci. Technol*, 7(01), 85-87.
- [15] Sonka, M., Hlavac, V., Boyle, R., Sonka, M., Hlavac, V. & Boyle, R. (1993). Image pre-processing. *Image processing, analysis and machine vision*, 56-111. [https://doi.org/10.1007/978-1-4899-3216-7\\_4](https://doi.org/10.1007/978-1-4899-3216-7_4)
- [16] Soni, N. S., Nagmode, M. S. & Komati, R. D. (2016). Online hand gesture recognition & classification for deaf & dumb. In *IEEE International Conference on Inventive Computation Technologies (ICICT2016)*, 3, 1-4. <https://doi.org/10.1109/INVENTIVE.2016.7830112>
- [17] Cabrera, M. E., Bogado, J. M., Fermin, L., Acuna, R. & Raley, D. (2012). Glove-based gesture recognition system. In *Adaptive Mobile Robotics* (pp. 747-753). [https://doi.org/10.1142/9789814415958\\_0095](https://doi.org/10.1142/9789814415958_0095)
- [18] Garg, P., Aggarwal, N. & Sofat, S. (2009). Vision based hand gesture recognition. *International Journal of Computer and Information Engineering*, 3(1), 186-191.
- [19] Bradski, G. & Kaehler, A. (2008). *Learning OpenCV: Computer vision with the OpenCV library*. O'Reilly Media, Inc.
- [20] Wang, S. C. & Wang, S. C. (2003). Artificial neural network. *Interdisciplinary computing in java programming*, 81-100. [https://doi.org/10.1007/978-1-4615-0377-4\\_5](https://doi.org/10.1007/978-1-4615-0377-4_5)
- [21] Chua, L. O. & Roska, T. (1993). The CNN paradigm. *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, 40(3), 147-156. <https://doi.org/10.1109/81.222795>
- [22] Pang, B., Nijkamp, E. & Wu, Y. N. (2020). Deep learning with tensor flow: A review. *Journal of Educational and Behavioral Statistics*, 45(2), 227-248. <https://doi.org/10.3102/1076998619872761>

**Authors' contacts:**

**G. K. Dayananda**

(Corresponding author)

Faculty of Electronics & Communication Engineering,  
Canara Engineering College,  
Benjanapadavu, Bantwal Taluk, Mangalore, D. K. District,  
574219 Karnataka, India  
dayanand.gk@gmail.com

**I. R. Manjunath**

Faculty of Electronics & Communication Engineering,  
Canara Engineering College,  
Benjanapadavu, Bantwal Taluk, Mangalore, D. K. District,  
574219 Karnataka, India

**J. G. Sreerama Samartha**

Faculty of Electronics & Communication Engineering,  
Canara Engineering College,  
Benjanapadavu, Bantwal Taluk, Mangalore, D. K. District,  
574219 Karnataka, India

**M. Vayusutha**

Faculty of Electronics & Communication Engineering,  
Canara Engineering College,  
Benjanapadavu, Bantwal Taluk, Mangalore, D. K. District,  
574219 Karnataka, India

**D. N. Pratham**

Electronics & Communication Engineering,  
Canara Engineering College,  
Benjanapadavu, Bantwal Taluk, Mangalore, D. K. District,  
574219 Karnataka, India

**Sakshath**

Electronics & Communication Engineering,  
Canara Engineering College,  
Benjanapadavu, Bantwal Taluk, Mangalore, D. K. District,  
574219 Karnataka, India

**A. U. Shashank**

Electronics & Communication Engineering,  
Canara Engineering College,  
Benjanapadavu, Bantwal Taluk, Mangalore, D. K. District,  
574219 Karnataka, India

**Swasthi Prasad Shetty**

Electronics & Communication Engineering,  
Canara Engineering College,  
Benjanapadavu, Bantwal Taluk, Mangalore, D. K. District,  
574219 Karnataka, India