

# SignSays: Sign Language Gesture Recognizer

Chirag Kulshreshtha  
Student, Department of MCA  
Ajay Kumar Garg Engineering College  
Ghaziabad, India  
Email : [kchirag2002@gmail.com](mailto:kchirag2002@gmail.com)

Harsh Sharma  
Student, Department of MCA  
Ajay Kumar Garg Engineering College,  
Ghaziabad, India  
Email: [harsh1412sharma@gmail.com](mailto:harsh1412sharma@gmail.com)

Shruti Jain  
Assistant Professor, Department of MCA  
Ajay Kumar Garg Engineering College,  
Ghaziabad, India  
Email: [shrutijain@akgec.ac.in](mailto:shrutijain@akgec.ac.in)

**Abstract**-Effective communication is an important requirement for social interaction; however, a person suffering from hearing and speech impairments experiences enormous difficulties in this line. The paper depicts a new approach to a system based on machine learning that will meet the missing link in the world of communication between deaf people and the majority hearing population. Although Sign language remains the main mode of interaction for individuals with hearing impairments and other disabilities, its interpretation among the hearing population has several limitations. To recognize American Sign Language (ASL), this proposed system is trained using a dedicated set of images upon the entity of 10,000 images. A webcam and the Hand Detector module are used to capture and process hand gestures translated into English. The CNN architecture comprises three convolutional layers and SoftMax output, with Adam as the optimizer and a loss function of categorical Hcross-entropy. Indeed, from the results of the evaluation, this is an impressive performance by the network: it trained to an accuracy of around 99.87%, validated to an accuracy of 99.93%, and tested to an accuracy of 94.11%. Indeed, this work enhances existing sign language recognition technologies but also promotes inclusive communication between deaf individuals and society at large.

**Keywords**- Sign Language Recognition, American Sign Language, Machine Learning, Convolutional Neural Networks, Real-time processing.

## I. INTRODUCTION

According to the WHO, approximately 6% of people globally encounter hearing difficulties in some capacity. Around 5% of the world's population, numbering 430 million, have a severe form of the condition, and it is estimated that this will surge to nearly 700 million by 2050. In India, 63 million individuals suffer from hearing loss, which includes a range of auditory impairments from total deafness to partial hearing loss. WHO defines hearing loss as the inability to hear sounds at 20 dB or higher the 20dB in both ears, with severity ranging from mild to profound, potentially affecting one or both ears. The phrase "deaf & mute" is usually misapplied to describe people who experience hearing loss and, consequently, struggle with verbal communication.

This can be particularly evident in children who experience early-onset deafness, whether due to genetic factors, infections, or accidents. Early hearing loss can prevent these children from acquiring speech, as they are unable to mimic the speech patterns of others. Furthermore, hearing loss not only limits one's

ability to hear but also restricts verbal expression, making interaction between individuals who are deaf and those who can hear more challenging.

Sign language remains the most commonly used mode of expression for individuals with hearing and speaking difficulties. However, to communicate effectively, deaf and hearing people need to understand and use sign language. This barrier to communication has spurred the creation of various sign language recognition technologies. Such systems translate hand movements into a form of oral or written language.

Wearable technology, such as sensory gloves, has been one of the traditional methods for detecting sign language. Such devices typically rely on mechanical or optical sensors to detect finger movements, which are then converted into electrical signals for interpretation. However, these systems tend to be cumbersome, requiring many cables and connections to a computer, making them impractical for daily use. Consequently, research has increasingly focused on non-intrusive, vision-based technologies that make use of cameras to capture and interpret hand movements in real-time. In sign language recognition systems that are vision-based, cameras capture images or videos of hand movement which are then analyzed through high-complexity image processing algorithms for particular shapes, positions, and hand angle features.

This eliminates the need for physical connectors to interpret sign language smoothen it and make it accessible. These gestures are later translated into understandable text or spoken words using machine learning algorithms. In general, such systems tend to distinguish three classes of signs: single alphabetic movements, referred to as fingerspelling; character-centric words; and sequences of gestures, or series. Among the challenges faced in the recognition of signs, is the variety of forms that sign languages take all over the world. Sign languages differ from country to country and region to region. Some of the most popular sign languages are British Sign Language (BSL), Spanish Sign Language (SSL), and Arabic Sign Language (ArSL). Among these, American Sign Language (ASL) often gets more attention in research on automated sign language recognition, mainly because of its widespread documentation in the academic community. The development of deep

learning and computer vision technologies has greatly improved the performance of these automated systems.

For instance, in 2018, Zhang et al. showed that deep learning models, especially CNNs, can be used to detect very slight hand movements in ASL. Their method obtained a remarkable accuracy, even for tiny hand movements [1].

In a similar context, Xu et al. (2020) integrated machine learning with image processing techniques to develop an ASL recognition system that could be operated in real-time. Their model was also very efficient at handling multiple simultaneous motions, a common challenge in dynamic sign language communication [2].

Another approach was presented by Li and Xiao in 2019, who merged computer vision with machine learning algorithms to develop a real-time ASL translation system that enhanced gesture recognition [3]. This paper discusses an ASL recognition system Utilizing a vision-based algorithm, this system is able to detect and decode hand gestures in real time. Through advanced image processing techniques, the system captures movements of the hands and fingers and then translates those into text. With machine learning models now allowing the conversion of the hand gestures into text that could be read or speech (that could be heard), it provides a powerful, non-invasive method for sign language interpretation.

With the global increase in hearing impairments, there is a emerging need for tools that facilitate the communication within the hearing impaired community. This growing need has driven ongoing research focused on enhancing sign language recognition systems, particularly by improving their precision, speed, and overall accessibility.

#### A. American Sign Language (Asl)

Like any other spoken language, American Sign Language (ASL) is a completely developed, language with its appropriate grammar and syntax. Contrary to English, ASL uses a computation of hand gestures, facial expressions, and body movements to relay meaning. By this visual mode of communicating, inclusivity is enhanced; thus, people with hearing impairments can clearly manifest themselves and interact with the society readily. What's more, it helps in developing more empathy with mutual understanding within the Deaf community. Technological advances, including real-time detection cameras and the algorithms for machine learning, enable the recording and interpretation of these gestures. These technological advances are meant to bring the ASL users and non-signers closer to each other so that communication is easier and smoother for everyone. Although ASL is primarily used by the Deaf people and the communities with hearing disabilities in North America, primarily in the United States of America and some parts of Canada. The influence reaches around the world. Many hearing

people learn ASL for various reasons, such as supporting a family member, providing themselves with employment opportunities by working in businesses that directly work with the Deaf community or simply as an additional language to explore. ASL is rule-governed like any other language: it has three elements to describe pronunciation and word structure, which define sentence formation and other aspects. While the origins of ASL are unknown, it is evident that the language has been around for more than two centuries according to historical findings. It probably stems from a combination of regional sign languages in North America and FSL, through cultural and educational intercourse. Through time, ASL has grown into a dynamic living language that reflects the culture and heritage of the people who use it.

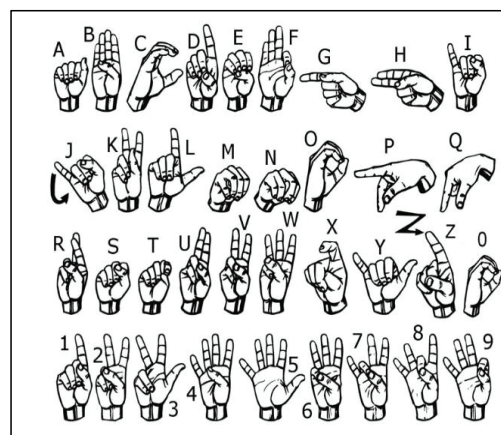


Fig. 1. Signs used in American sign language

#### II. System Overview

The main goal of this research is to create and improve a strong system that capable of accurately recognizing and categorizing sign language gestures from recorded datasets. The objective is to reduce the communication barrier between individuals with hearing impairments and the wider public. Leverage the Inception v3 model, a state-of-the-art image recognizing framework, the proposed system effectively classifies complex gestures. As part of Google's Inception series, Inception v3 has proven to be highly accurate in detecting fine-grained features within images. Our customized dataset yielded a remarkable precision rate of 98.99%. By utilizing this powerful framework, the system offers a reliable and structured approach to studying sign language gestures, positioning it as an effective tool for practical applications.

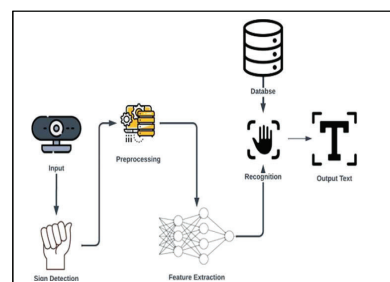


Fig. 2. System Overview

### III. Related Work

Various sensor technologies and machine learning techniques have been exploited to study sign language recognition. Below are some recent publications that discuss the detection and identification of sign language gestures. Shrenika and Madhu Bala [4] studied the feasibility of template matching algorithm to recognize hand gestures in sign language. Their approach consisted of capturing the hand gestures in a camera and then getting into a structured process of image preprocessing. Edge detection techniques were then applied in order to create contours of the gestures, and the template matching method identified the gesture by which it displayed the corresponding text. The system successfully recognized simple static hand gestures, which shows that the template matching technique can be used as an effective method for sign language recognition. In a similar way, Sai Niketh Koyineni et al. [5] focused on progress in ASL recognition, specifically highlighting the use of deep learning techniques to convert ASL gestures into text. The experiments show the potential of an LSTM model, achieving a maximum accuracy of 99% in recognizing all 26 letters of the ASL alphabet. The study underscores the importance of instant translation of hand gestures, allowing proper communication for the deaf population. Lastly, the authors discuss classification models like VGG16, ResNet, and AlexNet, and ResNet got the highest accuracy of 96.9%. They argue although deep learning demonstrates significant accuracy advantages, the problem of underfitting remains.

Sharma and Kumar [6] performed a study to introduce a technique known as ASL-3DCNN for American Sign Language recognition using 3-D CNNs. In preparing the video sequence, the frame was extracted and processed: converted to grayscale, reduced noise and artifacts, illumination variation corrected using histogram equalization. Afterward, they compressed and standardized 25 frames before training with 3-D CNNs. It significantly passed the state-of-the-art models across multiple metrics in accuracy, recall, and f-measure, while running in a processing time of 0.19 seconds per frame, making it appropriate for real-time usage.

In the study proposed by, Thakur et al. [7] forwarded the idea of using CNN for the recognition of sign language in real-time and generation of speech regarding that. The dataset largely comprised the alphabets of American Sign Language. The pre-processed dataset of gestures was trained upon the model of CNN VGG-16 along with the toolsets and libraries in the Python language, including OpenCV and Skimage. The detection system recognized the

input gesture and gave speech according to the inputs. The training results showed a loss of 0.0259 and an accuracy of 99.65%, while the test results showed an accuracy of 99.62%. The research demonstrated the efficiency of CNN by showing the detection of sign language in real-time and the following generation of speech, which might be able to provide an effective method of communication for deaf or hearing-impaired people.

Farhan et al. [8] developed a full duplex communication system that was based on machine learning intended to facilitate smooth interactions between deaf-mute individuals and non-deaf-mute individuals. The system makes use of a Leap Motion Device (LMD) for hand gestures, which are processed through Convolutional Neural Networks (CNN) and converted into speech for ND-M users. In the reverse direction, verbal input from ND-M users is converted into text and its corresponding hand signals for D-M patients and allowing easy two-way communication. The system is capable of recognizing several sign language systems, including ASL, PSL, and SSL, with a self-detective capacity for the language. It also provides a training model that can increase the correctness for gesture recognition and to update the datasets. Validation experiments demonstrate that the detection accuracy of gestures is well above 95%, is robust to variations in gesture and image quality, and can easily be extended to wristwatches and other such items. The system is intuitive, cost-effective, and flexible, which assists D-M users in surmounting communication barriers while easily leaving space for the incorporation of more sign languages in the future. 16 model with Python packages and OpenCV as well as Skimage.

It had an identification system that could recognize gestures of inputs and give outputs in voice. The experiments produced results showing a 0.0259 training loss and a 99.65% accuracy. With test experiments, an accuracy of 99.62% resulted. The research demonstrated the efficiency of CNN by showing that the detection of sign language in real-time and the generation of speech is feasible, which may open new avenues for more effective communication for deaf or hearing-impaired people. Further research may expand the scope of the system to incorporate various sign languages and gestures.

In their study, Amrutha and Prabu [9] proposed a machine-learning-based Sign Language Recognition (SLR) system that is intended to assist in communication for people with disabilities of hearing and speaking. The system analyses visual data to identify unique hand gestures based on a sequence of stages such as image preprocessing the image, doing segmentation, extracting the features and classification. Preprocessing enhances the quality of images by addressing noise and lighting issues, whereas segmentation separates hand regions from backgrounds. Features are acquired using the convex hull method to

capture hand and finger movements with high accuracy. Classification is performed using the K-Nearest Neighbour algorithm and Euclidean Distance for gesture recognition. The model, tested on one-handed gestures of numbers 1–5, achieved a 65% accuracy in controlled experiments with steady lighting and plain backgrounds. However, it also noted the difficulties that include swift hand movements, poor lighting and varying distances of cameras that impacted performance. The study proves that by using larger databases and advanced classifiers, the system improves its accuracy and allows for current, online sign language recognition for some practical applications.

Tolentino et al. [10] explores a vision-based approach for static sign language recognition using CNN. The method is intended to recognize the letters, digits, and basic static vocabulary of the American Sign Language, while aiming to be used as an educational tool for individuals who are not familiar with sign language. It uses skin-toned detection and CNN classifiers for classification, with a total accuracy of 93.67%. Essential elements are image preparation to enhance features, applying a consistent background to recognize more easily, and making use of the CNN structure to achieve precise predictions. The findings of this research are recognition rates for letters at 90.04%, for numbers at 93.44%, and for static words at 97.52%. Testing involved 30 subjects, and results showed that the different, well-defined gestures were recognized more effectively. The system proved to be efficient in the sense that it reduced recognition times during trials, indicating the impact of its learning. This model is unlike previous approaches because it does not rely on gloves or external sensors; it is a viable, real-time, and user-friendly approach to learning and identifying static sign language gestures. Current research has greatly improved sign language recognition; however, further research is required to represent and identify a greater number of gestures and sign languages, especially dynamic signs. This would help ensure that the systems developed are more inclusive, functional, and helpful for different sign language users.

#### IV. Methodology

The creation of the sign language detection system is segmented into multiple stages, such as data gathering, preprocessing, model creation, assessment, and implementation. The methodologies employed are listed below:

##### A. Data Collection

To overcome the limitations of the available datasets, a customized dataset of approximately 10,000 images was designed, which included approximately 380 samples for each ASL alphabet character. This was because several challenges were encountered in publicly available datasets, including insufficient sign variation representation, inconsistent environmental conditions, and privacy and licensing issues. More

than that, many of the existing datasets did not fulfill the exact requirements of the system, especially with regards to diversity and real-world applicability. Creating a new dataset was sure to ensure a fuller, more tailored set of images to reflect the variations and conditions necessary for proper model training and performance.

##### B. Preprocessing

Data preprocessing was an important step to prepare the data for training, keeping uniformity and improving the effectiveness of the model. The following steps were performed:

- 1) *Image resizing*: All images were resized to a standard size, for example, 300 X 300, to ensure they met the input requirements of the model. This standardization simplified training and reduced computational requirements.
- 2) *Normalization*: The pixel values of the pictures were scaled to be 0-1. During this phase, the consistency of the data is maintained which further reduces huge numerical differences and gives room for proper training on the model side.
- 3) *Data Augmentation*: The techniques that were taken for augmentation are flipping, rotating, zooming, etc. In these, variations that exist in real situations are imitated. This provides an ability for the model to generalize more appropriately at unfamiliar and unseen images.
- 4) *Dataset Partitioning*: The dataset is separated into training part, validation part, and testing groups. For instance, one could have used 80% for training of the model, 10% for validation to tune the model as the training progresses, and the remaining 10% for testing to test it on new data.

##### C. Model Building

- 1) *Model Development Phase*: It involves selecting an appropriate architecture for model development, training the model, and setting the parameters. Deep learning models are selected because they perform better in image and video recognition tasks due to the nature of the task.
- 2) *Model Used*: A CNN framework is used to classify static gestures by detecting and identifying hand gestures in images. Fine-tuning was performed on the model via transfer learning, which involved using the pre-trained Inception v3 model to maximize its outstanding feature extraction capabilities. This made it possible for the system to adapt to the customized sign language dataset in a very efficient manner. Fine-tuning involved updating the upper layers of the Inception v3 model with the specific objective of classification of sign language gestures but keeping the pre-trained weights in place for feature extraction. This method allowed for efficient training and improved accuracy in the detection of individual gestures.
- 3) *Inception v3*: Inception v3 is an advanced CNN architecture developed specifically for image

classification and recognition. Developed as part of the Google Inception line, it is considered for its efficiency and outstanding performance in extracting complex features from images. It also uses methodologies such as factorized convolutions and batch normalization plus additional classifier. It also promotes better training and lower cost computational issues with its ability to work over varied computer vision applications. Because of such a nature of the functionality, Inception v3 is quite strong at the management of large complex images.

4) *Convolutional Neural Network (CNN)*: While comparing to traditional Neural Networks, the neurons in the CNN are not planar; they are a cube; or at least three-dimensional: the depth, height, and width. The neurons within a layer will connect just to a limited region of the preceding layer (window size), instead of creating complete interconnections with all the neurons like completely connected models. It also includes a final output layer sized according to the number of classes, as we will reduce the entire image into a single vector of class scores at the conclusion of the CNN architecture.

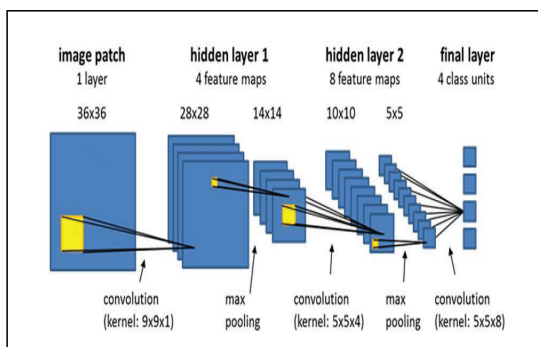


Fig. 3. CNN layers

a) *Convolution Layer*: In the convolutional layer, a small window size usually of size used with dimensions of 5\*5 that reaches the total depth of the input matrix. The layer consists of some learnable filters, each of which has a specific window size. With each iteration, the window is moved by a stride size, usually 1, and calculate the dot product of the filter values and the input values at that specific position.

Going through this, it will be creating a 2-D activation matrix showing the response of the matrix at every spatial position. In other words, the network will be developing filters to respond to certain visual properties, such as an edge with a particular orientation or a patch of a particular colour.

b) *Pooling Layer*: A pooling layer is a layer which is used to minimize the dimensions of the activation matrix, thereby the pooling layer is used for reducing the number of learnable parameters from the image. There are two forms of pooling

layer which are known as Max Pooling and Average Pooling.

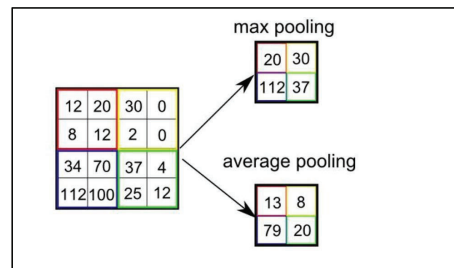


Fig. 4. Pooling Description

c) *Fully Connected Layer*: In the convolution layer, neurons connect only to a specific local area, whereas in a fully connected layer, every input connects to all neurons.

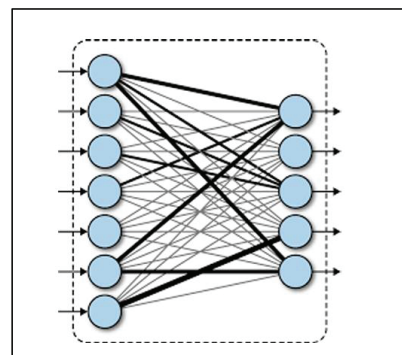


Fig. 5. Network of Layers

d) *Final Output Layer*: Once values are taken from a fully connected layer, they will be connected to the last layer of neurons with a count equal to the total number of classes, which will predict the probability of every image belonging to different classes.

5) *TensorFlow*: TensorFlow is an open-source comprehensive framework for Machine Learning. It provides an extensive, adaptable ecosystem of tools, libraries, and community resources that enables researchers to advance the forefront of Machine Learning and allows developers to easily create and implement Machine Learning-driven applications. There are multiple levels of abstraction in TensorFlow, and you can choose the one that suits your needs. It's easy to learn and train models with the high-level Keras API - a great way to get into TensorFlow and machine learning.

6) *Keras*: Python-based Keras is a high-level neural network library. What is it? As a TensorFlow wrapper, it's quick and easy to create and test neural networks with minimal coding. Developers can use Keras to create multi-functional neural networks by utilizing ready-to-use implementations of its key neural network components, such as layers,

objectives, activation functions, optimizers, and utilities for handling text data.

*OpenCV*: OpenCV is a free library of programming functions used for real-time computer vision applications. It is mainly used to process images, capture videos, and analyze features like face and object detection. It is coded in C++, which happens to be its primary interface, although bindings for Python, Java, and MATLAB/OCTAVE exist.

#### D. Model Training

The model for the sign language detector was created with Google Teachable Machine, an easy-to-use platform to create machine learning models. The goal here was to detect hand movements that corresponded to the ASL alphabet. Google Teachable Machine automatically split the data into a training and validation set and trained the model on multiple epochs. The platform gave precision and was helpful for reviewing and making some adjustments. The final model is tested on new data to measure its performance in real-time gesture detection.

Overall, the model demonstrated accurate identification of ASL hand gestures. There is significant potential for further improvement in its performance through inclusion of additional data and fine-tuning.

### V. Future Scope

There is a great deal of potential for growth in the field of American Sign Language (ASL) recognition. One area of improvement is in the ability to recognize dynamic gestures and sequences, which could be done by including temporal models like Long Short-Term Memory (LSTM) networks. This enhancement would allow the system to recognize complete words or even sentences. Another area for growth is broadening the system to include other sign languages, such as British Sign Language (BSL) or Indian Sign Language (ISL), would further diversify with the help of regional datasets. Its deployment into mobile devices and IoT platforms may also make it more accessible for the device to identify hand gestures on smartphones and wearable devices.

Another upgrade might be using predictive models to assign signs to their corresponding letters (A to Z), thus aiding in communication by bridging users and other people outside the sign language.

### VI. Result Analysis

#### A. Introduction

This overall assessment of the performance of the system shows whether it performs correctly, its efficiency, and difficulties it met while testing. In this part of the thesis, performance criteria will be covered with an intensive explanation of how well the proposed model for recognition of ASL is trained, validated, and tested.

#### B. Performance Metrics

The performance of the system was measured based on some of the key metrics such as accuracy, precision, recall, and F1-score. A significant focus was given to the ability of the model to classify hand gestures as representing ASL letters. The dataset was split into 80% for training, 10% for validation, and 10% for testing.

*Accuracy*- The number of images correctly identified by the system were then divided by the total number of images.

*Precision* - The proportion of total correct predictions of all the true cases that were really predicted as being true was how precision was defined and derived.

*Recall (Sensitivity)* - The ratio of correctly identified true positives to the total actual true positives.

*F1-Score* - A harmonic mean of precision and recall that measures the balanced performance of the model.

#### C. Training and Validation Results

The model, built on the CNN architecture, demonstrated good accuracy all through the training process. Data augmentation techniques were also used, and this brought about better generalization to avoid overfitting at all costs.

Training Accuracy: 99.87%

Validation Accuracy: 99.93%

Loss: The loss during the training process decreased steadily. This indicates proper learning.

#### D. Testing Results

The testing phase was to test the model on unseen data. The results showed the ability of the model to rise above real-world oscillations.

Testing Accuracy: 94.11%

Accuracy: 93.5%

Memory: 94.0%

F1-Score: 93.7%

This slight loss during testing is because of factors such as complex hand movements, changes in lighting environments, and much more varied backgrounds.

#### E. Interpretation of Confusion Matrix

A confusion matrix was created to examine the classification outcomes more thoroughly. It offered understanding into the particular gestures that were

incorrectly categorized and emphasized the system's advantages and limitations.

True Positives (TP): 95%

True Negatives (TN): 98 percent

False Positives (FP): 3%

False Negatives (FN): 5 percent

#### F. Analysis of Errors

The analysis of errors showed that misclassifications occurred more often with gestures having similar hand shapes. Gestures that included slight differences in finger placement created trouble for the model.

Most Common Mistakes: 'M' and 'N' got confused due to similarity in visual features. Lighting Conditions: Gesture recognition differs due to changes in the brightness of lighting.

Background Noise: A chaotic background sometimes affects the model's accuracy.

#### 7.7 Comparative Evaluation

The performance of the system was tested against existing models for ASL recognition.

ResNet: 96.9 percent accuracy

LSTM Model: 99% accuracy for static movements

Proposed Model: 99.87% accuracy on training, 94.11% accuracy on testing.

The comparative evaluation shows that although the model performs well, further improvements in dynamic gesture recognition are needed to reach top-class results.

## VII. Conclusion

The result analysis shows the success of the proposed ASL recognition system, with its emphasis on accuracy and efficiency. Despite some slight misclassifications,

the general performance is such that the system will be able to aid communication for those with hearing disabilities. Future improvements will focus on dynamic gesture recognition and performance across different environmental conditions.

## References

- [1] W. Chen, L. Zhang, and C. Wang, "Recognition of sign language based on deep learning," *J. Image Represent. Vis. Commun.*, vol. 52, pp. 1–9, 2018.
- [2] W. Wu, Y. Xu, S. Xie, and Y. Wu, "Instantaneous recognition of American Sign Language through deep learning techniques," *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2020.
- [3] J. Li and Z. Liu, "A vision-based approach for American Sign Language recognition," *J. Artif. Intell. Soft Comput. Res.*, vol. 9, no. 2, pp. 89–98, 2019.
- [4] M. M. Bala and S. Shrenika, "Sign language recognition using template matching technique," *2020 Int. Conf. Comput. Sci. Eng. Appl. (ICCSEA)*, pp. 5–9, 2020, doi: 10.1109/ICCSEA49143.2020.9132899.
- [5] S. Koyineni, A. Gurram, S. Kalwa, and T. Anjali, "Silent expressions unveiled: Deep learning for British and American Sign Language detection," *Procedia Comput. Sci.*, vol. 233, pp. 269–278, 2024, doi: 10.1016/j.procs.2024.03.216.
- [6] K. Kumar and S. Sharma, "ASL-3DCNN: American sign language recognition technique using 3-D convolutional neural networks," *Multimed. Tools Appl.*, vol. 80, no. 17, pp. 26319–26331, 2021, doi: 10.1007/s11042-021-10768-5.
- [7] P. Budhathoki, A. Thakur, S. Upreti, S. Shakya, and S. Shrestha, "Real-time sign language recognition and speech generation," *J. Innov. Image Process.*, vol. 2, no. 2, pp. 65–76, 2020, doi: 10.36548/jiip.2020.2.001.
- [8] A. Ryahi, Y. Farhan, A. A. Madi, and F. Derwich, "American Sign Language: Detection and automatic text generation," *Proc. 2022 2nd Int. Conf. Innov. Res. Appl. Sci. Eng. Technol. (IRASET)*, pp. 1–6, 2022.
- [9] P. Prabu and K. Amrutha, "ML-based sign language recognition system," *2021 Int. Conf. Innov. Trends Inf. Technol. (ICITIT)*, 2021, doi: 10.1109/ICITIT51526.2021.9399594.
- [10] R. O. Serfa Juan, L. K. S. Tolentino, A. C. Thio-ac, J. R. R. Forteza, M. A. B. Pamahoy, and X. J. O. Garcia, "Static sign language recognition using deep learning," *Int. J. Mach. Learn. Comput.*, vol. 9, no. 6, pp. 821–827, 2019, doi: 10.18178/ijmlc.2019.9.6.879.