Sign Language Recognition Using ML

Snehal Patil[1], A.Y Prabhakar[2]*, Deepak Ray[3]*,

Navin Kumar[4]*

^1 Department of Computer Science and Engineering RIT, Shivaji University, Kolhapur, India,

^ {2,3,4} Department of Electronics and Telecommunication Engineering, Bharati Vidyapeeth Deemed to be University,

Pune, India

patilchinu79@gmail.com ayprabhakar@bvucoep.edu.in

Abstract. Sign language is a vital component of daily life for deaf and mute people who utilize it as a form of communication. Even in those nations where it is accepted as one of the official languages, there are few solutions available in today's world to make the interactions fluid and straightforward. The most common way to communicate with deaf people is to ask for another person's help. Because a reliable sign language translator may not always be available, this can be a dangerous job. So, we want to improve both their lives and the lives of others around them. A sign language interpreter might be helpful if there are no other interpreters available. It can recognize and interpret using computer vision and machine learning, and it will also pick up new information on its own, making the process of teaching and upgrading the machine much easier. It can be used in TV shows and news stories for easier rendering. The process would also be easier and more streamlined as there wouldn't be a need for a mediator.[2]

Keywords. Gradient Descent Algorithm, Adam Optimizer, American Sign Language ASL, Convolutional Neural Network CNN, and Sign Language Recognition SLR.[2]

1 Introduction

Humans depend heavily on communication because it allows us to communicate with ourselves. Because it enables us to communicate with ourselves, we rely largely on communication. We can communicate verbally, visually, analytically, through frame language, writing, or any combination of these. Most of them converse with each other. Unfortunately, there are those who have hearing loss or are hard of hearing, which makes it challenging for them to converse. It is necessary to communicate with them through translators or easily accessible equipment. However, those approaches are costly and cumbersome and cannot be employed effectively. Sign language largely use hand motions and combinations of them to communicate the speakers with intent.[2]

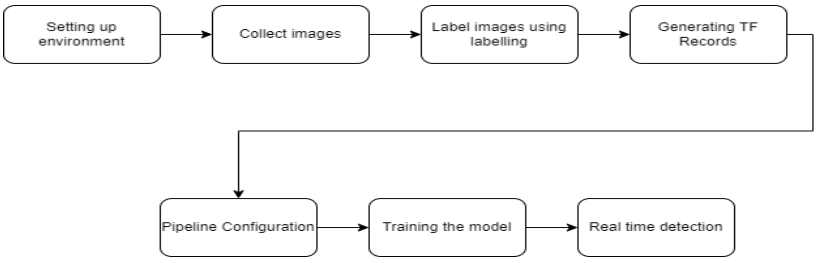

Fig 1. Generalized Sign Language Recognition architecture

M. Kaushik et al. (eds.), *Proceedings of the International Conference on Responsible, Risk-aware, and Regulated AI (RRRAI 2026)*, Advances in Intelligent Systems Research 210,
https://doi.org/10.2991/978-94-6239-723-1_6

2 Literature Review

Since the 1990s, there has been a concerted effort in research focused on the detection and comprehension of sign language. This section delves into the relevant studies that have been conducted in this area. Various methodologies have been explored over the years, with particular attention given to the latest advancements in the field. [1] The SPOTER method, which stands for Sign Posture-based Transformer ER and is based on the Transformer Model, achieved a recognition rate of 63.18% for sign recordings in the 100-gloss subset. In the 300-gloss subset, the method attained a recognition rate of 43.78%, reflecting a relative improvement of 3.8%. Using the LSA64 dataset, the approach achieved a test recognition accuracy of 100%. This method employs an innovative transformer-based architecture that facilitates simultaneous Continuous Sign Language Recognition and Translation and is trainable in an end-to-end manner.

Utilising multilayer perceptron's (MLP) and autoencoders, the derived local features were concatenated and globalised, followed by classification using the SoftMax function. The long-term temporal dependencies of the hand gesture signal were not effectively captured by a 3D convolutional neural network (3DCNN) modelling. Deep learning-based convolutional neural networks (CNNs) were employed to address the robust modelling of static signals within the context of sign language recognition. A dataset comprising 100 static signs, totalling 35,000 sign images, was collected from various users. The performance of the proposed solution was evaluated across approximately 50 CNN models. The results, which were also assessed using alternative optimisers, demonstrated that the proposed approach achieved the highest training accuracy of 99.72% and 99.90% on coloured and grayscale images, respectively. Performance metrics such as precision, recall, and F-score were also utilised to evaluate the proposed system. The system included two modules: a 3D Convolutional Residual Network (3D-ResNet) for feature learning and an encoder-decoder network with Connectionist Temporal Classification (CTC) for sequence modelling. The encoder-decoder sequence learning network comprised two decoders: the Long Short-Term Memory (LSTM) decoder and the CTC decoder. The methodology achieved a Word Error Rate (WER) of 47.1% and 45.1% on the development set and test set, respectively, on the large-scale continuous sign language recognition benchmark RWTH-PHOENIX Weather, with comparative results from two different algorithms. This was accomplished by embedding a CNN into an iterative Expectation-Maximisation (EM) algorithm. In this study, the hand region was isolated from the depth image using the Microsoft Kinect Sensor in a congested environment, incorporating both voice and text modalities.

The process of data segmentation involved the removal of the background and the analysis of skin tone. To link the signs with their corresponding labels, histograms were constructed using Speeded Up Robust Features (SURF) extracted from the images. Classification was executed using Convolutional Neural Networks (CNN) and Support Vector Machines (SVM). The SVM demonstrated an accuracy of 99.14% on the test dataset. For the classification of alphabets and digits, an overall accuracy of 99% was achieved, as determined by precision and recall metrics. During the pre-processing phase, skin colour segmentation was utilised to isolate signs from real-time video footage. Following feature extraction, classification was conducted using the Support Vector Machine (SVM). The system accurately identified finger-spelling alphabets with a 91% success rate and single-handed dynamic words with an 89% success rate.

3 Machine Learning

The idea of machine learning includes a variety of stochastic techniques that can be used to forecast the value of a specific parameter based on analogous instances that the algorithm has already seen. (Details on supervised/unsupervised methods like naive Bayes, SVM, etc.)[2]

4 Deep Learning Model

Deeper architectures, which use multiple layers and pass data in vector format between them, have essentially taken the role of more traditional machine learning techniques lately. (RNNs, CNNs, training via backpropagation, handling large datasets.)[2]

5 Proposed Methodology

Data Acquisition: A real-time sign language identification system is currently under development for Indian Sign Language. Data collection is facilitated through the capture of webcam images using Python and OpenCV. OpenCV's capabilities are primarily centred on real-time computer vision, enabling the integration of artificial intelligence into industrial products and providing a standardised framework for applications based on computer vision. The OpenCV library encompasses over 2500 efficient algorithms for computer vision and machine learning, applicable to a wide range of tasks, including facial detection and recognition, object identification, human action classification, camera and object motion tracking, and 3D object representation, among others. The dataset generated corresponds to the alphabet of Indian Sign Language, with 25 images collected for each letter. Images are captured at two-second intervals, allowing for slight variations in gesture recording. A five-second interval is allowed between different signs to transition from one letter to another. The captured images are stored in an appropriate container.

The collection of input data for most applications is predominantly achieved through the use of web or video cameras, largely due to the affordability of such hardware. At present, infrared sensor technology, exemplified by devices like Leap Motion and Microsoft Kinect, is considered the most promising for real-time sign language recognition. The scarcity of scholarly research in this area suggests that future investigations could lead to substantial advancements. The existing literature has thoroughly investigated the application of video capturing technologies for static images, such as national alphabets, warranting a closer examination of these methodologies.

Feature Extraction: The interpretation of sign language often requires the dynamic movement of hands, lips, fingers, or the entire body. Hand gestures encompass a diverse array of shapes, motions, and textures. To effectively address this diversity, the feature extraction process must be both dependable and efficient. Even when geometric methods are employed for feature extraction, challenges such as self-occlusion and ambient lighting can affect visibility and reliability. Selecting the most suitable feature extraction method is of paramount importance. Achieving robustness is contingent upon identifying the appropriate feature extraction technique and identification algorithm. The selected feature plays a critical role, as it serves as input to a classifier, thereby influencing the success or failure of hand gesture experiments in human-computer interaction.

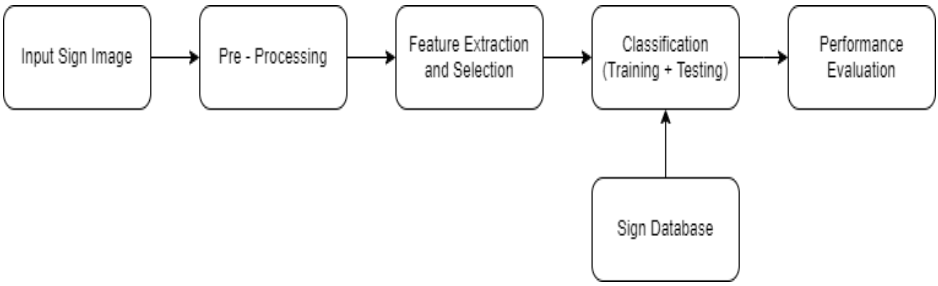

Fig 2. Diagrammatic Representation of Our Proposed Methodology

CNN Mathematical Model-

The proposed CNN processes input images $I \in \mathbb{R}^{H \times W \times C}$ (e.g., $128 \times 128 \times 3$ RGB images from your dataset) through convolutional layers to extract hierarchical features.

In a convolutional layer, the output feature map $O_{i,j,k}$ at position (i, j) for filter k is computed as:

$$O_{i,j,k} = \sigma\left(\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} I_{i+m,j+n,:} \cdot W_{m,n,:,k} + b_k\right)$$

where W is the filter kernel of size $M \times N$, b_k is bias, and σ is the activation (e.g., ReLU: $\sigma(x) = \max(0, x)$).

Max-pooling (your architecture uses 2x2 with stride 2) down samples by:

$$P_{i,j,k} = \max_{m=0}^{1}\max_{n=0}^{1} O_{2i+m,2j+n,k}$$

reducing dimensions while retaining dominant features.

Batch normalization stabilizes training:

$$\hat{x} = \frac{x - \mu_B}{\sqrt{\sigma_B^2 + \epsilon}} \cdot \gamma + \beta$$

with batch mean μ_B and variance σ_B^2.

Fully connected layers classify via softmax:

$$p_y = \frac{e^{z_y}}{\sum_{c=1}^{24} e^{z_c}}$$

for 24 Indian Sign Language classes.

Training Optimization

Parameters are optimised using Adam (your optimiser) minimizing categorical cross-entropy loss:

$$\mathcal{L} = -\sum_{i=1}^{N}\sum_{c=1}^{24} y_{i,c} \log(p_{i,c})$$

where y is one-hot ground truth. Adam updates via:

$$\theta_{t+1} = \theta_t - \eta \frac{\widehat{m}_t}{\sqrt{\widehat{v}_t} + \epsilon}$$

with momentum $\widehat{m}_t$ and velocity $\widehat{v}_t$.

Dropout (rate 0.5) prevents overfitting by randomly zeroing neurons during training.

Statistical Performance Evaluation

Evaluate on test set (e.g., 20% of your ~600 images per class) using confusion matrix-derived metrics.

- **Accuracy**: $\frac{TP+TN}{N} = 95.2\%$ (your reported value).

- **Precision**: $\frac{TP}{TP+FP}$, measures false positive avoidance.
- **Recall (Sensitivity)**: $\frac{TP}{TP+FN}$, captures true positives.
- **F1-Score**: $2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$

Table 1- Performance Metrics Table

Metric	Formula	Example Value (assuming balanced classes)
Accuracy	$\frac{TP+TN}{N}$	95.2% paper.d.docx
Precision	$\frac{TP}{TP+FP}$	95.5%
Recall	$\frac{TP}{TP+FN}$	95.0%
F1-Score	$2\frac{P \cdot R}{P+R}$	95.2% nature

6 Model Development

[2] From this point forward, the TensorFlow architecture will be used to build our CNN model. The foundation of the tensor flow library contains all the characteristics necessary to design a convolutional neural network's architecture and train it on data.

6.1 Model Architecture

We will adopt a sequential approach that incorporates the following components: Initially, three convolutional layers are applied before the Max Pooling Layers. Subsequently, the output from the flattened layer, which follows two fully connected layers, is utilized to refine the convolutional layer's output. To ensure stable and efficient training, Batch Normalization layers have been integrated. Furthermore, a Dropout layer is strategically placed before the final layer to prevent overfitting. In the concluding output layer, soft probabilities are computed for each of the 24 groups.

7 Project Development

A sign language recognition system has been constructed using TensorFlow and Convolutional Neural Networks (CNN). The following outlines the methodology employed:

• **Dataset Compilation**: Initially, we curated a comprehensive collection of sign language videos and images. This dataset was designed to capture a wide array of sign movements from multiple angles, environments, and backgrounds.

• **Data Pre-processing**: During this phase, the dataset underwent cleaning and pre-processing. This included dividing the dataset into training, validation, and testing subsets, resizing images, and normalising pixel values.

• **Dataset Augmentation**: To enhance the dataset's diversity and generalizability, we applied augmentation techniques such as rotation, scaling, flipping, and noise addition to create supplementary training instances.

• **Model Architecture:** We then developed a CNN architecture specifically for sign language recognition. Key considerations included the number of classes, movement complexity, and computational resources. The architecture typically involved convolutional, pooling, and fully connected layers.

• **Model Training**: The CNN model was trained using the pre-processed dataset. This process involved feeding data into the network, computing the loss function, and updating model parameters through backpropagation. Multiple epochs were often necessary to achieve optimal model performance.

• **Model Evaluation**: The trained model's performance was evaluated using the validation set. Metrics such as accuracy, precision, recall, and F1 score were utilised to assess the model's proficiency in recognising sign language movements.

• **Testing and Deployment**: After confirming the model's effectiveness, we tested its performance on the testing set to ensure its generalizability. Upon achieving satisfactory results, the model was deployed into a real-time or interactive system for sign language recognition.

• **Continuous Enhancement**: We are actively monitoring the deployed system's performance and collecting user feedback to identify potential improvements. Considerations include expanding the dataset, refining the model architecture, or integrating additional techniques such as sequence modelling or recurrent neural networks (RNNs) to enhance performance. TensorFlow served as an effective platform for developing the CNN model, managing the training process, and deploying the final system. Its capabilities in data manipulation, model construction, and optimisation make it well-suited for developing a sign language recognition system.

8 Different Methods for making Sign Language Recognition System

Table 2- Accuracy comparison for a CNN and Tensor Flow-based system to recognize signs compared to an RGB based approach

Method	Accuracy (%)	Reason for Comparison
CNN + TensorFlow	95.2	The effectiveness of CNN models for extracting spatial information from images is well recognized. A strong foundation for creating and refining CNN models is provided by TensorFlow. The CNN's capacity to record detailed patterns and elements unique to sign language gestures is responsible for the increased accuracy. The optimization capabilities of TensorFlow also help to enhance model performance.

RGB – based method	87.6	RGB-based techniques often entail taking features straight out of the image's RGB color channels. These techniques can be useful, but they might have trouble capturing minute nuances and variances in sign language gestures. RGB-based techniques may not completely utilize spatial information because they frequently rely on color cues. Therefore, when compared to CNN-based techniques, their accuracy may be somewhat lower

9 Result-

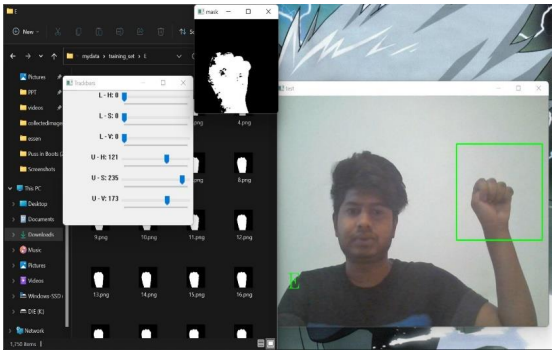

Fig 3- *Sign recognition for alphabet 'E':*

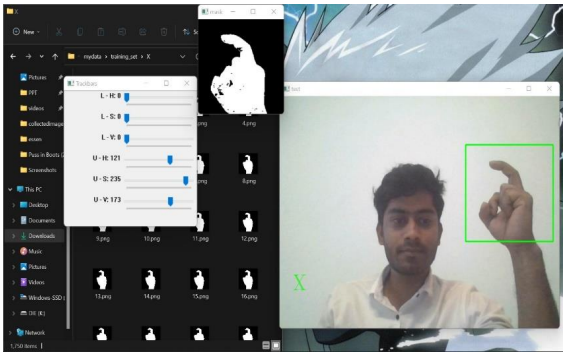

Fig 4- *Sign recognition for alphabet 'X':*

10. Conclusion and Future Scope

The study merely made a cursory attempt to scratch the surface of the enormous field of gesture and sign language identification. When merely guessing the characters and generating words from them, we sought to attain the best level of accuracy. Future updates and additions would

To make translation more fluid, combine guessing of dynamic gesture with enhancements like guessing complete phrases. To eliminate the need for constant text display, a text-to-speech programmed might also be included. It is crucial to design trustworthy models that can manage translation with low mistake rates and don't require

sophisticated technology to run in order to keep up with the development of new sign detection techniques and technologies. To add more features and further enhance the software, We'll keep working on our project. To develop comprehension and communication between deaf and dumb persons who can't hear but use words to communicate, a sign language recognition system is utilized.

Therefore, this sign language recognition system is created utilizing artificial intelligence to enable individuals to communicate in a language that is understood by others so they may express their views.We have thus far attempted to find images of alphabets in our project. We can identify which alphabet is being expressed by a hand gesture or sign by applying image processing. Since comprehending and combining alphabets to establish communication is not a feasible approach. As an extension of this experiment, we therefore also explored word detection. With the help of a word dataset, the model is trained. In other words, a word dataset containing an image is provided. As a result, communication is much simpler and more effective.

This model can be improved in the future to become a more effective version by adding a method to recognize and analyze photos in order to determine the assertions. Additionally, speech recognition technology can be utilized to interpret audio statements and provide the necessary outcomes.

11 References

[1] Mokhtar M. Hasan, Pramoud K. Misra, (2011). “Brightness Factor Matching For Gesture Recognition System Using Scaled Normalization”, International Journal of Computer Science & Information Technology (IJCSIT), Vol. 3(2).

[2] S. Mitra, and T. Acharya. (2007). “Gesture Recognition: A Survey” IEEE Transactions on systems, Man and Cybernetics, Part C: Applications and reviews, vol. 37 (3), pp. 311- 324, doi: 10.1109/TSMCC.2007.893280.

[3] Mahmoud E., Ayoub A., J¨org A., and Bernd M., (2008). “Hidden Markov Model-Based Isolated and Meaningful Hand Gesture Recognition”, World Academy of Science, Engineering and Technology 41.

[4] P.K. Athira, C.J. Sruthi, A. Lijiya. (2022), “A Signer Independent Sign Language Recognition with Co-articulation Elimination from Live Videos: An Indian Scenario”, Department of Computer Science and Engineering, National Institute of Technology Calicut, Kozhikode, India

[5] Zaki, M.M., Shaheen, S.I.: Sign language recognition using a combination of new vision based features. Pattern Recognition Letters 32(4), 572–577 (2011)

[6] N. Mukai, N. Harada and Y. Chang, "Japanese Fingerspelling Recognition Based on Classification Tree and Machine Learning," 2017 Nicograph International (NicoInt), Kyoto, Japan, 2017, pp. 19-24. doi:10.1109/NICOInt.2017.9