Enhanced Indian Sign Language Detection

Aditya Shinde[1], Shivam Dhawale[2], Vedant Lokhande[3], Chinmay Rawool[4], Prof. Pooja Pawale[5]
Department of Computer Science and Engineering, MIT Art, Design and Technology University[1],[2],[3],[4]
Professor at MIT ADT University[5]

Abstract -

The communication problem involving members of society who have speech and hearing impairments is still not fully resolved. In an earlier study, we created a real-time Indian Sign Language (ISL) recognition system which uses LSTM architecture for sequential gesture recognition. The focus of this paper is on further improving this system by changing the architecture from LSTM to CNN to enhance spatial feature extraction and overall system performance.

Using a more comprehensive ISL dataset, we trained and tested the model and added new advanced preprocessing techniques such as Gaussian blur and converting the images to grayscale. These modifications improved the accuracy of the model and reduced the processing power needed, allowing for more advanced, rapid, and reliable real-time ISL gesture recognition. The result of this study is in the direction of making available an effective, simple, and easy-to-use technological interface for the deaf and hearing-impaired people in India.

Keywords: Indian Sign Language (ISL), Gesture Recognition, Convolutional Neural Networks (CNNs), Real-time Communication, Image Preprocessing, Gaussian Blur, Grayscale Conversion, Sign Language Translation, Computer Vision, Human-Computer Interaction (HCI), Deep Learning

1.Introduction

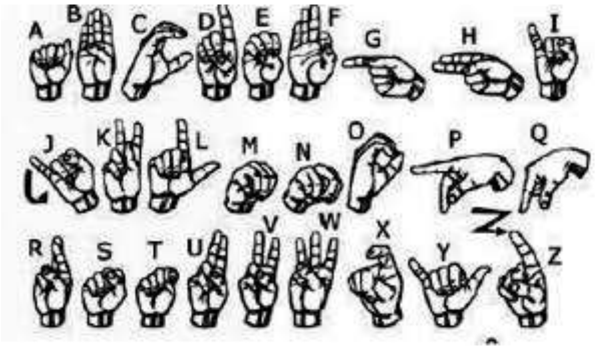


Figure 1: Indian sign language dataset([12]Amrutha, C.U. & Davis, Nithya & Samrutha, K.S. & Shilpa, N.S. & Chunkath, Job. (2016))

Communication is a basic human interaction, yet deaf people with speech and hearing disabilities are usually subjected to great hindrances in communicating with those who are unaware of sign language. Indian Sign Language (ISL), although widely practiced among the deaf in India, is not well understood by the populace at large, resulting in constant communication breakdowns and social isolation. Bridging this divide requires intelligent systems capable of interpreting sign language gestures and converting them into spoken or written language in real time.

In our prior work, we implemented a system based on LSTM for real-time recognition of ISL, allowing for simple translation of sign gestures into speech or text. Although functional to a certain degree, the sequential model suffered from processing speed, spatial feature extraction, and noise robustness in true-world scenarios. This paper proposes an enhanced system that substitutes the LSTM with a Convolutional Neural Network (CNN) framework, optimized for spatial hand gesture recognition from video frames.

Furthermore, we improved the image preprocessing pipeline by using Gaussian blur and grayscale conversion, which greatly enhanced gesture clarity and background noise reduction, resulting in improved recognition accuracy. We also presented a new, more varied ISL dataset for training and testing, including a wider variety of static and dynamic gestures that cover the ISL vocabulary.

To give a visual context, Figure 1 shows a subset of ISL gestures employed in our dataset. This visual reference is a basis for comprehending the kinds of hand signs that the model learns to identify and interpret.

With these enhancements, our system supports more rapid and precise real-time gesture recognition, improving the usability and reliability of sign language translation systems. The long-term aim of this research is to support the development of barrier-free communication technologies that empower the hearing-impaired community and promote greater accessibility in society.

2. Proposed methodology

A. Data Collection:

To gather gesture images of American Sign Language alphabets (A–Z), a Python and OpenCV-based custom data acquisition system was implemented. It uses a webcam to record hand gestures in a static Region of Interest (ROI) per frame. It establishes a hierarchical directory structure (dataset-alpha/train/ and dataset-alpha/test/) with individual alphabet class subdirectories to save the labelled image examples.

On execution, it displays the webcam feed along with a counter indicating the number of images gathered for each class. While the user makes a hand gesture within the ROI and hits the corresponding alphabet key on the keyboard, the processed image gets saved to the corresponding class folder. This interactive, user-driven process ensures proper labelling and permits balanced data gathering for all classes.

B. Preprocessing:

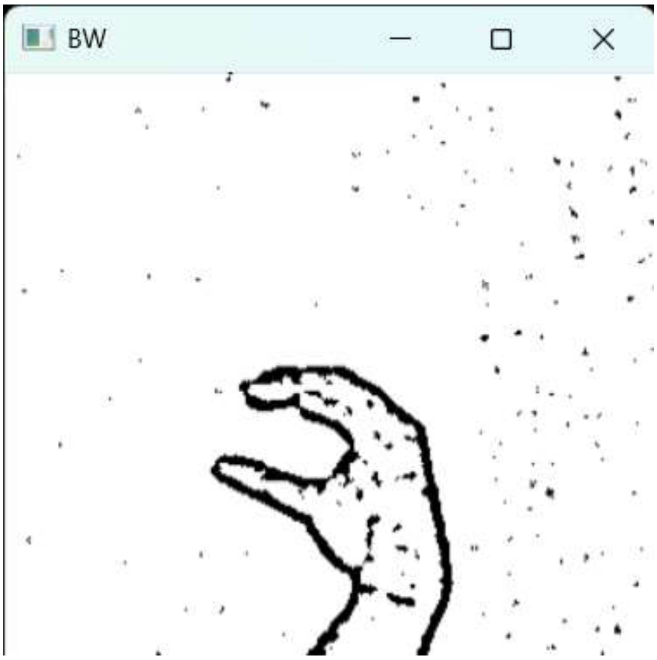


Figure 2: Preprocessing pipeline showing grayscale conversion, Gaussian blurring, and final binarized output of a hand gesture.

Before storage, each captured frame undergoes a series of preprocessing steps to enhance image quality and isolate the hand gesture from the background noise. The preprocessing pipeline includes:

- **Grayscale Conversion**: Converts the ROI to a single channel image to reduce complexity.
- **Gaussian Blur**: Applies a Gaussian filter to reduce noise and smooth the image.
- **Adaptive Thresholding**: Highlights gesture features by creating binary contrasts using local pixel variations.
- **Otsu's Binarization**: Further refines thresholding by determining the optimal global threshold value.
- **Resizing**: The final binarized image is resized to 300×300 pixels to standardize the dataset for model training.

These steps ensure that the dataset contains uniform and high-contrast gesture images, improving the performance of downstream machine learning models.

C. Model Architecture:

The hand gesture recognition system proposed is intended to classify both alphabet gestures (A–Z) and digit gestures (0–9) based on a common Convolutional Neural Network (CNN) model architecture. While the training files and the dataset are distinct for alphabets and digits, the underlying model for both tasks will be the same.

CNN Architecture Overview

The CNN is deployed with TensorFlow and Keras.It accepts grayscale input images of size 128×128 pixels and outputs class probabilities through a softmax layer. The structure consists of:

Input Layer: Accepts grayscale input images of shape (128, 128, 1).

Convolutional Layer 1: 32 filters of size 3×3 with ReLU activation.

MaxPooling Layer 1: 2×2 pooling to downsample feature maps.

Convolutional Layer 2: 64 filters of size 3×3 with ReLU activation.

MaxPooling Layer 2: Another 2×2 pooling to further reduce the dimension.

Flatten Layer: Reshapes the feature maps to a 1D vector.

Fully Connected Layer 1: 128 units, ReLU activation, and 20% dropout.

Fully Connected Layer 2: 112 units with ReLU activation and 10% dropout.

Fully Connected Layer 3: 96 units with ReLU activation and 10% dropout.

Fully Connected Layer 4: 80 units with ReLU activation and 10% dropout.

Fully Connected Layer 5: 64 neurons with ReLU activation.

Output Layer:

26 neurons (for A–Z) with softmax activation or

10 neurons (for 0–9) with softmax activation

The output layer is dynamically set depending on the task:

For alphabet recognition, the model is trained on 26 output classes.

For digit recognition, the same model architecture is trained on 10 output classes.

Training Details

Optimizer: Adam

Loss Function: Categorical Crossentropy

Regularization: Dropout layers are used to avoid overfitting.

Early Stopping: Used to stop training when validation performance stabilizes.

This common architecture provides homogeneity and scalability, which makes it cost-effective to train and deploy models for both types of gestures. The ability to change datasets with the basic architecture intact allows a modular and scalable system

3.Actual Execution and Results

The entire system was implemented in three main stages: preprocessing, training, and real-time inference. This section describes the end-to-end pipeline and implementation plan.

3.1. Data Collection and Preprocessing:

Two distinct datasets were employed for training:

Alphabets (A–Z) dataset

Digits (0–9) dataset

Both datasets processed each image through the following preprocessing operations:

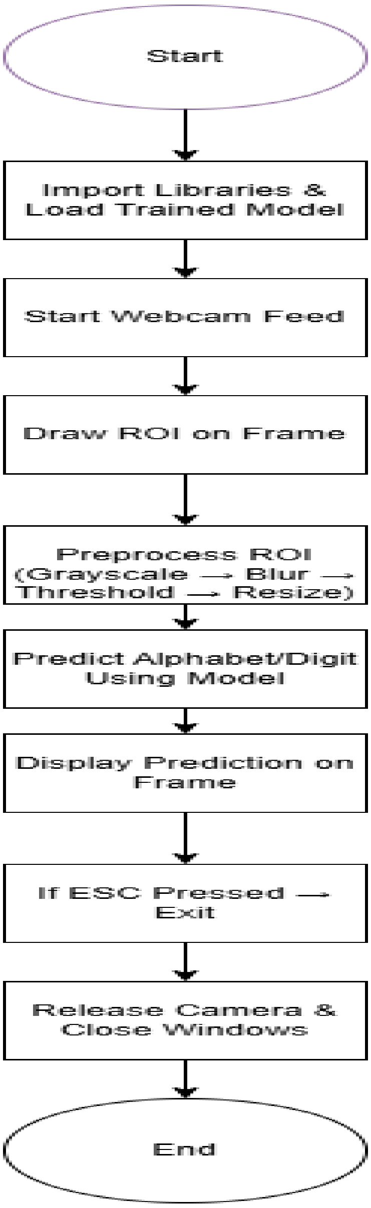
Grayscale Conversion using OpenCV.

Gaussian Blurring to remove noise.

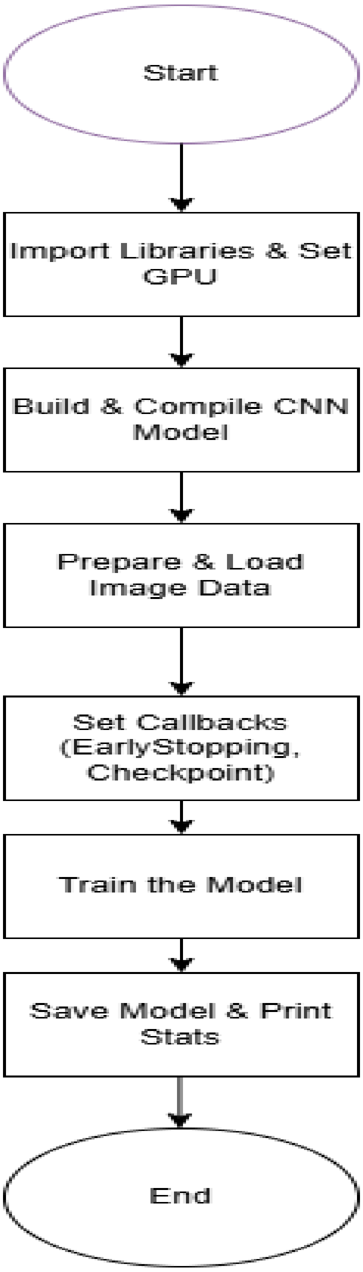
Adaptive Thresholding and Otsu's Binarization to emphasize the hand gesture in high contrast.

Resizing to 128×128 to enable uniform input to the CNN model.

The above preprocessing measures ensured lighting and background noise robustness. This was visually checked and confirmed with live feed samples.


Start
Import Libraries & Load Trained Model
Start Webcam Feed
Draw ROI on Frame
Preprocess ROI (Grayscale → Blur → Threshold → Resize)
Predict Alphabet/Digit Using Model
Display Prediction on Frame
If ESC Pressed → Exit
Release Camera & Close Windows
End

3.2. Training of Model


Start
Import Libraries & Set GPU
Build & Compile CNN Model
Prepare & Load Image Data
Set Callbacks (EarlyStopping, Checkpoint)
Train the Model
Save Model & Print Stats
End

A Convolutional Neural Network (CNN) architecture was employed, trained with grayscale input of the following properties:

Input shape: (128, 128, 1)

Layers: Conv2D → MaxPooling → Conv2D → Flatten → Dense (several layers with Dropout)

Final output layer: Softmax with 36 output units (26 alphabets + 10 digits)

Loss Function: categorical_crossentropy

Optimizer: Adam

Epochs: Set for 30, early stopping enabled at epoch 10

Dataset Split: Custom folder organization for training and testing subsets for both alphabets and digits.

Model achieved:

Training Accuracy: ~98.58%

Validation Accuracy: ~70.00%

3.3. Real-Time Prediction and Execution

The trained model was utilized in a real-time hand gesture recognition system employing a webcam. The process was:

Live Capture of Video using OpenCV.

Region of Interest (ROI) chosen in the frame to recognize hand gestures.

Preprocessing pipeline executed live on ROI:

Convert to grayscale

Apply Gaussian blur

Utilize adaptive thresholding and Otsu's technique

Reshape image to (1, 128, 128, 1) and feed into trained CNN.

Predicted output class decoded with a label dictionary (A-Z, 0-9).

Show prediction on video feed using cv2.putText.

The model correctly estimated hand gestures at runtime at very high confidence on distinctly provided signs. Webcam-inference caused trivial latency and delivered seamless gesture recognition experience.

4. Future Scope

Our enhanced Indian Sign Language (ISL) recognition system, now based on CNNs and more advanced image preprocessing, has good prospects to expand further. The following directions can shape its future development:

A. Cleverer Preprocessing

Adjusting to Real Conditions: Coming systems may learn to adapt automatically to changes in lighting or the background in order to maintain recognition accuracy.

Modern Camera Types: Utilizing depth or infrared cameras can enable enhanced gesture detection in nighttime or congested environments.

B. Mobile and Offline Operation

Run on Phones: The program can be lightened to run on smartphones or tablets without the use of the internet.

Wearables: It may be incorporated in smartwatches, glasses, or gloves so that with the help of appropriate electronic framework for wider usage.

C. Improved Datasets

More Diverse Users: Having individuals from various areas and ages in the training dataset will enable the system to see a greater variety of gestures.

More Information: Future data could have facial expressions and body language for better interpretation, further expansion in database will lead to increased practical efficiency.

D. Two-Way Communication

Speech/Text to ISL: Not only recognizing ISL, but the system might also convert speech or text to sign language.

Understand Emotions: Incorporating emotion detection (such as happy, angry, confused) can make it more natural to communicate.

E. Learning and Accessibility Tools

ISL Learning Apps: The system can drive engaging and interactive tools to educate ISL to both deaf and hearing individuals.

Application in Schools and Public Services: It can be integrated into classrooms, hospitals, or government offices to assist with inclusive communication.

F. Support by Community and Government

Collaboration with NGOs: Collaboration with disability support groups can make the system more beneficial and accessible to more individuals.

Open Source and Sharing: Opening up data and tools for free can enable other researchers to extend this work.

Government Assistance: With funding and policy support, this system can be applied across society more

5.Conclusion

This enhanced Indian Sign Language (ISL) detection system removes the communication gap between the hearing-impaired community and others by using computer vision and deep learning approaches. By switching from an LSTM model to a more efficient CNN design and applying preprocessing techniques like Gaussian blur and conversion to grayscale, the system results in higher accuracy in gesture recognition and quicker response. The addition of live text-to-speech capability further increases user engagement, allowing for more natural and effective communication.

References

[1] Sawant, S. Narayan, and Kumbhar, M. S., "Real-Time Sign Language Recognition using PCA," 2014.

[2] Hussein, B., Shareef, S., "An Empirical Study on the Correlation between Early Stopping Patience and Epochs in Deep Learning," *ITM Web of Conferences*, vol. 64, 2024. DOI:10.1051/itmconf/20246401003.

[3] Alzubaidi, L., Zhang, J., Humaidi, A.J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaría, J., Fadhel, M.A., Al-Amidie, M., Farhan, L., "Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions," 2021.

[4] O'Shea, K., Nash, R., "An Introduction to Convolutional Neural Networks," arXiv preprint arXiv:1511.08458, 2015.

[5] K. Shenoy, T. Dastane, V. Rao, D. Vyavaharkar, "Real-time Indian Sign Language (ISL) Recognition," Department of Computer Engineering, K. J. Somaiya College of Engineering, University of Mumbai.

[6] A. K. Sahoo, G. S. Mishra, and K. K. Ravulakollu, "Sign Language Recognition: State of the Art," Department of Computer Science and Engineering, Sharda University, Greater Noida, India.

[7] Yamashita, R., Nishio, M., Do, R. K. G., and Togashi, K., "Convolutional Neural Networks: An Overview and Application in Radiology," *Insights into Imaging*, vol. 9, pp. 611–629, 2018.

[8] P. K., "Study of Grayscale Image in Image Processing," *International Journal on Recent and Innovation Trends in Computing and Communication*, vol. 4, no. 11, pp. 309–311, 2016.

[9] Brigato, L., and Iocchi, L., "A Close Look at Deep Learning with Small Data," *Sapienza University of Rome*, 2018.