International Journal for Modern Trends in Science and Technology
Volume 12, Issue 04, pages 932-938
ISSN: 2455-3778 online
Available online at: http://www.ijmtst.com/vol12issue04.html
DOI: https://doi.org/10.5281/zenodo.19644344

Deep Learning – Based India Sign Language Gesture Classification using CNN & MediaPipe

Dr. B. V. S. Varma, S. Venu Gopala Swamy, A. Pavan Kumar Varam, T. Sai Dhanusha, B. Sai Vishwanath

Department of Computer Science and Engineering, D.N.R. College of Engineering & Technology, Balusumudi, Bhimavaram, Andhra Pradesh, India

To Cite this Article
Dr. B. V. S. Varma, S. Venu Gopala Swamy, A. Pavan Kumar Varam, T. Sai Dhanusha & B. Sai Vishwanath (2026). Deep Learning – Based India Sign Language Gesture Classification using CNN & MediaPipe. International Journal for Modern Trends in Science and Technology, 12(04), 932-938. https://doi.org/10.5281/zenodo.19644344

Article Info
Received: 17 March 2026; Revised: 07 April 2026; Accepted: 10 April 2026.

KEYWORDS

Indian Sign Language (ISL), Convolutional Neural Networks (CNN), MediaPipe, Hand Gesture Recognition, Deep Learning, Computer Vision, Real-Time Detection, Streamlit Application

ABSTRACT

Indian Sign Language (ISL) plays a vital role in enabling communication for the deaf and hard-of hearing community. However, the lack of automated and accurate sign language interpretation systems remains a significant barrier to inclusive interaction. This project presents a deep learning based approach for Indian Sign Language gesture classification using Convolutional Neural Networks (CNN) integrated with MediaPipe. MediaPipe is employed for real-time hand landmark detection and feature extraction, enabling precise capture of hand gestures while reducing background noise and computational complexity. The extracted landmark features are used to train a CNN model capable of learning spatial patterns associated with various ISL gestures. The proposed system achieves effective gesture recognition with high accuracy and robustness under varying lighting and background conditions. The application is deployed using Streamlit, offering a simple and interactive interface with two modes: image upload and webcam capture. Once the user provides an image, the system processes the hand region and predicts the corresponding ISL gesture along with confidence scores and class wise probability distribution. The modular structure of the code ensures efficient model loading, fast inference, and real-time performance.

1. INTRODUCTION

Gesture recognition has become one of the most promising fields in human–computer interaespecially for communication systems designed for individuals with hearing and speech impairments. Sign languages, such as American Sign Language (ASL) or Indian Sign Language (ISL), rely heavily on hand shapes, finger positions, and movements to convey meaning. Recognizing these gestures automatically through computer vision provides a foundation for building

intelligent assistive technologies. The project illustrated above presents a complete workflow for gesture recognition, beginning from raw gesture input and progressing through advanced image processing, feature extraction, learning, and final classification. The system starts by capturing hand gesture images representing different signs. These images act as the primary input and must clearly show the hand configuration to ensure accurate recognition. Once collected, the images undergo a pre-processing phase where noise is removed, the background is suppressed, and the hand region is segmented. Techniques such as grayscale conversion, thresholding, edge detection, and morphological operations help transform the input into a cleaner and more meaningful representation. Pre-processing enhances the important visual features of the gesture, reduces computational complexity, and prepares the images for the next stage. Finally, the system generates an output, which corresponds to the recognized gesture. This output may represent an alphabet letter, a word, or a command, depending on the application. The recognized gesture can be displayed as text, converted to audio, or used to control other systems. Such recognition platforms play a vital role in assistive technology by enabling communication between hearing-impaired individuals and the general public. **PURPOSE**: The primary purpose of this project is to detect offensive language on social media platforms using text classification techniques. It aims to create a model that can analyze user-generated content and classify it as offensive or not, helping platforms ensure safer online communication. **MOTIVATION**: The motivation behind this work stems from the growing issue of offensive and abusive language on social media, which negatively affects users and communities. With the increasing influence of social media in everyday life, there is a need to automatically detect and prevent toxic behavior to promote healthy digital interactions. Manual moderation is not scalable, which makes automation essential. PROBLEM **STATEMENT**: Communication is a basic need in everyday life, but individuals who use Indian Sign Language (ISL) often face difficulties when interacting with people who do not understand sign language. This communication gap creates challenges in education, workplaces, and social environments. Most people are not trained in ISL, making it hard to understand gestures made by deaf and mute individuals. This results in misunderstandings and limits effective communication. Existing gesture recognition systems have several drawbacks such as low accuracy, high cost, lack of real-time performance, and dependency on special hardware like gloves or sensors. These limitations make them less practical for daily use. There is a strong need for a system that is affordable, accurate, and capable of recognizing gestures in real time using simple devices like cameras. The solution should be user-friendly and accessible to everyone. MediaPipe for detecting and extracting hand landmarks Convolutional Neural Network (CNN) for classifying gestures The system captures hand gestures through a camera, processes the input, and predicts the corresponding ISL gesture. The developed system will help bridge the communication gap by converting ISL gestures into understandable output (such as text), making communication easier between deaf/mute individuals and others.

2. LITERATURE SURVEY

1) INEGI (2021) – Disability Statistics and Communication Needs. This report published by INEGI provides national-level statistics on individuals with disabilities and highlights the communication challenges faced by populations with hearing impairments. The document outlines demographic trends, prevalence of disabilities, educational barriers, and social limitations experienced by people who rely on non-verbal communication methods such as sign language. These statistics serve as a crucial foundation for understanding the social importance of sign language recognition systems. The report emphasizes the necessity of creating inclusive technologies and tools that bridge communication gaps between disabled individuals and the rest of society. By presenting real-world challenges and inequalities, this reference reinforces the importance of developing AI-driven systems such as automated sign language recognition platforms that improve accessibility, communication independence, and inclusiveness. **2) Chollet, F.** (2021) – Foundations of Deep Learning with Python. François Chollet's book, Deep Learning with Python, offers a comprehensive introduction to deep learning theory, neural network models, and practical implementation using the Keras framework. The author explains the mathematical intuition behind neural networks, the importance of

convolutional architectures for image tasks, and best practices for training efficient and scalable models. This reference is foundational for any system using CNNs or LSTMs, as it provides both theoretical knowledge and practical coding strategies that guide model development. In the context of sign language recognition, Chollet's explanations of convolution layers, feature extraction, and backpropagation directly support the design of gesture recognition networks. This reference strengthens the methodological grounding of projects involving training and deploying neural networks for classification, object detection, and image processing applications.

3)**Matel(2022**) – Margin LSTM Training for Language Models. This work presents an improved training strategy for Long Short-Term Memory (LSTM) neural networks through a large-margin objective function. The authors argue that typical training methods do not enforce sufficient separation between classes, leading to lower accuracy in challenging sequence tasks. By introducing a margin-based constraint, the model becomes more robust and better at distinguishing subtle differences in sequential patterns. While the paper focuses on neural language modeling, the underlying LSTM advancements can be applied to gesture recognition, especially dynamic signing or continuous sign translation systems. The research highlights the value of improving temporal modeling capabilities, which is essential for future extensions of ISL or ASL recognition systems that require continuous gesture sequence interpretation.

4) **Attention-Based LSTM for Action Recognition AUTHORS**: (2016) This study proposes a two-stream architecture combining spatial and temporal information processed through attention-enhanced LSTM layers for human action recognition. The model receives both RGB frames and optical flow images, enabling it to capture static posture and motion dynamics simultaneously. The attention mechanism improves the network's focus on relevant regions of the frame, reducing noise and improving accuracy. This approach is highly relevant to sign language recognition because gestures involve fine-grained movements that require both spatial and temporal analysis. The paper demonstrates how hybrid CNN–LSTM models can outperform single-stream networks, providing a valuable framework for designing advanced sign recognition systems that interpret continuous or dynamic gestures.

5) Agarwal et al. (2021) – DL Framework for Visual-to-Caption Translation Agarwal and colleagues explore a deep learning architecture that converts visual inputs into descriptive captions using CNN and RNN models. The framework extracts high-level image features using convolution layers and then passes them through LSTM-based sequence generators to produce natural-language captions. Although the paper focuses on caption generation rather than gesture recognition, its methodology is closely related to sign language translation systems. The integration of CNN feature extraction with sequence modeling demonstrates how visual information can be transformed into text — a capability that future ISL recognition systems may adopt for end-to-end gesture-to-text translation. This work is important because it shows the potential of multimodal deep learning for bridging communication gaps.

6) Indian Sign Language Generation Using GANs & Sentence Processing(2020) This study presents a deep learning framework for generating Indian Sign Language (ISL) gestures using Generative Adversarial Networks (GANs) combined with sentence processing techniques. The authors aim to automatically convert text-based sentences into sign language representations by learning the structural and grammatical properties of ISL. The use of GANs enables the creation of high-quality gesture frames that resemble real ISL signs, helping in situations where training data is limited. The paper emphasizes linguistic alignment between natural language and sign language structure, ensuring that generated gestures preserve semantic meaning. The proposed approach is particularly relevant for developing bidirectional sign language communication systems, where text or speech can be translated into sign gestures. This study contributes significantly by demonstrating that GAN-based models can support educational tools, virtual avatars, and assistive technologies for the deaf community.

PROPOSED METHODOLOGY

3.SYSTEM ARCHITECTURE:

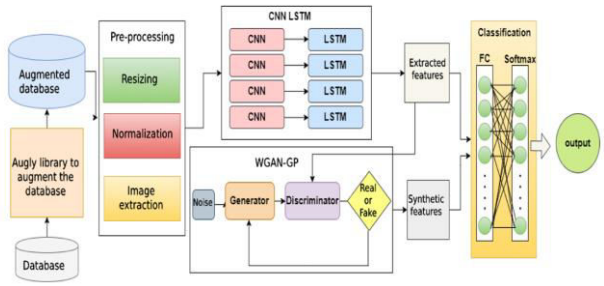


Fig 1: **system architecture**

MODULES

- Dataset & Input Acquisition
- Hand Detection & Preprocessing
- CNN-Based Classification
- Application Interface
- Prediction & Result Visualization

DESCRIPTION:

Dataset & Input Acquisition::

Manages ISL gesture image dataset (class-wise labeled folders for digits 0–9 and letters A–Y) during model training. In deployment, handles both Streamlit input modes: Upload Image (PIL reads JPG/PNG via st.file_uploader) and Webcam Capture (st.camera_input()). Output: raw RGB image forwarded to the hand detection module.

Hand Detection & Preprocessing

with margin, and crop the hand ROI. Falls back to center-crop if no hand detected. Then resizes to 64×64, Uses MediaPipe Hands (mp.solutions.hands.Hands()) to detect 21 hand landmarks, compute bounding box normalizes pixel values, converts to RGB, and adds batch dimension. Output: preprocessed numpy array of shape (1, 64, 64, 3).

CNN-Based Classification: Loads isl_sign_language_cnn_fast.h5 using keras.models.load_model(). Feeds preprocessed array to model.predict() to obtain Softmax probability vector. Applies np.argmax() to extract the predicted class index. Computes confidence as max(probabilities) × 100. Output: predicted class index + confidence score.

Application Interface:

Configures the Streamlit page (title, icon, layout). Renders radio buttons for input mode selection, corresponding input widgets (file_uploader / camera_input), preview image, and "Predict Sign" button. Sidebar displays app description, instructions, and model info. Output: triggers complete prediction pipeline on button click.

Prediction & Result Visualization:

Loads class_labels.json to map predicted index → ISL label. Displays predicted class prominently with confidence percentage in a styled success box. Renders the cropped hand region used for inference. Provides an expandable probability table showing class-wise Softmax scores. Output: complete, user-friendly recognition result.

UML Diagrams

UML stands for Unified Modeling Language. UML is a standardized general- purpose modeling language in the field of object-oriented software engineering.

The standard is managed, and was created by, the Object Management Group. The goal is for UML to become a common language for creating models of object oriented computer software. In its current form UML is comprised of **two major components:** a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems. The UML represents a collection of best engineering practices that have proven successful in the modeling of large and complex systems.

The UML is a very important part of developing objects-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects. GOALS: The Primary goals in the design of the UML are as follows:

1. Provide users a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development process.

4. Provide a formal basis for understanding the modeling language.

5. Encourage the growth of OO tools market.

6. Support higher level development concepts such as collaborations, frameworks, patterns and components.

7. Integrate best practices.

There are nine types of **UML Diagrams**

¬ Class Diagram

¬ Use case Diagram

¬ Object Diagram

¬ Sequence Diagram

¬ Collaboration Diagram

¬ State chart Diagram

¬ Activity Diagram

USE CASE DIAGRAM:

The Use Case Diagram represents the interaction between the user and the ISL Recognition System at a high level. It shows what major functions the system provides and how the user can access them. In your project, the main actor is a single User who wants to recognize an ISL hand sign using either an uploaded image or a webcam capture. The system is responsible for taking this input, processing it and returning a prediction.

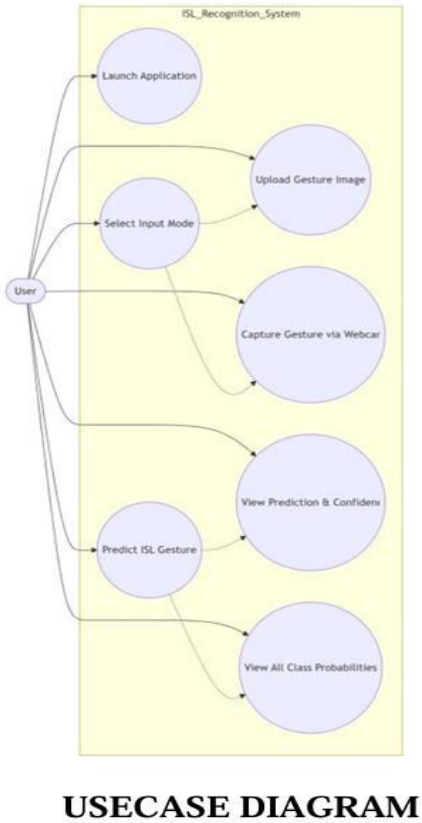


Fig 2: Use Case Diagram

CLASSDIAGRAM:

At the center is the ISLApp class, which coordinates user interaction through the Streamlit interface. It makes use of helper classes like Image Handler for managing uploaded or captured images, Hand Detector for Media Pipe-based hand cropping, Preprocessor for resizing and normalizing images, CNN Model for loading and running the trained model, and Label Manager for mapping indices to sign labels. Finally, Prediction Result stores the model output in a structured form, including predicted class and probabilities.

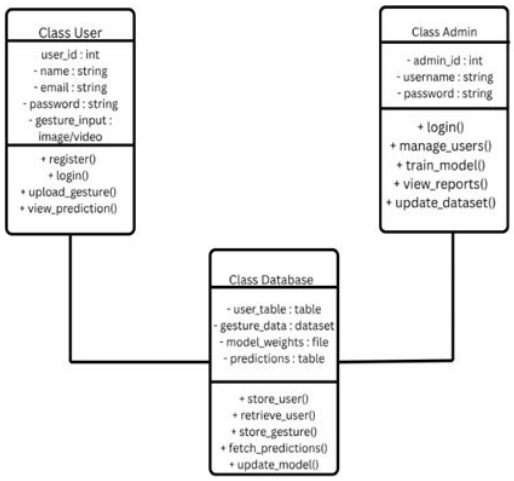


Fig 3: class diagram

SEQUENCE DIAGRAM:

The Sequence Diagram explains how the different objects of the system interact over time for a specific scenario. In your project, one of the most important scenarios is: "User captures an ISL gesture using webcam and gets the predicted sign." This diagram shows the flow of messages between the user interface and backend components in order. The sequence starts when the user chooses "Use Webcam" and captures an image. The ISLApp receives this event and calls ImageHandler to load and convert the camera image. Next, ISLApp invokes HandDetector to crop the hand region using MediaPipe. The cropped hand image is then passed to the Preprocessor, which resizes and normalizes it to meet the CNN's input requirements.

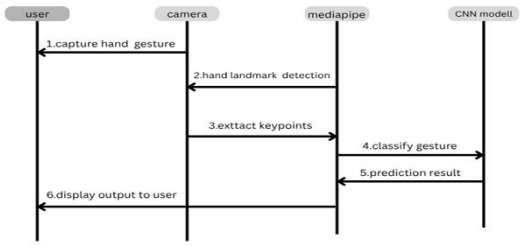


Fig 4: sequence diagram

DATABASE DESIGN

1. Storage Type:

The system primarily uses a file-based storage method. Training and testing data are stored in CSV (Comma-Separated Values) format for simplicity and ease of use.

2. Dataset Structure:

The CSV dataset includes the following fields:

- id (optional): A unique identifier for each entry
- text: The actual social media post or comment
- label: Indicates whether the text is offensive or non-offensive

3**. Data Usage**:

The data is loaded into memory using Python's pandas library during training and prediction. Preprocessed and labeled data is used to train the model. Prediction results and evaluation metrics may also be exported as CSV files for analysis.

4. Optio

nal Database Integration:

If a GUI or web application is built (e.g., using Flask), a simple database like SQLite or a FireBase-based storage system can be used to:

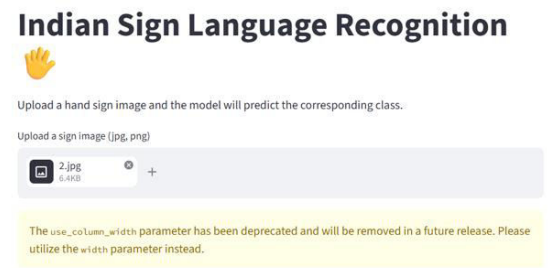
- Open website
- Upload image
- Store prediction history

Evaluation Metrics

The performance of the model is evaluated using the following standard classification metrics:

- **Accuracy**: Measures the overall percentage of correctly predicted gestures.
- **Loss (Sparse Categorical Crossentropy)**: Used during the training phase to measure how well the model is learning the mapping between gestures and labels.
- **Confusion Matrix**: Provides a summary of prediction results, showing where the model is confusing different gesture classes.
- **Classification Report**: Includes detailed metrics per class, such as:
  - **Precision**: The accuracy of positive predictions.
  - **Recall**: The ability of the model to find all positive instances.
  - **F1-Score**: The harmonic mean of precision and recall.
- **Confidence Scores**: The final Softmax layer provides a probability distribution, where the highest value represents the model's confidence in its prediction.

RESULTS:



PREDICTION SCREEN

CONCLUSION:

The project successfully implemented a real-time gesture recognition pipeline using deep learning and computer vision. By integrating MediaPipe for efficient hand detection and a Convolutional Neural Network (CNN)

for classification, the system is able to recognize ISL gestures with high accuracy and speed. Through systematic implementation and rigorous testing, the system demonstrated reliable performance under varying lighting conditions, backgrounds, and hand orientations.

The CNN model exhibited strong classification capabilities, while MediaPipe's landmark detection ensured consistent hand localization. The system's modular architecture allows seamless interaction between components and ensures smooth execution from input acquisition to final prediction.

FUTURESCOPE:

1. Support for Dynamic Gestures

Many ISL gestures involve movement, not just static hand poses. By integrating the system can evolve to recognize dynamic motions, enabling full-word or sentence-level interpretation

2. Expansion of Gesture Dataset

The system currently works with a limited set of gestures. Future versions can This will improve model generalization and robustness

3. Deployment on Mobile and Edge Devices

By converting the model using the system can be deployed on smartphones, enabling

4. Integration with Voice Output (Speech Synthesis)

This feature would allow hearing-impaired individuals to communicate directly with speaking individuals through the system.

5. Integration with 3D Hand Tracking / Depth Sensors

Depth cameras (Kinect, Intel RealSense) or 3D pose estimation can improve Media Pipe can also be extended to 3D hand skeleton modeling.

6.Real-Time ISL Learning Application

The system can be extended as an educational tool. This can promote ISL literacy in schools and communities

7.Cloud Deployment for Scalable Access

By hosting the application on platforms like AWS, Azure, or Google Cloud.

8. Continuous Model Improvement

With continuous data collection and retraining.

Conflict of interest statement

Authors declare that they do not have any conflict of interest.

REFERENCES

[1] INEGI. Estadísticas a propósito del día internacional de las personas con discapacidad (Datos Nacionales). In Comunicación Social; Comunicado de Presna Num. 713/21; INEGI: Aguascalientes, Mexico, 2021; pp. 1–5.

[2] Chollet, F. Deep Learning with Python, 2nd ed.; Manning Publications Co.: Shelter Island, NY, USA, 2021; pp. 1–20.

[3] Ma, Z.; Ma, J.; Liu, X.; Hou, F. Large Margin Training for Long Short-Term Memory Neural Networks in Neural Language Modeling. In Proceedings of the 2022 5th International Conference on Pattern Recognition and Artificial Intelligence (PRAI), Chengdu, China, 19 August 2022; Volume 5, pp. 673–677.

[4] Dai, C.; Liu, X.; Lai, J. Human action recognition using two-stream attention based LSTM networks. Appl. Soft Comput. 2020, 86, 105820. [CrossRef]

[5] Agarwal, A.; Garg, S.; Bansal, P. A Deep Learning Framework for Visual to Caption Translation. In Proceedings of the 2021 3rd International Conference on Advances in Computing, Communication Control and Networking (ICAC3N), Greater Noida, India, 17 December 2021; Volume 3, pp. 304 307.

[6] Vasani, N.; Autee, P.; Kalyani, S.; Karani, R. Generation of Indian sign language by sentence processing and generative adversarial networks. In Proceedings of the International Conference on Intelligent Sustainable Systems (ICISS), Thoothukudi, India, 5 December 2020; Volume 3, pp. 1250 1255.

[7] Jayadeep, G.; Vishnupriya, N.V.; Venugopal, V.; Vishnu, S.; Geetha, M. Mudra: Convolutional Neural Network based Indian Sign Language Translator for Banks. In Proceedings of the 2020 4th International Conference on Intelligent Computing and Control Systems (ICICCS), Madurai, India, 13 May 2020; Volume 4, pp. 1–5.

[8] Ru, T.S.; Sebastian, P. Real-Time American Sign Language (ASL) Interpretation. In Proceedings of the 2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN), Vellore, India, 5 May 2023; Volume 2, pp. 1–6.

[9] Srinivasa, K.G.; Anupindi, S.; Sharath, R.; Chaitanya, S.K. Analysis of Facial Expressiveness Captured in Reaction to Videos. In Proceedings of the 2017 IEEE 7th International Advance Computing Conference (IACC), Hyderabad, India, 7 January 2017; Volume 7, pp. 664–670.

[10] Rahman, A.I.; Akhand, Z.; Nahian, K.; Tasin, A.; Sarda, A.; Bhuiyan, S.; Rakib, M.; Ahmed Fahim, Z.; Kundu, I. Continuous Sign Language Interpretation to Text Using Deep Learning Models. In Proceedings of the 2022 25th International Conference on Computer and Information Technology (ICCIT), Cox's Bazar, Bangladesh, 19 December 2022; Volume 25, pp. 745–750.

[11] Cheng, S.; Huang, C.; Wang, Z.; Wang, J.; Zeng, Z.; Wang, F.; Ding, Q. Real-Time Vision-Based Chinese Sign Language Recognition with Pose Estimation and Attention Network. In Proceedings of the 2021 IEEE International Confer