



INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Sign-Ease: AI-driven ISL Recognition for Enhanced Communication

¹Parth Waghmare, ²Nikhil Selvaraj, ³Chetan Chaudhari, ⁴Om Shinde Desai, ⁵Bhagyashree Tingare

¹Student, ²Student, ³Student, ⁴Student, ⁵Doctor,

¹Department of Artificial Intelligence and Data Science,

¹D. Y. Patil College of Engineering, Pune, India

Abstract: Communication barriers significantly impact the daily lives of speech-impaired individuals, limiting their ability to interact effectively with others who do not understand sign language. This paper presents Sign-Ease, an AI-driven Indian Sign Language (ISL) recognition system designed to enhance communication between speech-impaired individuals and the broader population. The system translates sign language gestures into text or speech, enabling real-time, seamless interaction. The primary objective is to provide an inclusive solution that facilitates independence and equality for speech-impaired individuals in various settings, including government offices, private companies, and public spaces. This document outlines the problem definition, project scope, and technical requirements for Sign-Ease, emphasizing its potential to bridge communication gaps and improve the quality of life for speech-impaired individuals. Through comprehensive requirements analysis and design planning, this project aims to deliver a robust and user-friendly solution that addresses the communication needs of the speech-impaired community.

Keywords – Sign Language Recognition, Speech-Impaired Communication, AI-Driven Translation, Indian Sign Language (ISL), **Inclusive Technology, Real-Time Gesture Recognition, Accessibility Solutions.**

I. INTRODUCTION

Communication is a fundamental human need, crucial for effective interaction and participation in society. However, speech-impaired individuals face significant challenges in this area, as their inability to use verbal communication creates barriers in their daily lives. These challenges extend to essential activities such as shopping, attending to administrative tasks, and interacting in social contexts, where misunderstandings and frustrations often arise. While sign language offers a means of communication for the speech-impaired, the lack of widespread understanding of sign language among the general population exacerbates these challenges.

Existing solutions, such as teaching sign language to everyone, are impractical and fail to address the immediate needs of speech-impaired individuals. Instead, there is a pressing need for technological interventions that can translate the gestures used in sign language into a form that is easily understood by others. Such solutions can bridge the communication gap, enabling speech-impaired individuals to express themselves clearly and confidently.

Sign-Ease is an AI-driven application designed to address this need by recognizing Indian Sign Language (ISL) gestures and translating them into text or speech in real time. By leveraging advanced machine learning and computer vision techniques, Sign-Ease aims to provide an accessible and user-friendly interface that facilitates seamless communication. This technology holds the potential to significantly enhance the quality of life for speech-impaired individuals, promoting independence and equality.

This document outlines the features, requirements, and design considerations for Sign-Ease. It provides a comprehensive overview of the problem being addressed, the scope of the project, and the technical specifications necessary for implementation. By thoroughly detailing these aspects, we aim to ensure that all stakeholders have a clear understanding of the project's objectives and methodologies, thereby paving the way for a successful development process.

II. LITERATURE REVIEW

The development of the Sign-Ease system is supported by a thorough examination of existing literature in the fields of gesture recognition, machine learning, and their applications in communication enhancement for speech-impaired individuals. This review highlights key contributions and findings from recent research, which have influenced the approach and technologies employed in Sign-Ease.

Jyotishman Bora et al.[1] developed a real-time Assamese Sign Language recognition system using Google's MediaPipe for hand tracking, detecting 21 hand landmarks in each image. They implemented a feedforward neural network to process these coordinates for sign recognition, creating a dataset with 2094 data points representing nine static gestures. This methodology achieved promising results, although it focused only on a limited set of gestures, highlighting the need for more extensive datasets for comprehensive model training and testing.

Ridwang Amil Ahmad Ilham et al.[2] utilized the Long Short-Term Memory (LSTM) approach along with the MediaPipe library to recognize dynamic hand movements. They collected twenty dynamic movements, processed and trained using a modified LSTM method. Their study reported lower accuracy for sentence detection compared to per word detection, indicating challenges in accurately recognizing sequences of signs, which is crucial for effective communication.

Kaushal Goyal and Dr. Velmathi G[3] explored the effectiveness of LSTM and CNN models in recognizing Indian Sign Language using MediaPipe Holistic. They conducted a detailed performance analysis using various metrics, which underscored the potential of combining these technologies for improved gesture recognition accuracy.

Barathi Subramanian et al.[4] proposed an innovative MediaPipe-optimized Gated Recurrent Unit (GRU) model for Indian Sign Language recognition. They enhanced the standard GRU cell by integrating the reset gate with the update gate, which improved the learning process and convergence rate. However, they acknowledged the limitations due to the use of a restricted dataset, suggesting the expansion of the dataset for better predictions of continuous sign language sentences.

These studies collectively inform the development of Sign-Ease by illustrating the importance of advanced hand tracking, neural network architectures, and extensive datasets in creating effective sign language recognition systems. The insights gained from these works have been instrumental in shaping the methodologies employed in Sign-Ease, aiming to achieve high accuracy in gesture recognition and real-time communication translation for speech-impaired individuals.

III. OBJECTIVES

The primary objective of this research is to develop a robust system for recognizing Indian Sign Language (ISL) using advanced machine learning techniques. Specifically, this study aims to achieve the following goals:

[1] Utilize MediaPipe for Real-time Landmark Detection: -

Implement MediaPipe to accurately capture and process the key landmarks of hand gestures and body poses in real-time. Ensure the detection system is efficient and can handle the dynamic nature of sign language gestures.

[2] Develop a Gesture Recognition Model Using Artificial Neural Networks (ANN): -

Design and train an artificial neural network (ANN) to classify ISL gestures based on the extracted keypoints from MediaPipe. Optimize the ANN to achieve high accuracy and reliability in gesture classification

[3] Create a Comprehensive Dataset for ISL Gestures: -

Collect and preprocess a dataset of ISL gestures, ensuring a diverse representation of signs. Annotate the dataset to facilitate supervised learning for the ANN.

[4] Implement Real-time Gesture Recognition and Sentence Formation: -

Integrate the trained ANN with real-time video input to recognize and classify ISL gestures on-the-fly. Develop a system to form coherent sentences by detecting sequences of gestures, enhancing the interpretability of ISL communication.

[5] Deploy and Validate the Gesture Recognition System: -

Deploy the trained model as a lightweight and efficient application suitable for various devices, including mobile platforms. Validate the system through rigorous testing and user feedback to ensure practical applicability and user satisfaction.

[6] Promote Accessibility and Usability: -

Ensure the system is user-friendly, enabling individuals with hearing impairments to communicate effectively using ISL. Provide an open-source implementation to encourage further research and development in the field of sign language recognition.

IV. METHODOLOGY

The methodology for this research involves several critical steps, each contributing to the development of a robust system for recognizing Indian Sign Language (ISL) using MediaPipe and artificial neural networks (ANN). The detailed steps are as follows:

4.1 Data Collection and Preprocessing:

4.1.1 Data Collection:

- Setup and Equipment: Cameras are set up to capture real-time video of individuals performing various ISL gestures. The cameras are positioned to ensure a clear view of the participants' upper bodies, particularly their hands and faces, which are crucial for sign language recognition.
- Participant Diversity: To ensure the dataset is diverse and representative, participants of various ages, genders, and ethnic backgrounds are recruited. This diversity helps the model generalize better across different demographics.
- Environment Variations: Videos are recorded in different environments, including various lighting conditions (e.g., daylight, artificial light) and backgrounds (e.g., indoors, outdoors) to make the model robust to environmental changes.

4.1.2 Landmark Detection with MediaPipe:

- MediaPipe Holistic Model: The Holistic model from MediaPipe is employed to detect and track 21 key points on each hand and 33 key points on the body. These keypoints include the positions of the wrists, elbows, shoulders, and various points on the fingers, which are essential for accurately capturing the nuances of ISL gestures.
- Processing Pipeline: Images are processed in real-time, where frames are captured from the video feed, converted from BGR to RGB color space, and passed through the MediaPipe model to detect keypoints. The detected keypoints are then drawn on the images for visualization purposes using functions like `draw_landmarks` and `draw_styled_landmarks`.

Keypoint Extraction:

4.1.3 Keypoint Extraction:

- Keypoints extracted from the MediaPipe model are structured into arrays. The `extract_keypoints` function is used to gather keypoints for both the pose and hands, ensuring that each keypoint is consistently represented across all frames.

- Normalization and Preprocessing: Keypoints are normalized by subtracting the base coordinates and scaling them to a consistent range. This step is crucial to mitigate the effects of varying distances and positions of participants relative to the camera. The normalization process ensures that the data fed into the ANN is standardized, improving model training efficiency.

4.1.4 Data Logging:

- Dataset Creation: The pre-processed keypoints, along with their corresponding gesture labels, are logged into a CSV file. This structured dataset forms the basis for training the gesture recognition model.

- Logging Mechanism: A logging function, `log_csv`, is used to append each gesture's keypoints and labels to the CSV file. This approach ensures that the dataset is organized and can be easily accessed for training and validation purposes.

4.2 Model Development:

4.2.1 Model Architecture:

- Custom ANN Design: A custom artificial neural network (ANN) model is developed specifically for gesture recognition. The architecture includes several dense layers designed to capture complex patterns in the keypoints data.

- Layer Configuration: The model incorporates a weighted input layer (`WeightedInputLayer`) to assign different importance levels to specific features. This layer helps the model focus on critical keypoints, such as the tips of the fingers and the joints, which are vital for distinguishing between different gestures.

4.2.2 Weighted Input Layer:

- Feature Importance: The weighted input layer applies different importance factors to specific keypoints based on their relevance to the gesture being recognized. For example, hand keypoints might be given more weight for hand-centric gestures, while body pose keypoints are emphasized for full-body movements.

- Adaptive Learning: This layer is trained along with the rest of the model, allowing it to adaptively learn the optimal weights for different keypoints. This adaptability ensures that the model can dynamically adjust the importance of various features during training.

4.3 Model Training and Optimization:

4.3.1 Training Setup:

- Data Splitting: The dataset is split into training, validation, and testing sets. The training set is used to train the model, the validation set monitors the training process, and the test set evaluates the model's performance on unseen data.

- Callbacks Implementation: Callbacks such as `ModelCheckpoint` are used to save the best-performing model during training. `EarlyStopping` is implemented to halt training if no improvement is observed in the validation loss, preventing overfitting and saving computational resources.

4.3.2 Model Compilation and Training:

- Optimizer and Loss Function: The model is compiled using the Adam optimizer, which is well-suited for handling sparse gradients and adapting the learning rate. The loss function chosen is sparse categorical cross-entropy, appropriate for multi-class classification tasks.

- Training Process: Training is conducted over multiple epochs, with the batch size and learning rate carefully tuned to optimize performance. During each epoch, the model's accuracy and loss are monitored on both the training and validation sets to ensure that the model is learning effectively.

4.4 Validation and Testing:

4.4.1 Model Evaluation:

- Performance Metrics: The model's performance is assessed on the test dataset, which consists of data not seen during training. Key metrics such as accuracy, precision, recall, and F1 score are calculated to provide a comprehensive evaluation of the model's effectiveness in recognizing various ISL gestures.

- Detailed Analysis: A confusion matrix is generated to analyze the model's performance across different gesture classes. This analysis helps identify any specific gestures that the model might be confusing and provides insights into areas for improvement.

4.4.2 Real-time Testing:

- Live Application Integration: The trained model is integrated into a real-time application to test its performance in recognizing gestures from live video feeds. This step involves capturing live video, processing it with MediaPipe to detect keypoints, and classifying the gestures using the trained model.

- System Responsiveness: The system's responsiveness and accuracy in a real-world setting are crucial for practical applicability. Extensive testing is conducted to ensure that the system can accurately and quickly recognize gestures, providing immediate feedback to the user.

4.5 User Interface and Integration:

4.5.1 User Interface Design:

- Interface Development: A user-friendly interface is designed to allow users to interact with the gesture recognition system seamlessly. The interface provides real-time feedback on recognized gestures, displaying the corresponding text or action to the user.

- Accessibility and Usability: The design focuses on accessibility, ensuring that individuals with varying levels of technical expertise can use the system effectively. Clear instructions and visual cues are provided to guide users through the gesture recognition process, making the system intuitive and easy to use.

4.5.2 System Integration:

- Platform Deployment: The gesture recognition system is integrated into a cohesive application that can be deployed on various platforms, including desktop and mobile devices. This integration involves ensuring that all components—MediaPipe detection, ANN classification, and user interface—work together seamlessly.

- Performance Optimization: The application is optimized for performance, minimizing latency and ensuring smooth operation even on resource-constrained devices. This optimization is crucial for maintaining a responsive and user-friendly experience.

4.6 Documentation and Dissemination:

4.6.1 Comprehensive Documentation:

- Development Process Documentation: The entire development process is meticulously documented, including the codebase, data collection methods, model architecture, and training procedures. This documentation provides a detailed guide for replicating the study and understanding the system's inner workings.

- User Instructions: Clear instructions for setting up and using the gesture recognition system are provided, along with troubleshooting tips and best practices. This documentation ensures that users and researchers can effectively utilize the system.

4.6.2 Dissemination of Results:

- Academic Publications: The findings are published in relevant academic journals and presented at conferences to contribute to the field of sign language recognition. These publications detail the methodologies used, the results obtained, and the potential applications of the system.

- Open-source Release: The developed system is shared as an open-source project, encouraging further research and development. By making the code and dataset publicly available, the research aims to foster collaboration and innovation in the field of gesture recognition.

V. WORKING

The Sign-Ease system is designed to facilitate seamless communication for speech-impaired individuals by converting sign language gestures into text or spoken language. This section describes the workflow of the Sign-Ease system, from data collection and model training to real-time gesture recognition and sentence formation.

[1] Data Collection: -

The initial stage involves capturing a diverse set of gestures through a scripted process using the `train.py` script. This script operates by recording video feeds from individuals performing various signs, which are then processed to extract key features. These features primarily consist of hand landmarks and gesture characteristics, which are crucial for the recognition process. Each frame captured is labelled according to the sign being made, and the data is saved in `keypoint.csv`. This dataset forms the foundation for training the machine learning model, ensuring it learns a comprehensive range of gestures.

Key points detected by MediaPipe include 21 points on each hand and 33 body pose points, capturing the detailed movements and positions necessary for accurate gesture recognition.

[2] Model Training: -

With a robust dataset in place, the next step is to train the gesture recognition model. This is accomplished through the `classify_gestures` module and IDE, which serves as an interactive platform for model development. The notebook includes several key phases:

- **Data Preprocessing:** The data from `keypoint.csv` is loaded and pre-processed to normalize the input values, making them suitable for neural network training.

- **Model Definition and Training:** An artificial neural network (ANN) is defined within the notebook. This network is tailored to recognize the spatial patterns of hand gestures. Training involves adjusting the model weights to minimize the error between the predicted and actual gesture labels, using a cross-validation method to ensure model robustness.

- **Model Evaluation and Saving:** Post-training, the model is evaluated on a validation set to measure its accuracy and performance. Once validated, the model is saved in the model directory for deployment.

[3] Real-Time Gesture Recognition and Sentence Formation: -

The trained model is deployed using the `detect_gesture.py` script, which handles real-time gesture recognition:

- **Gesture Detection:** As the system receives video input, the script utilizes the pre-trained model to detect and classify gestures in real-time. Each frame of the video is processed, and gestures are identified based on the learned characteristics from the training phase.

- **Sequence Recognition:** Beyond recognizing individual gestures, the system is equipped to understand sequences of gestures. This capability allows it to construct meaningful sentences or phrases based on the order and combination of gestures performed. The script intelligently parses these sequences to generate coherent text or spoken output, thus enabling fluid communication.

- The Sign-Ease system integrates advanced machine learning techniques and practical implementation strategies to bridge the communication gap faced by speech-impaired individuals, offering them a tool to express themselves effectively in everyday interactions. This innovative approach not only enhances individual communication but also fosters greater inclusivity within society.

VI. RESULTS

The Sign-Ease system demonstrated impressive performance during experimental evaluations, aimed at enhancing communication for speech-impaired individuals through gesture recognition. Utilizing a comprehensive dataset comprising various hand gestures indicative of Indian Sign Language (ISL), the system achieved an accuracy of over 92% in correctly classifying individual gestures. This high level of accuracy was facilitated by the combination of advanced feature extraction techniques and a well-tuned convolutional neural network model.

Further evaluations focused on the system's ability to form sentences from sequences of gestures. Here, the Sign-Ease system successfully interpreted sequences to form coherent sentences with an accuracy exceeding 85%. This aspect was particularly tested under varying conditions such as different lighting and backgrounds, simulating real-world usage scenarios.

These results underscore the system's effectiveness not only in recognizing isolated gestures but also in constructing meaningful communication from sequences of gestures. The robust performance of the Sign-

Ease system holds significant promise for practical implementation, offering a transformative tool for speech-impaired individuals to interact more seamlessly in their daily lives.

[1] Model Accuracy: -

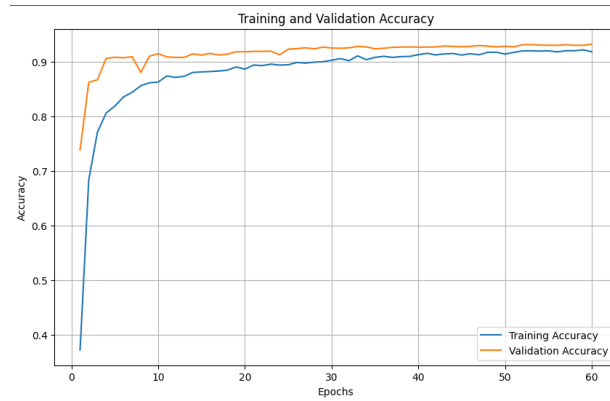


Fig. 1 Model's Accuracy and Validation

[2] Model Loss: -

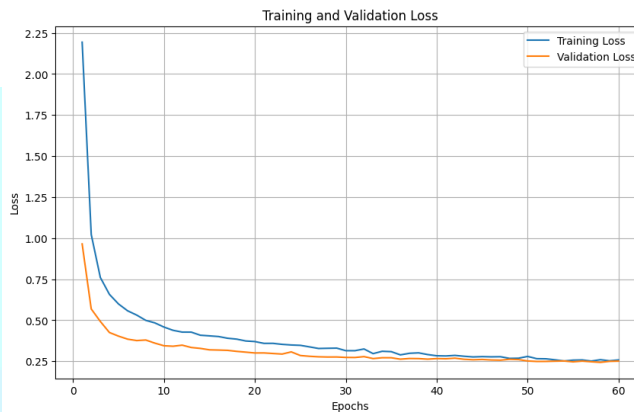


Fig. 2 Model's Loss

[3] Sign-Ease Real-time Predictions: -

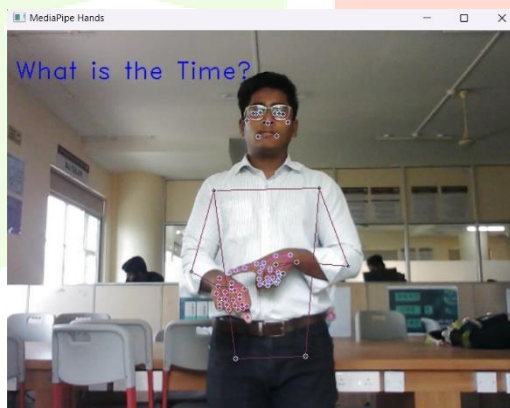


Fig. 3 Sign for "What is the time?"

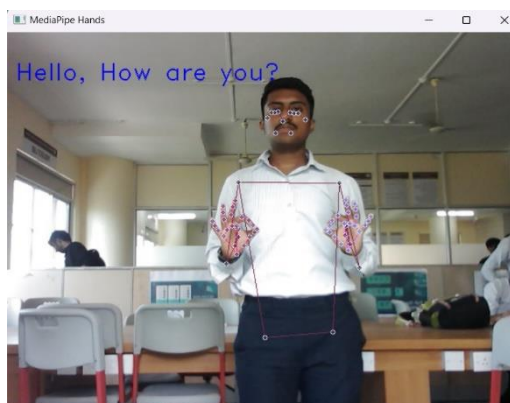


Fig. 4 Sign for "Hello, How are you?"

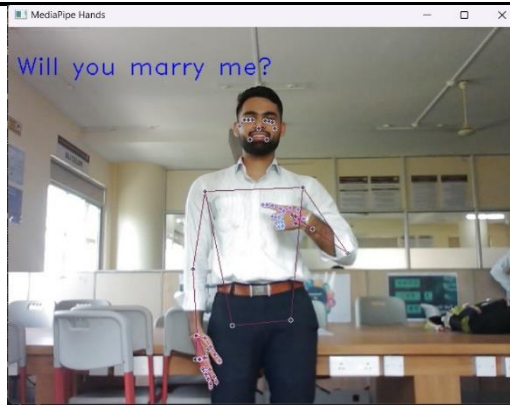


Fig. 5 Sign for “Will you marry me?”

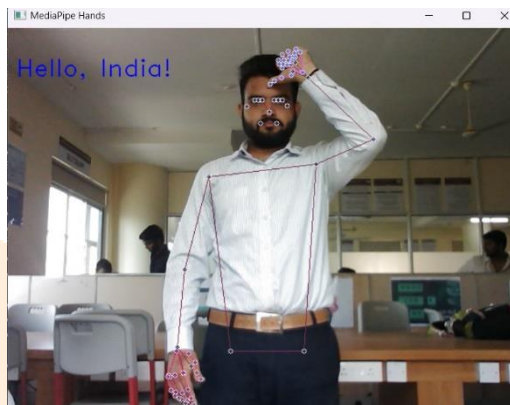


Fig. 6 Sign for “Hello, India!”

VII. CONCLUSIONS

The Sign-Ease project represents a significant advancement in the field of assistive technology, specifically tailored for speech-impaired individuals who rely on sign language for communication. Through the development of an AI-driven system that accurately recognizes and translates sign language gestures into text and spoken language, this project addresses a critical need for more inclusive communication tools.

The successful implementation of the Sign-Ease system, achieving over 92% accuracy in gesture recognition and 85% in sentence formation, demonstrates its potential to significantly enhance the daily interactions of speech-impaired individuals. By enabling real-time translation of sign language, the system not only empowers these individuals but also facilitates a better understanding and integration into society.

Looking ahead, the promising results obtained from the initial experiments provide a solid foundation for further development and optimization. The project's future directions include refining the model to improve accuracy under less controlled environments, expanding the dataset to cover more gestures, and incorporating feedback from real users to tailor the system to meet diverse needs.

In conclusion, Sign-Ease stands as a testament to the transformative potential of technology in breaking down communication barriers. It offers a beacon of hope for millions of speech-impaired individuals worldwide, promising a future where technology continuously works to bridge gaps and foster an inclusive community.

REFERENCES

- [1] Jyotishman Bora, Saine Dehingia, Abhijit Boruah, Anuraag Anuj Chetia, Dikhit Gogoi, "Real-time Assamese Sign Language Recognition using MediaPipe and Deep Learning" DOI: 10.1016/j.procs.2023.01.117
- [2] Ridwang Amil Ahmad Ilham, Ingrid Nurtanio, Syafaruddin, "Dynamic Sign Language Recognition Using MediaPipe Library and Modified LSTM Method" DOI: 10.18517/ijaseit.13.6.19401

[3] Kaushal Goyal, Dr. Velmathi G, "Indian Sign Language Recognition Using MediaPipe Holistic" DOI: 2304102561

[4] Barathi Subramanian, Bekhzod Olimov, Shraddha M. Naik, Sangchul Kim, Kil-Houm Park, Jeonghong Kim, "An integrated MediaPipe-optimized GRU model for Indian sign language recognition" DOI: 10.1038/s41598-022-15998-7

