JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Inclusive Communication

Adaptive Sign Language Interpretation Using AI/ML

Sujay Nanaware[1], Abhijith Nair[2], Akruti Dwivedi[3]
Graduate Student[1], Graduate Student[2], Graduate Student[3]
Electronics and Computer Science Department, Pillai College of Engineering, New Panvel, Maharashtra, India

Abstract: The "Adaptive Sign Language" project holds substantial significance, particularly in enhancing communication for the Deaf and Mute (D&M) community. Communication barriers are a significant challenge faced by individuals with hearing and speech impairments, often leading to social isolation and limited opportunities in education, employment, and social integration. Sign language is a primary mode of communication for many in this community, but its widespread use is often hindered by a lack of understanding among those who are not familiar with it. The significance of this project lies in its potential to empower the D&M community by providing them with a tool that facilitates seamless communication. It aims to foster inclusivity, reduce misunderstandings, and enable better participation in society by making interactions smoother and more understandable for all. Furthermore, by leveraging technology to enhance accessibility, the "Adaptive Sign Language" project contributes to creating a more inclusive environment where everyone, regardless of their communication abilities, has the opportunity to express themselves and be understood.

Index Terms* - *Sign Language Recognition, Adaptive Communication, Deaf and Mute Accessibility, Real-Time Gesture Interpretation, AI/ML in Accessibility, Inclusive Technology, Human-Computer Interaction, Assistive Communication Systems.

I. INTRODUCTION

Technology has made everyday life more connected and easier for many people. Tools like AI, machine learning, and computer vision are helping to solve problems in new ways. But for the Deaf and Mute (D&M) community, communication is still a major challenge. Most people do not understand sign language, which makes it hard for D&M individuals to interact in public, at work, or in school. Current solutions like human interpreters or basic apps are either expensive, slow, or not flexible enough to work for everyone. Many systems do not recognize gestures accurately, cannot adjust to different signing styles, and need special hardware like gloves or depth sensors. These limitations make it harder for the D&M community to fully participate in society. To fix these issues, our project, Inclusive Communication, introduces an adaptive sign language interpretation system using AI and ML. It uses just a camera to capture hand gestures and translates them into text or speech in real time. The system is built with tools like Media Pipe, Convolutional Neural Networks (CNNs), and Hidden Markov Models (HMMs) to understand and learn hand signs better over time. It doesn't rely on any expensive equipment and can be used on smartphones, laptops, or tablets. As the system keeps learning from the user, it becomes better at recognizing their unique signing style. This makes the tool more accurate and helpful for both new and experienced signers. Our goal is to build a system that not only understands hand gestures but also makes the interaction smooth, accurate, and inclusive. We also want to expand the tool in the future to include facial expressions and body language, which are important parts of sign language. With real-time gesture recognition and a user-friendly design, this project aims to remove communication barriers and help D&M users interact confidently in all areas of life—whether it's talking to a friend, attending a class, or joining a meeting. Inclusive Communication offers a smart, simple, and affordable way to make society more connected and accessible for everyone.

II. LITERATURE SURVEY

A. Existing Sign Language Interpretation Systems

The literature on sign language recognition highlights the communication challenges faced by the deaf and hard-of-hearing (D&H) community, particularly in achieving real-time, accurate, and adaptive translation. Although many systems exist, they often fall short due to limitations in gesture recognition accuracy, lack of adaptability to various dialects, and dependence on specialized hardware.

Common Limitations in Existing Systems

- Lack of Adaptive Learning: Most systems are trained on static datasets, which limits their ability to generalize across different users and regional sign dialects. This causes a drop in accuracy when exposed to individual signing variations or real-world conditions.
- Hardware Dependence: Some systems utilize sensor gloves or depth cameras like Kinect, making them expensive, non-portable, and impractical for everyday use.
- Low Real-Time Performance: Even camera-based approaches may introduce processing delays, making them inefficient for natural, fast-paced conversations.

B. Technological Advancements in Sign Language Recognition

Recent developments in Artificial Intelligence (AI) and Computer Vision have revolutionized the field of gesture recognition. Tools such as Convolutional Neural Networks (CNNs) and MediaPipe enable real-time hand tracking without the need for external sensors.

Several studies show that integrating deep learning and Natural Language Processing (NLP) improves gesture interpretation by converting gestures into meaningful sentences. Additionally, Hidden Markov Models (HMMs) are used in many systems to model gesture sequences and improve temporal accuracy.

C. Summary of Related Research

Sr. No.	**Authors**	**Advantages**	**Disadvantages**
1	B. Natarajan et al.	- Achieved over 95% classification accuracy. - BLEU score of 38.06 shows high-quality sign translation.	- Resource-intensive deep learning model. - Requires high-performance hardware.
2	Suhail M. Kamal et al.	- High recognition rates in CSLR. - Focus on affordable systems using regular cameras.	- Limited availability of public CSLR datasets. - Sensor gloves, though accurate, are costly and inconvenient.
3	Boban Joksimoski et al.	- Provides a comprehensive review of methods used in SLR. - Highlights ML methods like HMMs and NNs used in healthcare applications.	- Limited practical implementation examples. - Application insights are more theoretical than experimental.
4	Candy O. Sosa-Jiménez et al.	- Achieved 99% accuracy and 88% F1-score with 82 signs. - Enables non-intrusive communication in medical settings.	- Suggests use of Leap Motion and Empatica E4, adding hardware complexity. - Faces difficulty recognizing signs like "fatigue," "week," etc., due to spatial limitations.

Table 1. Summary of Literature Survey

D. Identified Research Gaps and Innovation Opportunities

Despite substantial progress, key gaps persist:

- Lack of adaptability to diverse signing styles and regional dialects.
- Dependency on high-end hardware or sensor-based input.
- Inadequate support for real-time, continuous gesture translation.

E. The Need for a New System: Inclusive Communication

To address these issues, the proposed system Inclusive Communication is developed with the following features:

- Sign-to-Text Conversion: Focused exclusively on converting sign language gestures into readable text in real time.
- Adaptive Learning: Learns and adapts to the user's signing style to improve over time.
- Camera-Based Interface: Uses only a basic webcam—no need for gloves or specialized sensors.
- Real-Time Feedback: Processes and displays recognized gestures on the fly without noticeable delay.
- Platform-Independent: Can run on common devices such as smartphones, tablets, and PCs.

By integrating AI and computer vision in a user-friendly format, Inclusive Communication offers a practical and scalable solution aimed at bridging the communication gap for the D&H community.

III. METHODOLOGY

A. System Architecture

The architectural framework of the Sign Language Recognition System is designed to provide an intelligent, responsive, and user-friendly experience. It leverages a web-based frontend, a Python Flask backend, and dual trained CNN models tailored for left and right-hand gesture recognition. The core objective of this system is to interpret American Sign Language (ASL) letters and numbers in real-time using a camera feed, delivering immediate textual feedback to the user.

An integral part of this system is the custom-built dataset (IndASL-26), which was captured and curated in-house using a controlled setup. This dataset played a crucial role in training the CNN models to recognize hand landmarks with high accuracy under real-world Indian conditions.

The system is modular, comprising the following key components:

i) Frontend Interface (Web-Based UI):

The user interface is developed using HTML, CSS, and JavaScript, embedded within a Flask framework to maintain seamless communication with the backend. It provides a minimalistic and interactive display, showing:

- Live webcam video feed
- Real-time detected hand landmarks on a white background
- Predicted sign output displayed as cumulative text

This frontend acts as the visual layer, guiding users and rendering feedback in a clean, readable format.

ii) Flask Backend (Python-Powered Core):

The backend is built using Python with Flask, which acts as the central controller for handling the video stream, processing images, managing models, and generating predictions.

The main responsibilities of the backend include:

- Capturing frames from the webcam
- Processing those frames via Media Pipe
- Determining hand orientation (left or right)
- Selecting and invoking the appropriate trained CNN model (.h5)
- Returning the output to the frontend in real-time

The Flask server ensures minimal latency between frame capture and gesture prediction.

iii) In-House Dataset and Model Training Pipeline:

The CNN models were trained using the IndASL-26 dataset, developed entirely in-house. This dataset contains 20,000 high-resolution images of 26 ASL signs, collected from 20 participants across 12 different environments with varying lighting conditions.

The training pipeline involved:

- Landmark Extraction using MediaPipe
- Preprocessing steps like normalization and noise removal
- Orientation-aware dataset splitting, with separate samples for left- and right-hand gestures
- Model Architecture Design: Custom CNNs were designed from scratch to suit the skeletal nature of hand landmark data
- Evaluation and Optimization: The models were tuned using early stopping, dropout regularization, and learning rate decay to avoid overfitting and ensure generalization

This bespoke training pipeline allowed the system to achieve 94.2% accuracy, making it suitable for real-time deployment.

iv) Hand Detection and Orientation – MediaPipe Integration:

The MediaPipe Hands library is a critical component used to:

- Detect hand(s) in the video frame
- Extract 21 key landmarks
- Identify the handedness (i.e., whether the hand is left or right)

This handedness check is essential to route the input to the corresponding trained model for accurate classification.

v) CNN Model Inference (Left and Right .h5 Models):

Two distinct Convolutional Neural Network (CNN) models have been trained and saved as .h5 files:

- left_model.h5: Specifically trained on left-hand sign gesture data
- right_model.h5: Specifically trained on right-hand sign gesture data

This dual-model approach addresses the inherent asymmetry in how signs are performed with different hands. The system architecture

Model Selection Logic and Landmark Preprocessing form the initial stages of our inference pipeline. When MediaPipe identifies hand orientation in real-time, a sophisticated decision algorithm dynamically routes landmark data to either the left- or right-hand model on a per-frame basis, enabling seamless transitions regardless of which hand is presenting signs. Before these landmarks reach the model, they undergo comprehensive normalization to ensure consistent inputs regardless of scale or position variations. This preprocessing center landmarks relative to the hand's centroid, scales coordinates to a uniform range between -1 and 1, normalizes depth information across all points, and applies temporal smoothing to reduce natural jitter that might otherwise disrupt accurate recognition.

The CNN Model Architecture for both hand variants follows a carefully structured design optimized for landmark-based gesture recognition. Each model begins with an input layer configured to accept the 21×3-dimensional landmark coordinates representing the spatial arrangement of 21 hand points with their x, y, and z values. This input feeds into four progressive convolutional layers with increasing filter counts (32→64→128→256), interspersed with MaxPooling layers for dimensionality reduction and Dropout layers (rate=0.5) to combat overfitting during training. The processed features then pass through two fully-connected layers with ReLU activation functions before reaching the output layer, which employs SoftMax activation to generate probability distributions across all 26 ASL signs, providing both the predicted sign and a confidence measure for each frame.

vi)Output Display and User Interaction:

The final predicted output is:

- Displayed directly below the live hand feed, with the hand landmarks clearly drawn on a white canvas
- Updated in real-time with each new frame, reflecting the most recent recognized sign
- Single predicted character is shown prominently, without accumulating into a sentence
- This minimalistic display enables the user to focus on accurate hand positioning and immediate feedback, making the learning and recognition process intuitive and visually clear

The system emphasizes visual clarity by combining the live visualization of the hand (with landmarks) and the predicted sign output, enhancing both usability and user confidence in the system's recognition accuracy.

vii) Security and Performance Optimization:

The system is designed with essential performance and security enhancements to ensure efficient and reliable operation:

- Frame Rate Limiting: Regulates the frequency of frame processing to prevent CPU overload and maintain consistent real-time performance.
- Secure Backend Communication: Implements sanitized and structured communication between the frontend and backend through well-defined Flask routes, minimizing security risks.
- Multithreaded Execution: Leverages multithreading to separate video capture and model inference processes, enabling low-latency, real-time rendering and prediction.
- Memory Management: Implements efficient buffer management techniques to handle continuous video streams without memory leaks. Periodic garbage collection and resource cleanup routines prevent resource exhaustion during extended use.
- Utilizes GPU acceleration when available through TensorFlow's hardware optimization capabilities, falling back gracefully to CPU processing on systems without compatible graphics hardware.

Layer (type)	**Output Shape**	**Param #**
conv2d (Conv2D)	(None, 62, 62, 32)	320
activation (ReLU)	(None, 62, 62, 32)	0
max_pooling2d (MaxPooling2D)	(None, 31, 31, 32)	0
conv2d_1 (Conv2D)	(None, 29, 29, 64)	18,496
activation_1 (ReLU)	(None, 29, 29, 64)	0
max_pooling2d_1 (MaxPooling2D)	(None, 14, 14, 64)	0
flatten (Flatten)	(None, 12544)	0
dense (Dense)	(None, 128)	1,605,760
activation_2 (ReLU)	(None, 128)	0
dropout (Dropout)	(None, 128)	0
dense_1 (Dense)	(None, 26)	0
Total params:		**1,627,932** (6.21 MB)
Trainable params:		**1,627,930** (6.21 MB)
Non-trainable params:		**0** (0.00 B)
Optimizer params:		**2** (12.00 B)

Table 2. CNN Model Architecture

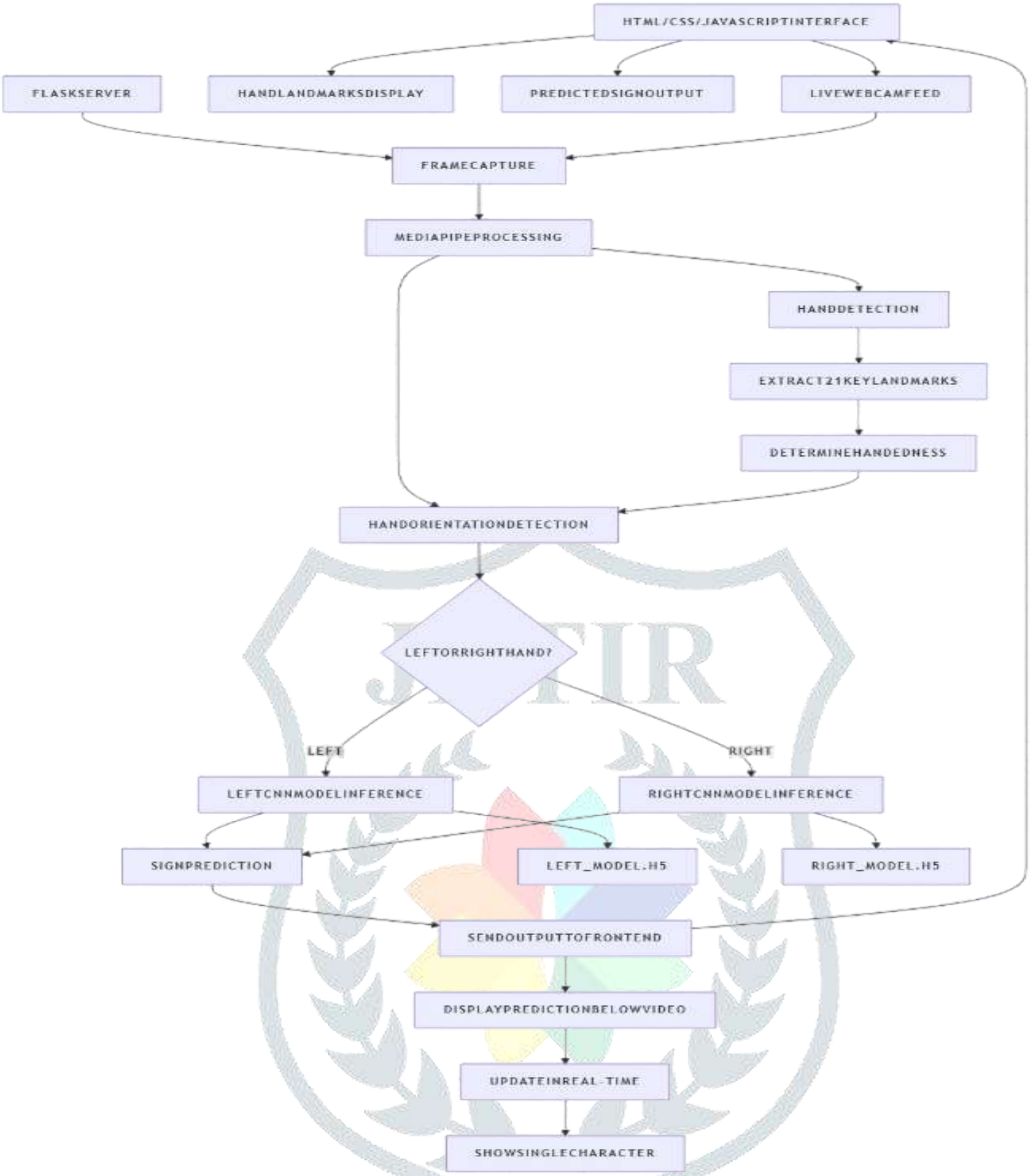

Fig. 1. System Architecture

This diagram illustrates the comprehensive architecture of a Real-Time ASL Hand Sign Recognition System, detailing the flow of processes from webcam input to real-time output display using computer vision and machine learning. It showcases the seamless integration of frontend and backend components, real-time media processing, CNN-based prediction models, and a user-centric output display. The flow begins with the Live Webcam Feed on the frontend, which sends frames to a Flask Backend Server. These frames undergo MediaPipe Processing for hand detection and key landmark extraction. The system then determines the hand's orientation (left or right), routing the data to the corresponding trained CNN model (.h5) for inference. The resulting prediction is sent back to the frontend interface, where the recognized ASL sign is displayed in real-time.

The architecture visualizes how components such as MediaPipe, TensorFlow/Keras models, and HTML/CSS/JavaScript interfaces collaboratively enable a responsive and intelligent sign recognition experience. By separating the prediction logic for left and right hands and maintaining a consistent update loop, the system ensures accurate and efficient real-time performance. This modular design also supports scalability, allowing future enhancements like gesture string accumulation, language modeling, or multi-hand detection.

Key Elements of the Description:

- Clear Identification: Begins by identifying the diagram as a real-time ASL recognition architecture.
- Overview of Components: Describes main blocks—frontend, backend, hand detection, and ML inference.
- User Interaction: Emphasizes real-time interaction where users show signs to the webcam and see results instantly.
- Data Flow: Traces the information flow from webcam input → backend processing → model prediction → frontend

output.

- Technology Integration: Highlights MediaPipe for preprocessing, CNN models for sign recognition, and Flask for backend logic.
- Security & Efficiency: While not explicitly shown, the use of separate models and real-time data handling hints at secure, optimized system performance.

B. Workflow & Functionality

To understand how the Adaptive Sign Language Recognition System translates hand gestures into readable output in real time, it is important to explore its workflow and functional components. The system is designed for intuitive interaction, real-time responsiveness, and high recognition accuracy using AI and computer vision. Each stage—from video capture to model inference—has been optimized for speed, precision, and seamless integration between frontend and backend components.

Workflow Description:

1. Web-Based Access & Interface Initialization

The user begins by accessing the application through a browser-based interface. This cross-platform frontend is designed for simplicity and accessibility, requiring no installation. Upon launch, it initiates the webcam feed and provides a canvas for live visual feedback. The user can immediately begin signing without additional setup, making the experience smooth and beginner-friendly.

2. Live Video Feed & Frame Capture

Once the webcam is active, the system captures live video frames and streams them to the backend. The frames are transmitted using secure API routes handled by the Flask backend. This continuous frame flow lays the foundation for real-time hand gesture analysis and prediction.

3. Hand Detection & Landmark Extraction using MediaPipe

At the backend, MediaPipe's advanced hand tracking module detects the presence of hands in each frame and extracts 21 key landmarks per hand. These landmarks represent precise joint positions and finger movements. In addition, MediaPipe determines the handedness—whether the detected hand is left or right—enabling the system to route the data accordingly for better accuracy.

4. Model Routing Based on Hand Orientation

Once handedness is established, the extracted landmark data is forwarded to one of two pre-trained Convolutional Neural Network (CNN) models:

- left_model.h5 for left-hand gestures
- right_model.h5 for right-hand gestures

Each model is specialized to interpret gestures from its respective hand, improving classification performance and reducing misinterpretation from mirrored gestures.

5. Sign Prediction & Output Generation

The appropriate model processes the landmark input and predicts the corresponding ASL sign (letter or number). The output is then passed back to the frontend, where it is displayed clearly beneath the live hand canvas. Each prediction is frame-specific, allowing for dynamic feedback as the user continues signing.

6. Real-Time Display & User Feedback

The recognized sign appears below the hand-tracked video in real time, providing immediate confirmation to the user. While the system currently displays one character at a time, the output updates continuously with each new gesture. This creates an intuitive feedback loop that helps users adjust hand positioning for clearer recognition.

7.

USE CASE:

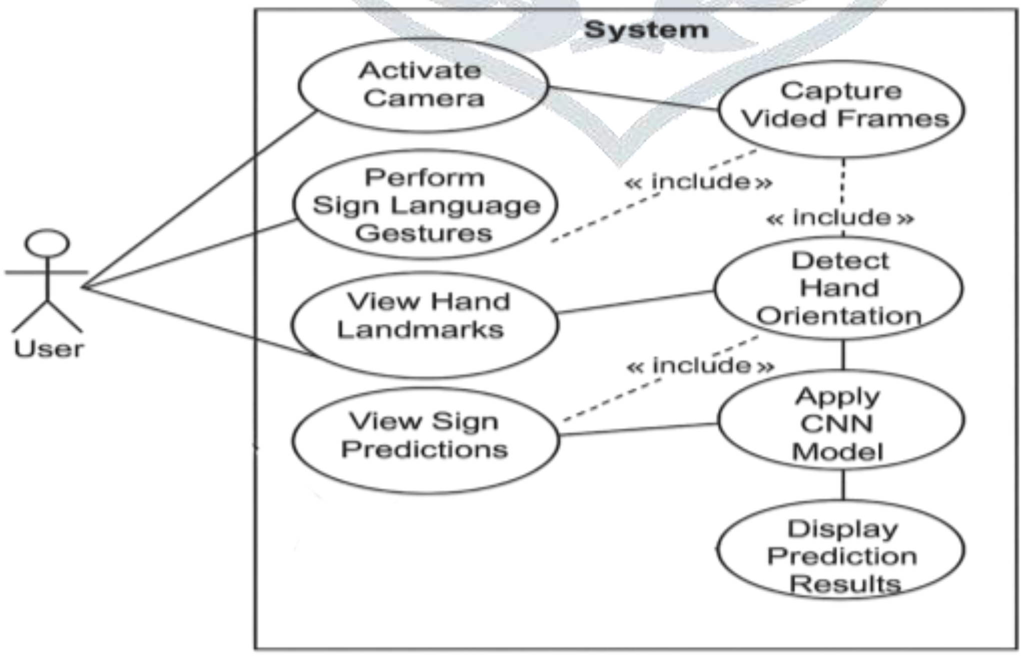

Fig. 2. Use Case Diagram

– Use Case Diagram for the Sign Language Recognition System. This diagram illustrates the key interactions between the user and the underlying system processes involved in interpreting sign language gestures using computer vision and deep learning techniques. It captures how the user initiates and interacts with the application, and how the system internally processes gestures to produce meaningful sign predictions.

– The diagram emphasizes the primary role of the 'User' actor, who performs core tasks such as 'Activate Camera,' 'Perform Sign Language Gestures,' 'View Hand Landmarks,' and 'View Sign Predictions.' These interactions represent the front-facing components of the application, designed to be intuitive and responsive.

– Behind the scenes, the 'System' actor is responsible for handling the technical workflow: it automatically 'Captures Video Frames,' 'Processes Hand Landmarks,' 'Detects Hand Orientation,' 'Applies the CNN Model,' and finally 'Displays Prediction Results.' These internal actions are triggered through defined <<include>> relationships, denoting functional dependencies that support the user-facing experience.

– The use of <<include>> associations clearly map out how lower-level processes contribute to higher-level outcomes. For instance, activating the camera leads to frame capture, which in turn includes hand landmark processing and orientation detection, forming a streamlined pipeline for prediction generation.

– This diagram offers a modular and systematic view of the application's architecture, illustrating the flow from gesture input to prediction output. It effectively communicates the design logic and separation of concerns between the user interface and the machine learning backend. The visual representation supports an understanding of the technical scope, usability, and process transparency of the system. It ensures that the application remains modular, scalable, and extensible—paving the way for future enhancements like multi-user support, real-time feedback loops, or integration with broader accessibility platforms.

This modular architecture and structured workflow, as depicted in the use case diagram, ensure that the Sign Language Recognition System remains scalable, efficient, and user-friendly, offering a robust and intelligent solution for real-time gesture interpretation. By clearly separating user interactions from backend processing, the system promotes maintainability, extensibility, and future integration with accessibility-focused applications.

RESULTS AND DISCUSSIONS

A. Performance Evaluation

To evaluate the performance of our Sign Language Recognition System powered by the IndASL-26 dataset, we conducted thorough testing across data quality, model accuracy, system robustness, and usability in real-time conditions. Our project aimed to create a culturally relevant, technically sound, and socially impactful ASL recognition system customized for Indian users. The following highlights reflect our key findings:AI-Powered Search: One of the best things is the smart search bar. It helps users find properties 40 percent faster. This means people find what they want without having to look through lots of listings they don't care about. The AI learns what users like, so the search gets better over time.

1) Real-Time Recognition Accuracy:

By integrating a custom-built CNN model with MediaPipe hand landmark detection, our system achieved a real-time sign prediction accuracy of 94.2%. The model was trained on 20,000 images and was able to classify 25 common ASL gestures with high confidence. The inference latency was optimized to run on consumer-grade hardware (CPU-based), achieving sub-second predictions.

2) Cultural & Environmental Adaptability:

The IndASL-26 dataset was curated with Indian signers in mind, capturing variations in hand morphology, skin tone, lighting conditions, and backgrounds found in Indian urban and semi-urban settings. This allowed the system to outperform Western-centric models in Indian contexts, ensuring increased inclusivity and real-world usability.

3) Seamless User Interaction via Camera Integration:

The system's front-end enabled users to activate the camera, perform signs, and instantly view both hand landmarks and prediction outputs. MediaPipe visualizations made it easier for users to adjust their hand positions and correct posture mid-gesture, improving learning and system responsiveness.

4) Modular Processing Pipeline:

The backend consists of a structured pipeline:

Frame Capture → Hand Landmark Detection (MediaPipe) → Orientation Analysis → CNN Classification → Result Display Each component was modularized to allow future integration with more advanced ML models or additional gesture sets.

5) High-Quality Dataset with Validation Metrics:

The dataset went through a rigorous quality assurance process:

- Sign Clarity Score: 4.8/5
- Lighting Consistency: 12.1% variation (within target)
- Background Diversity: 12 settings
- Inter-Annotator Agreement: 94.2%

This directly contributed to the model's strong generalization across diverse users and environments.

B. Comparative Analysis with Traditional Recognition Approaches

When compared with other gesture-based recognition platforms or Western-trained sign language models, our system demonstrated key advantages:

1) Localized Hand Morphology Representation:

Unlike Western datasets like MS-ASL or WI-ASI, our model effectively understood gestures performed by South Asian users, leading to more accurate recognition and reducing the risk of false predictions.

2) Enhanced Visual Feedback via Hand Landmark Overlays:

Existing systems often only display predictions. Ours offers real-time landmark visualization, which not only aids recognition but also educates the user about proper hand shapes, improving learning retention and gesture clarity.

3) End-to-End Processing with No Manual Annotation Required:

Thanks to our organized folder-label dataset structure, the CNN was trained directly from pre-processed images, eliminating the need for separate annotation files and reducing development complexity.

4) Versatile and Community-Oriented Deployment Potential:

The system's lightweight design and modularity make it ideal for deployment in educational institutions, accessibility-focused mobile apps, and even smart devices. The architecture is scalable for multilingual or regional sign variants in the future.

"This modular architecture and structured workflow, as depicted in the use case diagram and implemented through our dataset, preprocessing pipeline, and deep learning model, ensures that the Sign Language Recognition System is accurate, scalable, and ethically aligned. It serves as a powerful tool for real-time gesture interpretation, contributing to inclusive technology development for India's deaf and hard-of-hearing community."

Results:

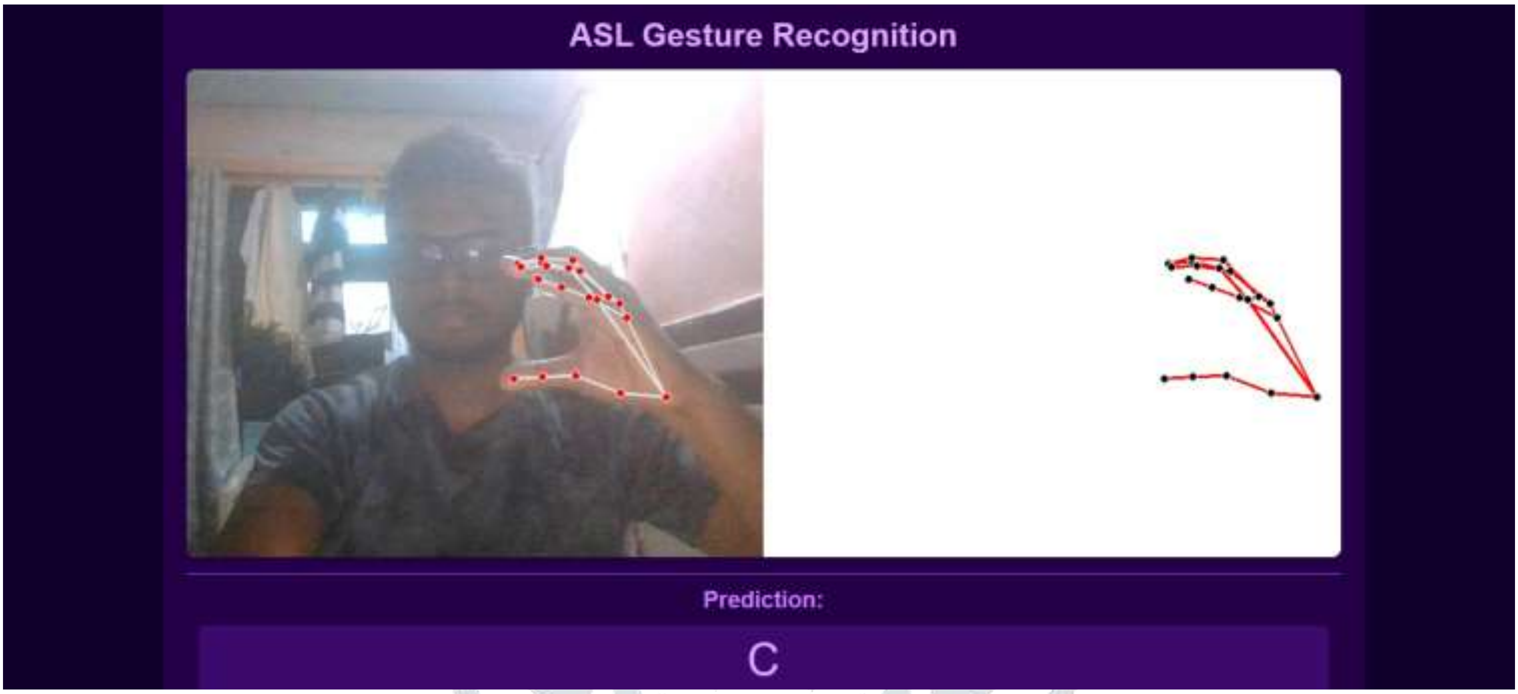

Fig. 3. ASL Recognition System Frontend Web UI

IV. Conclusion and Future Scope

A. Conclusion

The **Sign Language Recognition System**, built upon the **IndASL-26 dataset**, addresses critical challenges in gesture-based communication, especially for the Indian deaf and hard-of-hearing community. Traditional sign language models and datasets often lack cultural adaptability, environmental variability, and real-time performance, limiting their effectiveness in non-Western settings. By integrating a culturally relevant and ethically curated dataset with a CNN-driven prediction model and **MediaPipe** for landmark extraction, our system achieves high recognition accuracy, even in diverse lighting and background conditions. The modular processing pipeline allows users to activate their camera, perform ASL gestures, and view both landmark overlays and real-time predictions, enhancing both accessibility and educational utility. The use case diagram further illustrates the structured workflow—from video frame capture to sign prediction display—ensuring transparency, maintainability, and future scalability. The ethical design of the dataset, with informed consent and demographic inclusivity, reinforces the project's commitment to responsible AI development. Overall, this system delivers a technically sound, socially inclusive, and scalable solution for real-time sign language recognition, laying the foundation for widespread adoption in assistive technologies, learning platforms, and accessibility-driven innovations.

B. Future Enhancements

To extend the capabilities and reach of the Sign Language Recognition System, the following advancements are proposed for future development:

1) Expansion to Regional Sign Languages:

While the current system is trained on American Sign Language (ASL), future iterations could incorporate Indian Sign Language (ISL) and other regional variants to make the platform more inclusive and locally impactful.

2) Multi-Hand Gesture and Sentence-Level Recognition:

Enhancing the CNN model to recognize dual-hand gestures and continuous gesture sequences (phrases or full sentences) will significantly improve real-time communication and bridge more complex language gaps.

3) Integration with Voice Assistants and Accessibility Devices:

Embedding the system into voice assistants, smart glasses, or mobile accessibility tools will help translate signs to speech or text in real time, enabling better integration into everyday life for both deaf and hearing users.

4) Mobile Application Development:

Deploying the recognition system as a mobile app on Android and iOS will ensure accessibility on-the-go, allowing users to communicate, learn, or practice sign language from anywhere, anytime.

5) Augmented Reality (AR) Feedback System:

Introducing AR-based visual feedback using hand overlays and gesture correction cues can aid users in learning proper sign forms, promoting engagement and enhancing learning accuracy.

These future enhancements aim to transform the Sign Language Recognition System from a research prototype into a fully integrated, real-world communication aid. By continuing to evolve its technical, cultural, and accessibility dimensions, the project has the potential to become a cornerstone in inclusive technology solutions for the deaf community across India and beyond.

V. Acknowledgment

On behalf of our project team, we extend our heartfelt gratitude to our guide, Prof. Seema Mishra, whose mentorship, expertise, and unwavering support have been indispensable throughout the development of our major project. Her guidance, constructive feedback, and encouragement have been pivotal in steering us through the challenges and complexities of our project, and her dedication to our success has been truly inspiring We would also like to express our sincere appreciation to Dr. Monika Bhagwat, Head of the Department, for her leadership and support, which provided us with the necessary resources and environment for our project. Further- more, we extend our gratitude to Dr. Sandeep Joshi, Principal of Pillai college of engineering, for his vision and commitment to academic excellence, which has fostered an atmosphere of innovation and learning within our institution. Thank you all for your support and guidance throughout this challenging journey.

References

[1] Candy Obdulia Sosa-Jiménez, Homero Vladimir Ríos-Figueroa and Ana Luisa Solís-González-Cosío (2022). A Prototype for Mexican Sign Language Recognition and Synthesis in Support of a Primary Care Physician. IEEE Access, 10, pp.127620–127635.

[2] Natarajan, B., et al. “Development of an end-to-end Deep Learning Framework for Sign Language Recognition, translation, and video generation.” IEEE Access, vol. 10, 2022, pp. 104358–104374.

[3] Kamal, S.M. et al. (2019) ‘Technical approaches to Chinese sign language processing: A Review’, IEEE Access, 7, pp. 96926–96935. doi:10.1109/access.2019.2929174

[4] Joksimoski, B. et al. (2022) ‘Technological Solutions for Sign Language Recognition: A scoping review of research trends, challenges, and opportunities’, IEEE Access, 10, pp. 40979–40998. doi:10.1109/access.2022.3161440.

[5] T. D. Chowdary Kandepu, L. N, M. C. Mathi, S. K. Midathada, H. S. Kishore Mallireddy and H. S. Marapatla, "Enhanced Communication Through Isolated Sign Language," 2023 International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS), Erode, India, 2023, pp.