International Journal of Multidisciplinary Engineering in Current Research - IJMEC
Volume 10, Issue 1, January-2025, http://ijmec.com/, ISSN: 2456-4265

SIGN LANGUAGE TO MULTILINGUAL SPEECH

[1]Amit G Bhat, [2]Mobesh D, [3]Muhurth S Reddy, [4]Yuvaraj G, [5]Mrs Rajeshwari

[1234]Ug Students, Dept. Of ISE , Dr. Ambedkar Institute Of Technology Bangalore

[5]Asst. Professor, Dept. Of ISE, Dr. Ambedkar Institute Of Technology, Bangalore

ABSTRACT

Communication barriers faced by individuals who rely on sign language often limit their ability to interact with a broader audience. This project, Sign Language to Speech Conversion using Machine Learning, addresses this challenge by developing a system capable of recognizing American Sign Language (ASL) gestures and translating them into spoken language in real time.

The system utilizes a custom dataset comprising gestures for all 26 alphabets (A-Z), 10 digits (0-9), and special gestures for space and full stop, enabling the formation of words and sentences. Key hand landmarks are detected using MediaPipe, and the extracted features are used to train a machine learning model. Among various algorithms tested, the Random Forest Classifier demonstrated high accuracy and robustness, making it the algorithm of choice for this project.

To enhance system usability, a graphical user interface (GUI) was developed to display the recognized text while a text-to-speech (TTS) engine converts it into audible output. The real-time recognition process is further stabilized using buffer techniques to minimize misclassification. Preprocessing techniques such as feature normalization and hyperparameter optimization were employed to improve model performance.

The system's performance was evaluated using metrics like accuracy, precision, and recall, achieving reliable recognition rates for both static gestures and transitions between them. By bridging the gap between sign language users and non-sign language users, this project highlights the potential. of machine learning in improving accessibility and inclusivity. Future enhancements will include support for dynamic gestures, integration of additional sign languages, and deployment on mobile platforms, expanding the reach and impact of this innovative solution.

1- INTRODUCTION

Machine Learning (ML) is a subset of artificial intelligence (AI) that focuses on developing algorithms and statistical models that enable computers to perform specific tasks without explicit instructions. Instead, these systems learn from data patterns and make decisions or predictions based on that data. ML has become integral in various domains, including healthcare, finance, marketing, and more, due to its ability to handle vast amounts of data and uncover insights that are often beyond human capabilities.

APPLICATIONS OF MACHINE LEARNING

Machine Learning (ML) has made significant strides across various industries, transforming how tasks are performed and decisions are made. Its ability to analyse large datasets, uncover patterns, and make predictions has enabled innovative solutions in numerous fields. Here, we explore some of the prominent applications of ML.

PROBLEM STATEMENT

Sign language is crucial for communication among individuals with hearing impairments, but there is a gap in communication between sign language users and non-sign language speakers. Traditional translation methods, like human interpreters, can be costly and are not always accessible.

Machine learning (ML) offers a solution by enabling real-time recognition of sign language gestures and converting them into spoken words. This project aims to use ML to build a system that recognizes American Sign Language (ASL) gestures and translates them into speech, enhancing communication accessibility for those who rely on sign language.

Introduction to Sign Language to Speech Conversion Using Machine Learning

Sign language serves as an essential mode of communication for individuals with hearing and speech impairments. However, communication between sign language users and non-sign language speakers often presents challenges. This project, Sign Language to Speech Conversion using Machine Learning, aims to address these challenges by converting American Sign Language (ASL) gestures into spoken words or sentences.

The system works by capturing hand gestures in real-time using a webcam. These gestures, which represent letters, numbers, and special symbols (such as space and full stop), are then processed by a machine learning model trained to recognize the corresponding ASL gestures. The model uses a Random Forest Classifier to predict the correct sign based on hand landmarks captured during gesture recognition.

To build this system, a custom dataset of ASL gestures was created by capturing various hand signs in different conditions (angles, lighting, etc.). The captured gestures are processed using MediaPipe to extract key hand landmarks, converting these into a feature vector. The dataset is then used to train the machine learning model.

Once trained, the model is capable of recognizing hand gestures in real-time and converting them into text. A Text-to-Speech engine then converts the recognized text into audible speech, making the system accessible to a wider audience. This project not only improves communication for sign language users but also provides an efficient, scalable solution for real-time sign language interpretation.

PROPOSED SYSTEM

Our proposed system for sign language to speech conversion uses advanced machine learning techniques to recognize American Sign Language (ASL) gestures and convert them into spoken words or sentences. This system enhances existing models by improving gesture recognition accuracy and providing a seamless, real-time conversion experience. The system offers a user-friendly interface, robust data processing, and real-time performance for effective communication between sign language users and non-signers.

2- LITERATURE SURVEY

A literature survey is crucial in the early stages of software development. It helps assess the strengths, economics, and timelines before building a tool. Once these factors are considered, decisions on the operating systems and programming languages are made. External assistance, in the form of books, journals, and online resources, plays an important role during the development phase. These factors are considered when designing the proposal for the proposed system.

The field of sign language recognition has gained attention in recent years, particularly in the area of converting gestures to speech. Various machine learning techniques have been explored to improve gesture recognition accuracy. This literature survey highlights significant studies, their findings, and identifies gaps in the research.

Starner et al. (1998): Introduced a vision-based system for recognizing American Sign Language (ASL) signs. Their work utilized hidden Markov models (HMM) for sign recognition, contributing to the foundation of real-time sign language recognition systems.

● **Liu et al. (2018)**: Proposed a deep learning model for hand gesture recognition, utilizing Convolutional Neural Networks (CNNs) to identify ASL letters. Their study demonstrated high accuracy in classifying individual hand gestures but lacked real-time conversion to speech.

● **Ding et al. (2020)**: Conducted a study using a combination of pose estimation and machine learning models to recognize sign language gestures. They used MediaPipe for hand tracking, which allowed for a more efficient system compared to earlier gesture-based approaches.

Key findings across these studies include:

● **Hand Gesture Recognition**: Hand gesture recognition remains a crucial component, with models like CNNs and HMM showing promising results in classifying ASL gestures.

● **Real-time Conversion**: While many systems focus on gesture recognition, integrating these models with real-time speech output remains a challenge.

● **Technological Integration**: Modern studies, such as Ding et al.'s work, highlight the importance of integrating advanced tools like MediaPipe and pose estimation for improved recognition.

However, gaps exist in:

● **Real-time Conversion to Speech**: Few studies address the gap between sign language recognition and speech conversion, which is critical for communication.

● **Diversity in Dataset**: Most models are trained on limited datasets, which may not account for variations in hand shapes, sizes, or regional dialects within sign languages.

Our research aims to bridge these gaps by integrating hand gesture recognition with real-time speech conversion. We plan to explore deep learning models to improve accuracy and leverage tools like MediaPipe for better hand tracking. Collaborating with sign language experts will also ensure the model is applicable in real-world settings, addressing the needs of diverse users.

3- HARDWARE COMPONENTS

Computing Power

● **Minimum:** A computer with a multi-core processor (e.g., Intel Core i5 or AMD Ryzen 5) for handling moderate-sized datasets and model training.

● **Recommended:** A computer with a powerful multi-core processor (e.g., Intel Core i7 or AMD Ryzen 7) or higher, capable of handling large datasets and intensive computational tasks.

-Memory (RAM):

● **Minimum:** 8 GB RAM for basic model training and evaluation.

● **Recommended:** 16 GB RAM or higher for handling larger datasets and more complex machine learning algorithms.

- Storage:

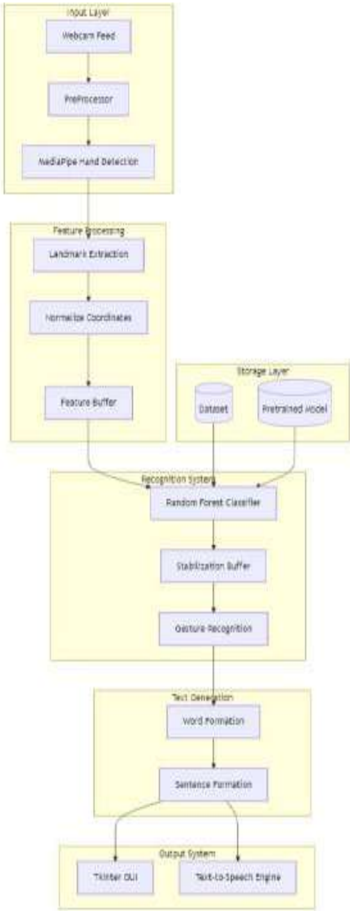
● **Minimum:** 256 GB SSD for storing datasets, software, and model files.

● **Recommended:** 512 GB SSD or higher for faster data access and storage capacity

4- SYSTEM DESIGN

The system architecture defines how different components of the sign language to speech conversion system interact with each other. The key components include data storage, preprocessing, machine learning model, and the user interface.

International Journal of Multidisciplinary Engineering in Current Research - IJMEC
Volume 10, Issue 1, January-2025, http://ijmec.com/, ISSN: 2456-4265



-Fig 1: sign language to speech conversion system architecture-

Architecture Overview:

1. High-Level Architecture

The system consists of three main components:

1. **Input Module**: Captures real-time video feed from the webcam.

2. **Processing Module**: Extracts features, predicts gestures, and forms text output.

3. **Output Module**: Displays recognized text in a GUI and converts it into speech.

5- IMPLEMENTATION

1. Data Collection and Preprocessing

1. **Data Collection**

o Use a webcam to capture gesture images for creating a custom dataset.

o Collect data for **38 classes**: 26 alphabets (A-Z), 10 digits (0-9), **space**, and **full stop** gestures.

o Ensure diverse scenarios (lighting, angles, hand positions) to improve model generalizability.

o Use the collectImgs.py script for capturing images:

o Save images in class-specific folders (e.g., data/A, data/B).

2. Data Cleaning

o Remove blurry, poorly lit, or incorrectly labeled images.

o Automate detection of outliers by checking image dimensions and pixel distributions.

3. Feature Extraction

o Use **MediaPipe** to extract 21 hand landmarks from each image.

o Convert landmarks into normalized (x, y) coordinates.

o Generate a feature vector of 42 dimensions (21 x 2) per sample.

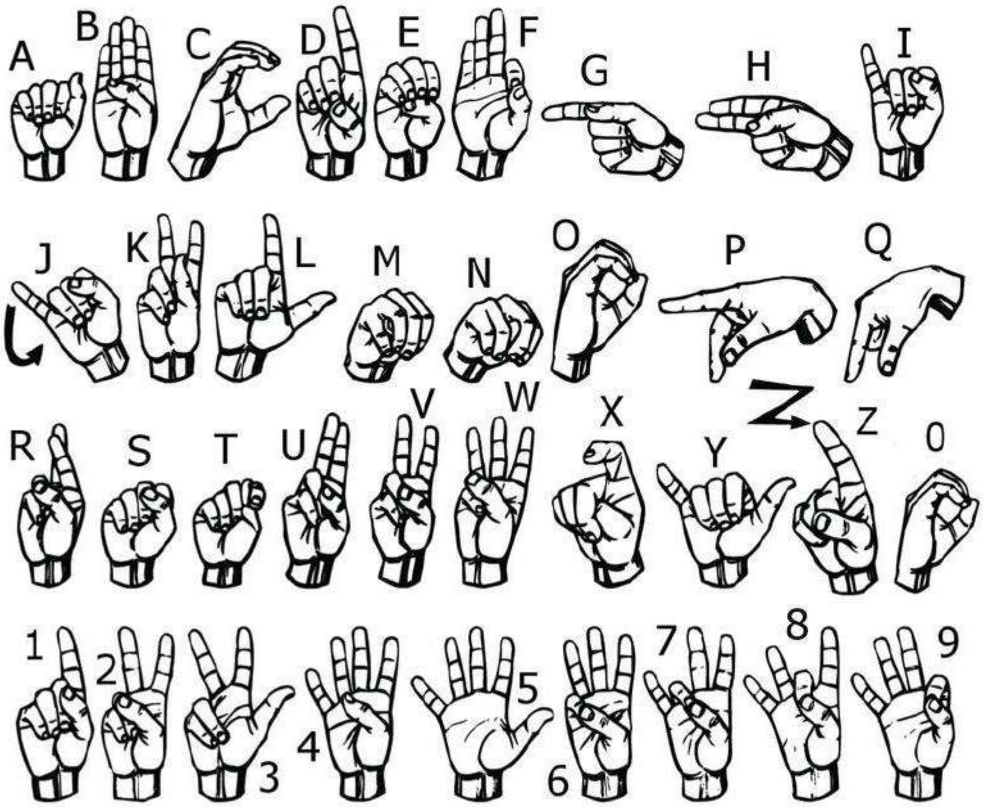


Fig 2: American Sign language symbol-

SYSTEM TESTING

Testing is the process of evaluating a system or its component(s) with the intent to find whether it satisfies the specified requirements or not. Testing is executing a system in order to identify any gaps, errors, or missing requirements in contrary to the actual requirements.

Testing Principle Before applying methods to design effective test cases, a software engineer must understand the basic principle that guides software testing. All the tests should be traceable to customer requirements.

Sl # Test Case	**Name of Test**	**Items being tested**	**Sample Input**	**Expected Output**	**Actual Output**	**Remarks**
UTC-1	Hand Detection	Camera Stream	Hand visible in frame	Should detect and highlight hand in the input stream	Hand detected successfully	Pass

UTC-2	Gesture Classification	Detected Hand Data	Symbol for L alphabet	Should classify gesture correctly as L alphabet	Gesture classified correctly	Pass
UTC-3	Background Noise Handling	Camera Stream	Hand with cluttered background	Should detect hand andignore background noise	Hand detected successfully	Pass
UTC-4	Real-Time Processing	Camera Stream	Continuous gestures	Should classify gestures in real-time without delay	Real-time classification achieved	Pass

Integration Testing for Hand Gesture Recognition

Sl # Test Case	Name of Test	Items Being Tested	Sample Input	Expected Output	Actual Output	Remarks
ITC- 1	Hand Gesture Recognition and Classification	Hand Detection and Gesture Classification	Video Stream or Image	Hand gestures should be detected and classified	Gestures detected and classified correctly	Pass

ITC- 2	Real-Time Gesture Recognition	Real-Time Processing	Continuous gestures in real-time video	Gestures should be classified in real-time without delay	Real-time recognition achieved	Pass
ITC- 3	Background Noise Handling	Gesture Detection with Background	Hands with cluttered background	Hand gestures should be accurately detected and classified despite background noise	Detection successful	Pass
ITC- 4	Gesture Detection Under Poor Lighting	Video Stream with Low Lighting	Hand in low-light conditions	Hand gesture should be detected even under low-light conditions	Detection successful	Pass
ITC- 5	Partial Hand Visibility	Hand Detection	Partially visible hand	Partially visible hand should still be detected and classified correctly	Detection and classification successful	Pass
ITC- 6	Gesture Consistency Across Frames	Gesture Detection	Same gesture in multiple frames	The system should consistently classify the same gesture across frames	Consistent classification	Pass

ITC- 7	Conversion of each letter to words and sentences	Hand Gestures	Recognized Series of gesture outputting “hello world”	The system should consistently recognize the letters and form the words and sentences correctly	Command executed successfully	Pass
ITC- 8	Conversion of sentences from English to user chosen language	Gestures and UI commands	Recognized Series of gesture outputting “hello world”	Converts “hello world” in English to hindi correctly	Conversion executed successfully	Pass

6- RESULTS

Fig.2 Example of the application

Fig.3 Example of the application

7- CONCLUSION

The **Sign Language to Speech Conversion System** bridges the communication gap between individuals who use American Sign Language (ASL) and those who do not. By leveraging advancements in machine learning and computer vision, this project successfully recognizes ASL gestures and translates them into audible speech in real time.

The project demonstrates the feasibility of using lightweight machine learning models such as the Random Forest Classifier, combined with tools like MediaPipe and OpenCV, to achieve robust and accurate gesture recognition. The addition of text-to-speech (TTS) conversion further enhances accessibility, making the system practical for real-world use.

The development process encompassed building a custom dataset, implementing a reliable feature extraction pipeline, training a high-performing model, and integrating these components into a user-friendly interface. The system's modularity also ensures that it can be customized or expanded with minimal effort.

This project highlights how machine learning and artificial intelligence can be applied to tackle communication barriers, opening up new opportunities for inclusivity and accessibility in diverse settings, such as education, healthcare, and customer service.

FUTURE SCOPE

1. **Dynamic Gesture Recognition:**

o Extend the system to recognize dynamic gestures, such as "hello" or "thank you," which involve hand movement and multiple frames of input.

o Use deep learning techniques like Recurrent Neural Networks (RNNs) or Transformer-based architectures for temporal gesture analysis.

2. **Support for Additional Sign Languages:**

o Expand the system to support other sign languages, such as British Sign Language (BSL) or Indian Sign Language (ISL).

o Create datasets for these languages to accommodate global users.

3. **Enhanced Model Accuracy:**

o Incorporate advanced deep learning models like Convolutional Neural Networks (CNNs) or Vision Transformers for gesture recognition.

o Explore transfer learning using pre-trained models to reduce training time and improve accuracy.

4. **Real-Time Performance Optimization:**

o Optimize the real-time processing pipeline to reduce latency, enabling smoother gesture recognition and faster TTS response.

o Use tools like TensorFlow Lite for deploying the system on resource-constrained devices.

5. Mobile and Web Deployment:

o Develop a mobile application using Flutter or React Native to make the system portable.

o Create a web-based version using Flask or FastAPI for greater accessibility.

REFERENCES

[1] Tom M. Mitchell, *"Machine Learning"*: This book provides an in-depth introduction to machine learning principles, techniques, and algorithms, emphasizing foundational concepts and their practical applications.

[2] Yashwant Kanetkar, *"Let Us Python"*: This book serves as a comprehensive guide to Python programming, offering beginner- friendly explanations and practical examples for mastering Python basics and advanced features.

[3] Chung, D., & Do, S. (2020), *"Sign Language Recognition and Translation: A Survey"*: This paper presents a detailed survey of various methodologies, challenges, and

ISSN: 2456-4265
IJMEC 2025

advancements in the field of sign language recognition and translation systems.

[4] Cheng, X., & Liu, J. (2021), *"A Survey of Machine Learning for Sign Language Recognition"*: This article reviews machine learning approaches for sign language recognition, highlighting key trends, innovations, and challenges in the field.

[5] Zhao, X., & Zhang, D. (2017), *"Random Forests for Gesture Recognition: Applications to Sign Language"*: This study explores the use of Random Forest classifiers for gesture recognition, focusing on their application to sign language and demonstrating their effectiveness in real-world scenarios.