International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Real-Time Sign Language Interpreter

Shraddha Kosare[1], Sakshi Kamdi[2], Payal Jibhakate[3], Subodini Gote[4], Tanushri Aote[5], Vaishnavi Wagh[6], Prof. Sabyasachi Bhattacharya[7]

[1,2,3,4,5,6,7]Final Year Student, Department of Electronics & Telecommunication Engineering, Cummins College of Engineering for Women, Nagpur

Abstract

This paper presents a Real-Time Sign Language Interpreter that bridges the communication gap between hearing-impaired individuals and those without knowledge of sign language. The system integrates an ESP32-CAM for image capture, a pre-trained and quantized MobileNet model for real-time gesture recognition, and text-to-speech and speech-to-text conversion for bidirectional communication. It supports English, Hindi, and Marathi, providing a multilingual, customizable, and accessible communication interface. The system is portable, cost-effective, and does not require wearable equipment, making it ideal for various public and private sector applications.

Keywords: Sign Language Recognition, MobileNet, ESP32-CAM, Real-Time Processing, Multilingual Communication

1. Introduction

Communication is a fundamental human need. For individuals with hearing and speech impairments, interacting with non-sign language users often poses challenges, limiting access to education, employment, and essential services. Traditional methods like human interpreters and text-based apps are either unavailable or lack natural interaction. Wearable sensor-based alternatives are often costly and impractical. To address these limitations, this paper presents a non-invasive, real-time sign language interpreter using ESP32-CAM, MobileNet, and multilingual text-to-speech and speech-to-text systems.

The proposed system provides dynamic gesture recognition, customizable sign storage, and bilingual communication support in English, Hindi, and Marathi. Using a lightweight MobileNet model optimized for embedded devices, the solution ensures fast and accurate interpretation of gestures. It is cost-effective, portable, and does not require wearables, making it ideal for real-world applications in education, public services, and healthcare environments.

2. Literature Review

Recent advancements in deep learning have significantly improved the accuracy of vision-based sign language recognition systems. Multiple studies have utilized convolutional neural networks and transfer learning to develop robust recognition models. For instance, MobileNetV2 has been widely adopted for its efficiency in edge computing environments [1][2]. Custom datasets have also shown promise in adapting systems to regional languages like Indian Sign Language (ISL) [3][4].

Other approaches include wearable technologies like smart gloves equipped with sensors [5], though these often face limitations in terms of affordability and comfort. Studies integrating ESP32-CAM modules

have demonstrated potential for low-cost, real-time processing [6]. However, most systems lack support for multilingual output and customizable sign storage.

3. Proposed Methodology

The methodology consists of five primary stages: (1) image acquisition using ESP32-CAM, (2) gesture classification via MobileNet, (3) display and speech synthesis of recognized signs, (4) two-way communication using speech-to-text, and (5) dynamic gesture storage for personalized use. The system architecture allows users to form sentences by storing and retrieving custom signs, while also converting spoken responses into text on an OLED screen. The methodology comprises data acquisition**,** model training and optimization**,** real-time inference**,** dynamic sign storage**,** and two-way communication**.**

3.1 System Architecture Overview

The system is divided into the following modules:

Table 3.1.1 System Module Descriptions

Module	**Description**
Gesture Input Module	Captures real-time sign gestures using ESP32-CAM.
Image Preprocessing Module	Applies resizing, noise reduction, and normalization.
Gesture Recognition Module	Uses MobileNet CNN model for classification.
Output Display Module	Displays recognized gestures on an LCD screen.
Text-to-Speech Module	Converts recognized text into audible speech using a 5V DC speaker.
Speech-to-Text Module	Converts speech from non-signers into text for visual output.
Custom Gesture Management	Allows users to add, store, edit, and delete signs.
Firebase/Cloud Integration	Stores user-defined gestures and supports syncing across sessions.

3.2 Block Diagram

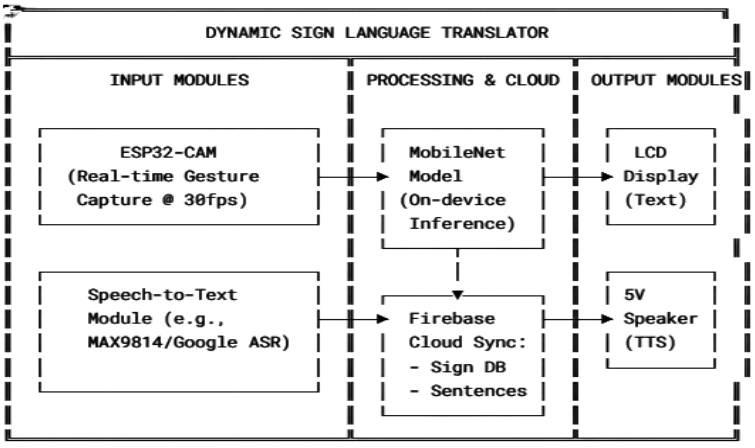


Figure 3.2.1 Block Diagram

ESP32-CAM

- This module serves as the primary image capturing unit.
- It continuously captures hand gestures in real-time from the user.
- Captured images are passed into the next stage for preparation before classification.

Preprocessing Layer

- This layer applies essential image processing techniques such as:
  - Resizing the image to match the input dimensions required by the model (e.g., 224×224).
  - Denoising to reduce visual clutter and enhance the features of hand gestures.
  - Normalization to scale pixel values, ensuring consistency and improving model performance.
- The processed image is now clean and ready for gesture classification.

MobileNet Gesture Classifier

- A lightweight deep learning model (MobileNet) trained on sign language gestures.
- It analyzes the preprocessed image and predicts the most likely gesture class (e.g., "Hello", "Thank You").

Text Output → OLED Display]

- The predicted gesture is converted into textual output.
- This text is then shown on the OLED screen, making it visible to both the deaf/mute user and others.

Text-to-Speech → Speaker Output

- The same textual output is passed to a Text-to-Speech (TTS) module.
- The system speaks out the translated gesture using a 5V DC speaker, aiding communication with non-signers.

3.3 Data Flow Pipeline

Step	Action	Tool/Component Used
1	Capture image of gesture	ESP32-CAM
2	Preprocess image	Gaussian Blur, Resizing
3	Gesture classification	MobileNet (CNN)
4	Convert text to speech	TTS Module (Speaker)
5	Display text	LCD Display (16x2)
6	Capture voice input	Microphone Module
7	Convert speech to text	Google Speech API / Offline STT
8	Display converted speech text on LCD	LCD Display
9	Save new gesture	Firebase (via Flask Web)

3.4 MobileNet Model Summary

Property	Value
Model Name	MobileNetV2
Pretrained Dataset	ImageNet
Input Size	224 × 224 × 3
Output Classes	20 (custom signs)
Optimizations Applied	Quantization, Pruning
Training Epochs	

4. Data Collection

Effective sign language recognition systems require robust and diverse datasets. To improve real-time gesture classification, a custom dataset was collected using ESP32-CAM. This includes both static and dynamic hand gestures under various lighting conditions and angles to improve the model's generalization. Unlike standard datasets limited to American or British Sign Language, the custom dataset supports Indian languages and user-defined gestures. Data preprocessing included resizing, noise reduction, and augmentation. Labels and metadata were stored in structured formats to assist with supervised training.

5. Design and Implementation

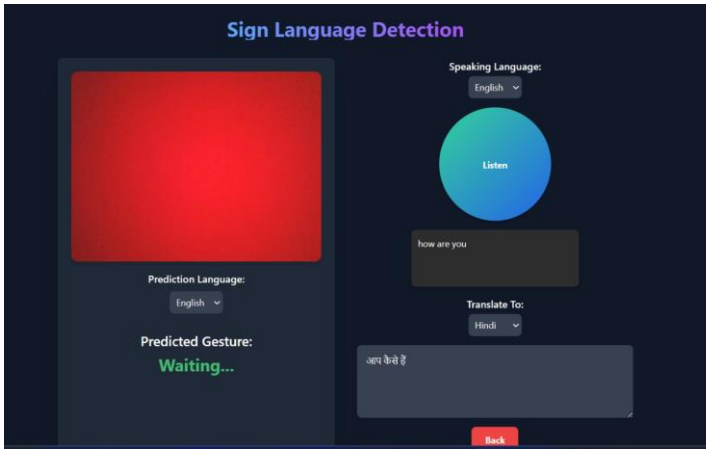
The hardware includes an ESP32-CAM module for capturing images, an LCD for text display, and a 5V DC speaker for audio output. The ESP32-CAM is interfaced with peripherals via I2C and GPIO protocols. Firmware and model deployment are handled using the Arduino IDE, while the deep learning model is trained using Python in PyCharm.

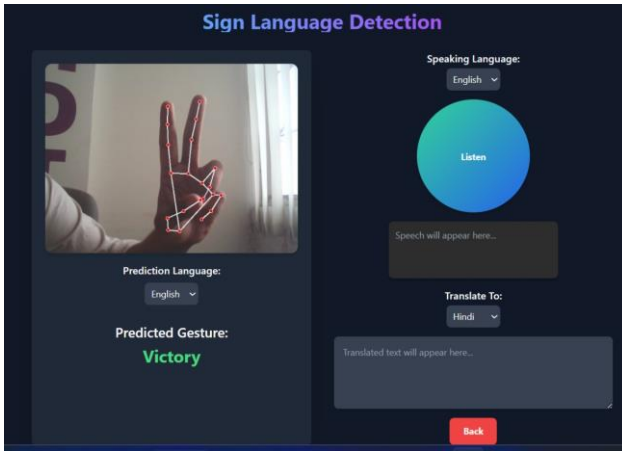
The software pipeline involves capturing and preprocessing images, classifying gestures using a MobileNet model, and outputting the result via LCD and speaker. The model is quantized and optimized for low-power inference. Firebase is optionally used for storing user-defined gestures and syncing data. Flask provides a basic web interface for interaction and monitoring.

6. Testing and Results

The system was evaluated for recognition accuracy, response time, and robustness in varying environments. Accuracy reached 90–95% under standard lighting. Speech-to-text transcription accuracy was approximately 93%, while text-to-speech output achieved 98% clarity.

Real-time processing delay was under 2 seconds, making it suitable for live communication. Custom gestures were successfully stored and retrieved, demonstrating the system’s flexibility and adaptability.





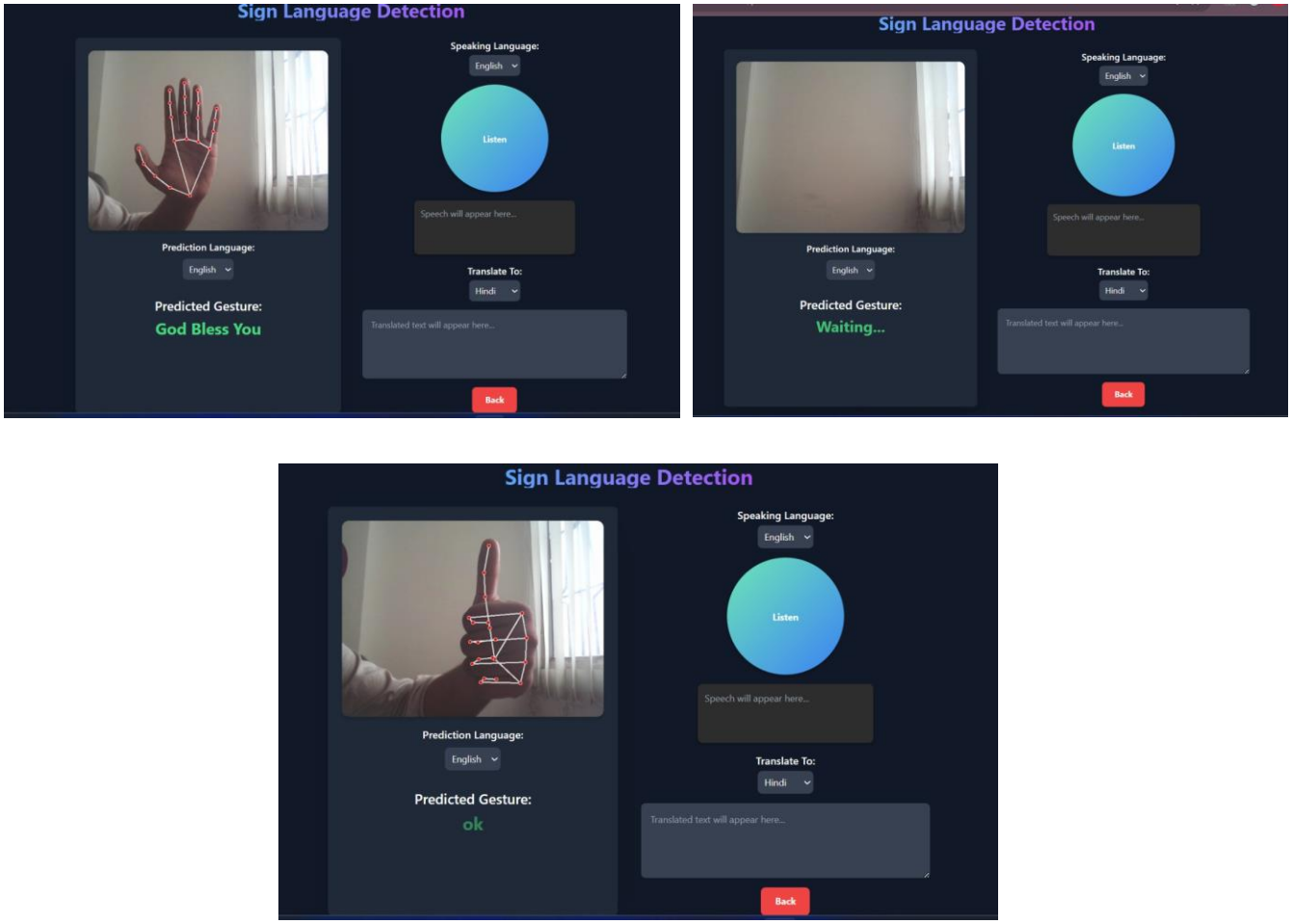


Figure 6.1 Sign Language Predictor Output Images

7. Conclusion and Future Scope

This research presents an accessible, cost-effective Real-Time Sign Language Interpreter using deep learning and embedded vision. Integrating dynamic sign storage and bilingual support ensures that the solution is adaptable and scalable for diverse environments.

Future work includes expanding the gesture vocabulary, enhancing dynamic gesture recognition, incorporating edge-AI accelerators for speed, and developing a mobile companion app for wider usability.

8. References

1. T. Starner and A. Pentland, “Real-time American Sign Language recognition from video using hidden Markov models,” *Proc. Int. Symp. Comput. Vis.*, 1995, pp. 265–270.
2. J. Wu, M. Zhang, and X. Wang, “A Vision-Based Real-Time Sign Language Recognition System Using CNN and LSTM,” *IEEE Trans. Multimedia*, vol. 24, pp. 270–282, Jan. 2022.
3. A. V. Nandhini, G. Muthukumaran, and R. Sivakumar, “Deep Learning-Based Sign Language Recognition: A Review,” *IEEE Access*, vol. 9, pp. 124678–124695, 2021.
4. G. K. Verma and D. G. Gaur, “Hand Gesture Recognition Using CNN for Indian Sign Language,” *2021 Int. Conf. Signal Process., Comput. Control (ISPCC)*, Solan, India, 2021.
5. A. B. Raut and S. S. Talbar, “Real-Time Sign Language Recognition Using Deep Convolutional Networks,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 8, pp. 3562–3572, Aug. 2022.

6. M. Pigou, S. Dieleman, P.-J. Kindermans, and B. Schrauwen, "Sign Language Recognition Using Convolutional Neural Networks," *Eur. Conf. Comput. Vis. (ECCV)*, Zurich, Switzerland, 2014.
7. K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 6, pp. 1228–1239, Jun. 2017.
8. H. Cooper, B. Holt, and R. Bowden, "Sign Language Recognition," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 82–87, Nov. 2012.
9. A. L. Ko and Y. H. Kim, "Hand Gesture Recognition Using Sensor Fusion for Real-Time Sign Language Translation," *IEEE Sens. J.*, vol. 21, no. 12, pp. 14028–14036, Jun. 2021.
10. N. A. Badshah, I. Ullah, M. I. Khan, and S. A. Khan, "Real-Time Pakistani Sign Language Recognition Using Deep Learning," *2022 Int. Conf. Emerg. Trends Electr. Electron. Sustain. Energy Syst. (ICETEESES)*, Bannu, Pakistan.
11. S. F. Hussain, A. Nasir, and F. Zafar, "Application of Deep Learning to Sign Language Recognition Using MobileNet," *2021 Int. Conf. Mach. Learn. Technol. (ICMLT)*, pp. 63–68.
12. S. Ghosh and A. K. Roy, "A Lightweight CNN Model for Real-Time Indian Sign Language Gesture Recognition," *2020 IEEE 17th India Council Int. Conf. (INDICON)*.
13. M. Zhang, Y. Liu, and Z. Yan, "Speech Recognition and Synthesis for Assistive Communication Using IoT," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 12183–12190, Dec. 2020.
14. R. Rahim et al., "Smart Glove and CNN-Based Real-Time Sign Language Recognition," *IEEE Access*, vol. 9, pp. 145698–145711, 2021.
15. Y. Hossain and M. H. Kabir, "Gesture Recognition Using ESP32-CAM for Smart Interaction," *2023 IEEE Global Humanitarian Technology Conference (GHTC)*.
16. M. A. Hossain, T. A. Islam, and R. Kabir, "ESP32-Based Smart Surveillance System Using CNN," *2022 Int. Conf. Electr. Comput. Commun. Eng. (ECCE)*.
17. V. V. L. M. R. Dinesh and A. S. Mohan, "Integration of IoT and Firebase in Real-Time Monitoring Systems," *2021 IEEE Int. Conf. Comput. Intell. Commun. Netw. (CICN)*.
18. S. K. Roy, M. S. Islam, and F. Ahmed, "IoT-Based Assistive Devices for the Hearing-Impaired: A Review," *IEEE Access*, vol. 8, pp. 150478–150493, 2020.
19. A. K. Singh and P. Bhattacharya, "A Comparative Study of CNN Models for Image-Based Sign Language Classification," *2021 IEEE Students Conf. Eng. Syst. (SCES)*.
20. S. Singh and R. K. Sharma, "Augmented Reality and AI for Sign Language Translation," *2020 IEEE Int. Conf. Comput. Intell. Virtual Environ. Soc. Dev. (CIVEMSA)*.