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

Vision-Based Real-Time Sign Language Translator Using MediaPipe and OpenCV on Raspberry Pi

Gangaraju Sai Jayanth[1], Bareddy Dinesh Reddy[2], Annem Aswini[3], P. Surya Prakash Reddy[4]

[1,2,3]Student, Electronics and Communication, G.Pulla Reddy Engineering College
[4]Assistant Professor, Electronics and Communication, G. Pulla Reddy Engineering College

Abstract
Sign language is an essential communication medium for individuals with speech and hearing impairments. However, communication barriers still exist in daily life because many people are not familiar with sign language interpretation. This paper presents a vision-based real-time sign language translator implemented on a Raspberry Pi using MediaPipe and OpenCV for low-cost and portable assistive communication. The proposed system captures live hand gestures through a USB webcam and extracts 21 hand landmarks using MediaPipe Hands. A rule-based finger-state analysis method is employed to classify 25 predefined static hand gestures corresponding to essential communication messages. The recognized message is displayed on a 16×2 I2C LCD and simultaneously converted into speech using an offline text-to-speech engine, enabling multimodal feedback. Unlike many existing systems that depend on glove-based sensors, high-end computers, or cloud connectivity, the proposed approach provides a fully offline, embedded, contactless, and practical solution suitable for real-time deployment. Experimental results under indoor conditions demonstrate reliable recognition of 25 predefined gestures with approximately 80% accuracy and a response time of 3–5 seconds. The proposed system is suitable for assistive communication in homes, hospitals, educational environments, and public spaces.

Keywords: Sign language translation, MediaPipe, OpenCV, Raspberry Pi, gesture recognition, assistive communication, embedded system, text-to-speech.

I. INTRODUCTION

Sign language plays a crucial role in enabling communication for individuals with speech and hearing impairments. Despite its importance, a significant communication gap remains because a large portion of the general population is not trained in sign language interpretation. This limitation creates difficulties in day-to-day interactions in hospitals, educational institutions, homes, workplaces, and public service environments. As a result, there is a growing need for practical systems that can automatically translate sign gestures into understandable text and speech.

Conventional sign language recognition systems are commonly based on sensor gloves, specialized depth cameras, PC/GPU-based computer vision systems, or cloud-assisted platforms. Although glove-based

systems can provide precise sensing, they are intrusive, uncomfortable, and often expensive. Similarly, high-performance PC or GPU-based systems reduce portability and increase deployment cost. Cloud-based approaches may introduce latency, dependency on internet connectivity, and privacy concerns, making them less suitable for real-time assistive applications in constrained environments.

To address these challenges, this work proposes a vision-based real-time sign language translator using MediaPipe and OpenCV on a Raspberry Pi platform. The system performs all gesture processing locally using a standard USB webcam, eliminating the need for wearable sensors, specialized cameras, or cloud infrastructure. The proposed system recognizes a predefined set of static hand gestures and maps them to commonly required communication messages. The recognized output is then presented through both a 16×2 LCD display and offline speech synthesis, thereby improving accessibility and usability.

The main contributions of this work are: (1) development of a fully offline embedded sign language translation system using Raspberry Pi; (2) implementation of rule-based gesture recognition using MediaPipe hand landmarks and finger-state analysis without requiring large training datasets; (3) simultaneous multimodal output through text display and speech synthesis; and (4) a compact and contactless design that improves portability and usability compared with glove-based and cloud-dependent systems.

The remainder of this paper is organized as follows: Section II discusses related work, Section III describes the system architecture, Section IV presents the hardware design, Section V explains the software workflow, Section VI details the gesture recognition method, Section VII discusses experimental results, Section VIII concludes the paper, and Section IX presents future scope.

II. RELATED WORK

Sign language translation systems reported in the literature can broadly be categorized into glove-based systems, vision-based systems, and cloud-assisted intelligent systems.

Glove-based systems utilize flex sensors, accelerometers, gyroscopes, or inertial sensing units to capture finger bending and hand movement [6]. These systems generally provide accurate motion sensing; however, they require the user to wear hardware devices, which can reduce comfort and naturalness of communication. In addition, the cost of sensors and maintenance of wearable components may limit practical adoption.

Vision-based systems use cameras and computer vision algorithms to recognize hand gestures without physical contact. Recent advances in image processing, machine learning, and deep learning have improved the performance of such systems [3][4][7]. However, many vision-based approaches rely on desktop computers or GPU-enabled platforms for computation, which increases power consumption and reduces portability. Some systems also require large training datasets and extensive model training.

Cloud-assisted systems perform gesture recognition using remote servers, often enabling more complex models and larger vocabulary support. Nevertheless, these systems depend on internet connectivity, may introduce communication delays, and raise concerns regarding privacy and data security [5]. Such limitations can be problematic in real-time assistive communication scenarios where immediate and reliable response is required.

Recent developments in MediaPipe Hands [1][8] have enabled efficient real-time extraction of hand landmarks on resource-constrained devices. MediaPipe provides 21 key hand landmarks that can be used for robust geometric analysis of finger positions. This has made it possible to design lightweight and accurate hand gesture recognition systems without requiring expensive sensors or deep learning inference

on powerful hardware.

Although several studies have explored gesture recognition using MediaPipe and OpenCV, many implementations focus only on gesture detection or human-computer interaction tasks. In contrast, the present work emphasizes assistive communication by integrating gesture recognition, embedded local processing, LCD text output, and offline speech generation into a single compact system. The proposed work therefore addresses the need for a low-cost, portable, fully offline, and practical real-time sign language translator.

III. SYSTEM ARCHITECTURE

The overall architecture of the proposed system is designed to support real-time gesture acquisition, processing, recognition, and output generation on an embedded platform. The system consists of the following major components: Raspberry Pi as the central processing unit, USB webcam for live hand gesture capture, MediaPipe Hands for hand landmark detection, OpenCV for video frame acquisition and preprocessing, a rule-based gesture recognition module for gesture classification, a 16×2 I2C LCD display for visual output, and speaker/audio output for speech feedback using offline text-to-speech.

The webcam continuously captures video frames of the user's hand gestures. Each frame is processed using OpenCV and passed to the MediaPipe Hands model, which extracts 21 hand landmarks. Based on the spatial relationships among these landmarks, the system determines whether each finger is open or closed. The resulting finger-state pattern is matched against a predefined gesture library. When a valid gesture is recognized, the corresponding message is displayed on the LCD and spoken through the speaker using an offline text-to-speech engine.

The proposed architecture is fully self-contained and performs all computation locally on Raspberry Pi, making it suitable for embedded assistive communication applications.

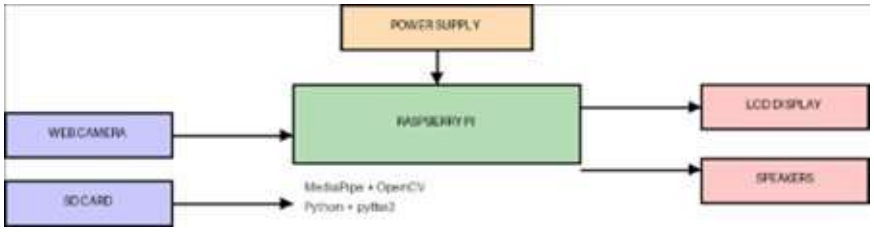


Fig. 1. Block diagram of the proposed sign language translator system.

IV. HARDWARE DESIGN

The hardware implementation of the proposed system is based on a compact and low-cost embedded platform. The major hardware components used in the prototype are: (1) Raspberry Pi board, which serves as the main processing unit for executing the gesture recognition algorithm; (2) USB webcam (USB 2.0), which captures live video frames of hand gestures; (3) 16×2 I2C LCD module (16×2 I2C), which displays the recognized message in text form; (4) speaker/audio output device connected via 3.5 mm audio jack, which provides audible feedback through text-to-speech conversion; and (5) a regulated power supply (5V/3A USB-C), which powers the Raspberry Pi and connected peripherals.

The webcam is interfaced through the Raspberry Pi USB port. The LCD module communicates with the Raspberry Pi through the I2C bus (SDA/SCL), reducing the number of required GPIO pins and simplifying hardware wiring. The SD card (Micro SD) serves as the storage medium for the operating system and software. The speaker is connected through the audio output interface for speech synthesis. This hardware

arrangement results in a portable and practical assistive device that can be deployed in real-world environments.

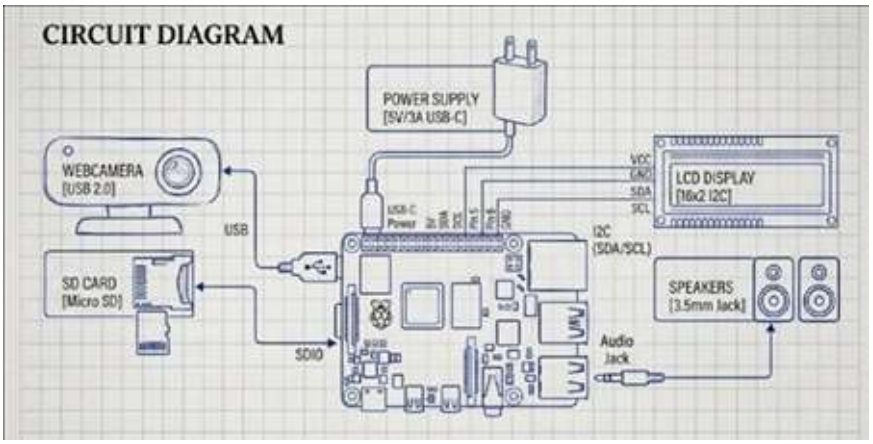


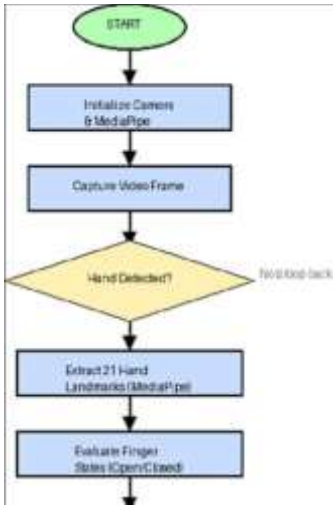
Fig. 2. Circuit diagram of the Raspberry Pi-based sign language translator system.

V. SOFTWARE WORKFLOW

The software workflow of the proposed system consists of a sequence of operations executed continuously in real time. Initially, the webcam, LCD interface, audio output module, and MediaPipe Hands model are initialized when the program starts. Live video frames are captured continuously from the USB webcam using OpenCV. Each frame is optionally resized and converted into the appropriate color format required by MediaPipe for efficient hand landmark detection.

MediaPipe Hands processes the frame and extracts 21 hand landmarks representing the wrist, finger joints, and fingertips. The positions of fingertip landmarks are compared with intermediate finger joints to determine whether each finger is extended or folded. The finger-state combination is encoded into a binary pattern and matched against a predefined gesture library containing essential communication messages. When a valid gesture is detected, the corresponding message is displayed on the 16×2 LCD and converted to speech using an offline text-to-speech engine (pyttsx3). This process repeats for each incoming frame, enabling real-time gesture translation.

The complete software workflow is illustrated in Fig. 3. The flowchart shows the sequential steps from initialization through gesture detection, landmark extraction, finger-state evaluation, gesture matching, and multimodal output generation.



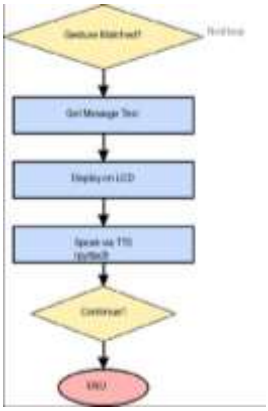


Fig. 3. Flowchart of the real-time gesture recognition and translation process.

VI. GESTURE RECOGNITION METHOD

The proposed system performs gesture recognition using MediaPipe Hands, which provides 21 hand landmarks for each detected hand. These landmarks correspond to the wrist, finger joints, and fingertips, enabling geometric analysis of finger posture.

For each detected hand, MediaPipe returns 21 landmarks represented by normalized coordinates (x_i, y_i, z_i), where i = 0, 1, 2, …, 20. These landmark positions are used to determine the state of each finger.

The recognition logic is based on whether each finger is open (1) or closed (0). For the index, middle, ring, and little fingers, a finger is considered open if the fingertip is above its corresponding proximal interphalangeal (PIP) joint in the image coordinate system (y_tip < y_PIP). Otherwise, the finger is considered closed. For the thumb, horizontal comparison is used because its orientation differs from the other fingers; depending on the detected hand orientation, the thumb can be considered open based on the relative x-coordinates of the thumb tip and thumb joint.

The state of the five fingers is represented as a binary gesture vector G = [T, I, M, R, P], where T, I, M, R, and P correspond to the thumb, index, middle, ring, and little finger states, respectively. Each element is either 0 or 1. The generated binary vector is compared against a predefined gesture library stored in the program. Each valid pattern is mapped to an essential communication message such as "I Need Food," "I Want to Go Out," "Help Me," "Need Water," or "Call Doctor."

This rule-based method is computationally efficient, easy to implement, and suitable for low-latency embedded execution. It is particularly effective for a limited vocabulary of static gestures and does not require any training datasets or machine learning model inference.

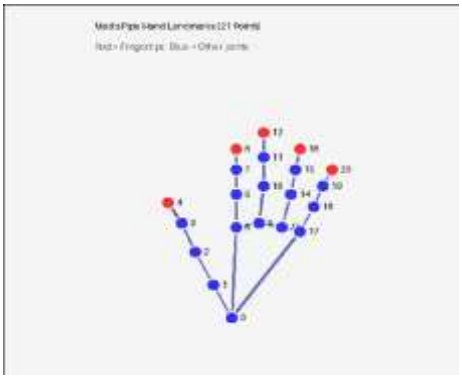


Fig. 4. MediaPipe hand landmark model showing 21 keypoints used for gesture analysis (Red = Fingertips, Blue = Other Joints).

VII. EXPERIMENTAL RESULTS AND DISCUSSION

The proposed system was implemented successfully on a Raspberry Pi platform and tested under typical indoor lighting conditions. The prototype was evaluated for real-time recognition of a predefined set of static hand gestures corresponding to essential communication messages.

A. Experimental Setup

The experimental setup consists of a Raspberry Pi embedded platform, USB webcam for live hand capture, 16×2 I2C LCD display for text output, speaker for speech synthesis, and Python implementation using OpenCV and MediaPipe. The system was tested using 25 predefined static hand gestures representing commonly used communication messages. Multiple trials were conducted in indoor conditions with the user's hand placed within the camera's field of view. The software stack used Python 3, OpenCV 4.x, MediaPipe 0.9.x, RPLCD for I2C LCD control, and pyttsx3 for offline text-to-speech synthesis.

B. Performance Evaluation

The performance of the system was evaluated based on recognition accuracy and response time. Table 1 summarizes the key performance parameters of the proposed system. The results demonstrate that the proposed system can recognize the predefined gestures with reasonable reliability while maintaining acceptable response time for real-time assistive communication.

Table 1. Performance Summary of the Proposed System

Parameter	Value / Description
Number of Predefined Gestures	25
Gesture Type	Static single-hand gestures
Processing Platform	Raspberry Pi
Camera Type	USB Webcam (USB 2.0)
Processing Mode	Fully offline / local
Output Modes	LCD text + speech (TTS)
Test Environment	Indoor lighting conditions
Recognition Accuracy	~80%
Average Response Time	3–5 seconds
Internet Dependency	None (fully offline)

C. Representative Output Results

During testing, several gestures were successfully recognized and translated into both text and speech outputs. Fig. 5 and Fig. 6 show representative LCD outputs for two recognized gestures. When a gesture is detected, the corresponding message is displayed on the LCD and simultaneously spoken through the speaker using the offline text-to-speech engine.

Fig. 5. LCD output for the recognized gesture corresponding to the message "I Want to Go Out".

Fig. 6. LCD output for the recognized gesture corresponding to the message "I Need Food".

D. Comparative Discussion

Compared with glove-based and cloud-assisted systems, the proposed approach offers several practical advantages, including contactless interaction without wearable sensors, low hardware cost, fully offline operation, portable embedded implementation, and multimodal output through text and speech. The rule-based approach also enables deployment without requiring any model training or dataset collection, which further reduces development complexity.

E. Limitations

Although the proposed system performs effectively for a limited set of predefined static gestures, certain limitations remain. Performance may degrade under poor illumination conditions, complex or cluttered backgrounds, partial hand occlusion, variations in hand orientation, and gestures with similar finger-state patterns. In addition, the current rule-based method is best suited for static gestures with limited vocabulary and may not scale efficiently to continuous sign language recognition or dynamic temporal gestures. These limitations motivate future enhancement using machine learning or sequence-based models.

VIII. CONCLUSION

This paper presented a vision-based real-time sign language translator using MediaPipe and OpenCV implemented on a Raspberry Pi platform for assistive communication. The proposed system integrates live gesture capture, hand landmark extraction, rule-based finger-state analysis, LCD text display, and

offline speech synthesis into a compact embedded solution. Unlike many conventional systems that depend on sensor gloves, desktop-class hardware, or cloud infrastructure, the proposed approach provides a low-cost, portable, contactless, and fully offline alternative suitable for real-world deployment.

The experimental implementation demonstrates that the system can reliably recognize 25 predefined static gestures and generate multimodal feedback with approximately 80% recognition accuracy and a response time of 3–5 seconds under indoor conditions. The proposed system is therefore a practical and effective solution for bridging communication gaps for individuals with speech and hearing impairments in everyday environments.

IX. FUTURE SCOPE

Future work can extend the proposed system in several directions to improve functionality, scalability, and robustness. These include expanding the gesture vocabulary, extending the system from predefined messages to alphabet-based or word-based sign translation, incorporating temporal modeling techniques for dynamic gesture recognition, integrating machine learning or deep learning models for improved generalization, enabling bidirectional communication through speech-to-text conversion, and improving performance under varying lighting conditions, complex backgrounds, and different user hand shapes.

REFERENCES

1. C. Lugaresi et al., "MediaPipe: A Framework for Building Perception Pipelines," arXiv preprint arXiv:1906.08172, 2019.
2. G. Bradski, "The OpenCV Library," Dr. Dobb's Journal of Software Tools, 2000.
3. N. C. Camgoz, S. Hadfield, O. Koller, and R. Bowden, "SubUNets: End-to-End Hand Shape and Continuous Sign Language Recognition," in Proc. IEEE ICCV, 2017, pp. 3075–3084.
4. O. Koller, H. Ney, and R. Bowden, "Deep Hand: How to Train a CNN on 1 Million Hand Images When Your Data Is Continuous and Weakly Labelled," in Proc. IEEE CVPR, 2016.
5. T. Starner, J. Weaver, and A. Pentland, "Real-Time American Sign Language Recognition Using Desk and Wearable Computer-Based Video," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 12, pp. 1371–1375, 1998.
6. S. Mitra and T. Acharya, "Gesture Recognition: A Survey," IEEE Trans. Systems, Man, and Cybernetics, Part C, vol. 37, no. 3, pp. 311–324, 2007.
7. A. Rastgoo, K. Kiani, and S. Escalera, "Video-Based Isolated Hand Sign Language Recognition Using a Deep Cascaded Model," Multimedia Tools and Applications, vol. 79, pp. 22965–22987, 2020.