Hand Gesture Recognition System with Voice Feedback Using MediaPipe and OpenCV

Raksha K[1], Prof. Bhavya P S[2], Dr. Divya A K[3], Prof. Naseema C A[4]
[1,2,3,4]*K.V.G College of Engineering, Sullia, Karnataka*

Abstract*— A real-time hand gesture recognition system with voice feedback is presented in this paper. It was created with MediaPipe and OpenCV. By converting hand gestures into audible voice commands, the system seeks to help people with speech and hearing impairments. Real-time hand landmark tracking is done with MediaPipe, and image processing and gesture recognition are done with OpenCV. The identified gesture is translated into voice output by a text-to-speech engine. The suggested approach is portable, effective, and compatible with consumer-grade hardware that has a webcam. It illustrates the possibility of creating accessible and reasonably priced assistive communication technologies.

***Index Terms*— Hand Gesture Recognition, MediaPipe, OpenCV, Text-to-Speech, Real-time System**

I.INTRODUCTION

Beyond spoken words, human communication also involves body posture, facial expressions, and gestures. Millions of people with speech or hearing impairments rely on hand gestures to express themselves. However, communication barriers arise because most members of the general public are not familiar with sign language. Real-time hand gesture detection and interpretation is now possible thanks to developments in computer vision and deep learning. In order to help close the communication gap between hearing and differently-abled people, this project presents a system that uses MediaPipe and OpenCV to recognize hand gestures and provide immediate voice feedback via a text-to-speech engine.

II.LITERATURE REVIEW

MediaPipe was used by Lavanya Vaishnavi et al. (2022) [1] to implement a gesture recognition system. Their work showed the precision and speed of MediaPipe's landmark detection and concentrated on real-time recognition of simple hand gestures. It highlighted how recognition of gesture can be used in interactive systems like virtual interfaces and automation. MediaPipe Hands, a sophisticated framework for identifying and tracking 21 3D hand landmarks with a webcam, was submitted by Zhang et al. (2020) [2]. The model supports real-time applications and operates effectively on-device. Because they eliminate the need for powerful GPUs, their contribution serves as the basis for numerous lightweight recognising hand gestures was. A thorough analysis technique for recognising hand gestures was provided by Mohamed et al. (2021) [3], who contrasted various methods such as image processing, machine learning, and deep learning. The capabilities of MediaPipe were expanded by Sung et al. (2021) [4] to enable on-device real-time gesture classification. Their system used a lightweight classifier built on top of MediaPipe's landmarks to recognize predefined gestures. High-speed performance was ensured by the model's optimization for embedded and mobile platforms.
A CNN-based model for real-time sign language detection was proposed by Saiful et al. (2022) [5]. Their system demonstrated excellent performance in identifying dynamic hand signs after being trained on custom datasets. Their approach automatically learned features from labeled data, in contrast to rule-based systems.Tayade and Halder (2021) [6] used a combination of MediaPipe and machine learning to recognize vernacular sign language. Their system could help make sign language systems more inclusive by recognizing gestures unique to regional languages.An early review of the literature on hand gesture recognition was given by Khan and Ibraheem (2012) [7]. By contrasting glove-based, vision-based, and sensor-driven approaches, their work established the foundation. It brought attention to the difficulties with processing speed, background noise, and gesture variation.An IoT-based vision-based gesture-controlled home automation system was proposed by Jayanthi et al. (2021) [8]. Fans, lights, and other

appliances were controlled by gestures recorded by a webcam. This IoT and gesture recognition integration demonstrated the usefulness of the technology in smart home settings Bora et al. (2023) [9] used a deep learning model that was integrated with MediaPipe to work on the recognition of Assamese sign language. Their system proved the efficacy of integrating CNN architectures with skeletal data, accurately classifying region-specific signs in real time.A basic home automation system based on hand gestures was created by Anand and Mishra (2018) [10]. Their work was one of the first attempts to use gestures for contactless control, laying the groundwork for more sophisticated systems even though it only used simple image processing techniques.

III. PROPOSED SYSTEM

The suggested system is a voice feedback application and real-time hand gesture recognition tool made to support people who have trouble communicating verbally. The system continuously records video frames of the user's hand gestures using a webcam. It makes use of MediaPipe hand tracking module to determine which fingers are raised and to detect the locations of 21 landmarks on each hand. Based on identified hand landmarks, the system counts raised fingers using rule-based logic, with each count being associated with a predetermined message. Pyttsx3, an offline text-to-speech engine, is used to vocalize the gesture that has been recognized and display it on the video frame. The system, which was created in Python with OpenCV and pyttsx3, functions well on a typical. PC or laptop with a webcam and doesn't require any extra hardware, internet, or cloud services.

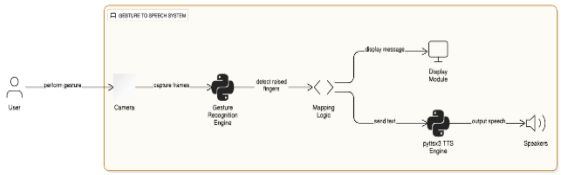

Fig 1.Block diagram of the gesture-to-speech system

IV. IMPLEMENTATION

A. System Workflow

To record real-time video frames, the system first sets up the webcam using OpenCV. To give the user a mirror-like experience, each frame is resized to 640x480 pixels and horizontally flipped. Each frame is then processed by MediaPipe's hand detection module, which recognizes 21 hand landmarks, such as joints and fingertips. The system determines which fingers are raised based on these landmarks. The thumb uses x-coordinates because of its side position, whereas the majority of fingers compare the y-coordinates of the tips and lower joints. The total number of raised fingers is counted, and each finger's binary status is recorded in a list.The offline text-to-speech engine pyttsx3 is used to speak the message aloud, and OpenCV is used to display it on the video frame. Until the user presses the 'q' key to exit, the system keeps detecting and reacting to gestures. The complete process is depicted in Fig 2

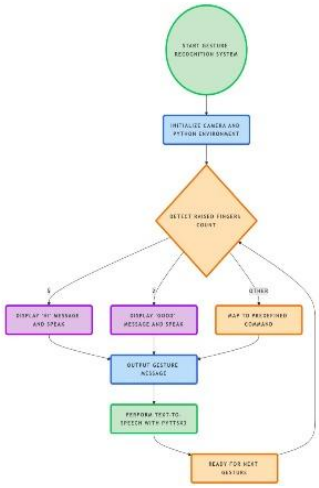

Fig 2. Flowchart of the Gesture Recognition and Voice Feedback System

B. Tools and Libraries Used

1.OpenCV

One popular open-source library that offers real-time image and video processing tools is called OpenCV (Open Source Computer Vision Library). The webcam of the system is interfaced with in this project using OpenCV, which also records frames and preprocesses images before sending them to the gesture recognition module. Additionally, it overlays the video stream with text and visual annotations like labels and finger counts, allowing for a dynamic user interface.

2. MediaPipe

Google created the open-source MediaPipe ML framework, which enables programmers to create

cross-platform, high-performance ML pipelines. MediaPipe, which is widely used in applications like gesture control, hand tracking, face detection, and video augmentation, was created to process streaming media, including audio and video, in real time. It is composed of four essential parts: Pre-built models for tasks like pose estimation, face detection, and hand tracking are available from MediaPipe Solutions. A versatile toolkit for building unique machine learning pipelines is the MediaPipe Framework.A web-based tool for pipeline testing and visualization is called MediaPipe Studio Model Maker: A tool for creating or refining models using particular datasets.

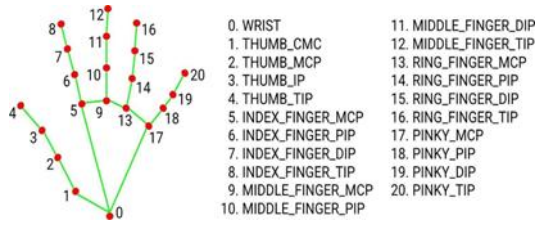

Fig 3. Mediapipe Representation

Working Principle: MediaPipe recognizes gestures by detecting hands in live video, extracting 21 important landmarks (wrist, joints, and fingertips), and providing structured data that is subjected to rule-based logic analysis. It is perfect for real-time gesture-driven applications like assistive communication or sign language interpretation because it is portable and works well even on low-resource devices.

V.HARDWARE SPECIFICATIONS

The suggested system is made to be lightweight and affordable, and it doesn't require any expert hardware like expensive GPUs, depth cameras, or external sensors. It runs effectively on common computer systems with simple add-ons. To run the application, a computer or laptop must have at least an Intel Core i3 processor (or equivalent), 4 GB of RAM, and a dual-core CPU operating at 1.6 GHz. An AMD Ryzen 5 or Intel Core i5/i7 processor with at least 8 GB of RAM is advised for improved real-time response and seamless performance. To record hand gestures, the system makes use of a simple webcam, either built-in or USB, with a minimum resolution of 640 x 480 pixels and a frame rate of 30 frames per second. The synthesized speech produced by the Pyttsx3 engine can be output from any internal or external speaker for audio feedback. Viewing the live video feed with gesture overlays requires a standard display screen that is 13 inches or larger. A microphone is optional for future extensions involving voice commands or feedback input, but it is not necessary for the current configuration.

VI. SOFTWARE REQUIREMENTS

The suggested system only needs a small amount of hardware and software and is made to be portable and easily accessed. Hardware-wise, the application can be run with a dual-core processor like the Intel i3 or its AMD equivalent, though a quad-core processor or more is advised for best results. For more seamless real-time processing, at least 4 GB of RAM is required, but 8 GB or more is ideal. While a high-definition (HD) webcam with 720p resolution or higher can improve gesture clarity and recognition accuracy, the system works well with a basic webcam with 640×480-pixel resolution at 30 frames per second (fps). Software-wise, the system is compatible with a number of widely used operating systems, including Windows, Linux, and macOS. To guarantee maximum compatibility, it is advised to use the most recent version of these platforms. Although Python 3.10 or the most recent stable release is recommended for better library support and performance, Python 3.7 or higher is needed to develop the core application. The main libraries used are pyttsx3 for offline text-to-speech functionality, MediaPipe for hand landmark detection and tracking, and OpenCV for image processing and webcam integration. The system is versatile and simple to implement because all necessary tools are open-source and platform-independent.

VII. RESULTS AND DISCUSSION

The accuracy, efficiency, and usability of the suggested hand gesture recognition system were confirmed through successful development and evaluation under various test conditions. To guarantee stability and adaptability, the system was tested on numerous users with a range of hand sizes and skin tones in various lighting conditions. The results of those tests are shown in this section along with the system's performance metrics and a discussion of its

advantages and disadvantages while operating in real time.

Three primary areas were the focus of the evaluation: The system's ability to accurately recognize and categorize hand gestures according to the quantity of raised fingers is known as gesture detection accuracy. Real-Time Performance is how quickly the system interprets video frames *and provides both audio and visual feedback. Practicality and User Experience: The system's* responsiveness and ease of use for end users, especially those who have speech or hearing impairments.

Gesture	Detected Finger Count	Expected Output	Actual Output	Voice Output
Open Hand	5	"Hi"	"Hi"	✅ Spoken
Two Fingers Raised	2	"Good"	"Good"	✅ Spoken
Fist (Closed Hand)	0	(Not mapped)	(No output)	❌ Not triggered
Three Fingers	3	"Need water"	"Need water"	✅ Spoken

Table I: Sample Test Results for Hand Gesture Recognition System

VIII. CONCLUSION

The suggested hand gesture recognition system uses open-source tools such as MediaPipe, OpenCV, and pyttsx3 to efficiently detect and interpret simple hand gestures in real time and translate them into voice output. It is accessible, works offline, and runs smoothly even on inexpensive hardware. The system provides instant text-to-speech feedback after correctly identifying gestures like "Hello" and "Good." It is appropriate for assistive communication as well as other uses, such as automation and education, due to its lightweight and intuitive design. This project demonstrates how computer vision and voice technology can be combined to create inclusive and interactive systems.

IX. FUTURE ENHANCEMENTS

Many of improvements are planned, even though the current system converts simple hand gestures into speech quite successfully. These include using time-series models to recognize dynamic gestures and broadening the gesture vocabulary to include all finger combinations and sign language alphabets. Custom gesture-to-command mapping might be possible with an intuitive user interface, and regional languages could be supported through integration with multilingual TTS engines (like gTTS). The system is helpful for assistive applications and smart homes since it can be expanded to control Internet of Things devices. Future iterations might have better background handling through segmentation or blurring, and be tailored for mobile and edge devices like Raspberry Pi. In the end, real-time sentence-level communication may be made possible by deep learning-based full sign language translation.

REFERENCE

[1] da, Lavanya Vaishnavi & C., Anil & S., Harish & L., Divya. (2022). MediaPipe to RecognisethHandGestures.WSEASTRANSACTIONSONSIGNALPROCESSING.18.134-139. 10.37394/232014.2022.18.19.

[2] Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C. L., & Grundmann, M. (2020). Mediapipe hands: On-device real-time hand tracking. arXiv preprint arXiv:2006.10214

[3] N. Mohamed, M. B. Mustafa and N. Jomhari, (2021) A Review of the Hand Gesture Recognition System: Current Progress and Future Directions, in IEEE Access, vol. 9, pp. 157422-157436, doi: 10.1109/ACCESS.2021.3129650.

[4] Sung, G., Sokal, K., Uboweja, E., Bazarevsky, V., Baccash, J., Bazavan, G., ... & Grundmann, M. (2021). On-device real-time hand gesture recognition. arXiv preprint arXiv:2111.00038.

[5] Saiful, Md & Isam, Abdulla & Moon, Hamim & Tammana, Rifa & Das, Mitul & Alam, Md & Rahman, Ashifur. (2022). Real-TimeSignLanguageDetectionUsingCNN.10.1109/ICDABI56818.2022.10041711.

[6] Tayade, Akshit & Halder, Arpita. (2021). Real-time Vernacular Sign Language Recognition using MediaPipe and Machine Learning. 10.13140/RG.2.2.32364.03203.

[7] Khan, Rafiqul Zaman & Ibraheem, Noor. (2012). Hand Gesture Recognition: A Literature Review. International Journal of Artificial Intelligence & Applications (IJAIA). 3. 161-174. 10.5121/ijaia.2012.3412.

[8] Jayanthi, R & Anbalagan, Bhuvaneswari & Rajkumar, S & Prabha, Rama. (2021). Vision

based Hand gesture pattern
[9] Bora, Jyotishman & Dehingia, Saine & Boruah, Abhijit & Chetia, Anuraag & Gogoi, Dikhit. (2023). Real-time Assamese Sign Language Recognition
[10] Anand, N., & Mishra, S. (2018). Home Automation Using Hand Gestures