Sign Language Translation Using Image Processing to Audio Conversion

V Subashini
Department of Electronics and Communication Engineering
Rajalakshmi Institute of Technology, Chennai, India
subashini.v@ritchennai.edu.in

B Someshwaran
Department of Electronics and Communication Engineering
Rajalakshmi Institute of Technology, Chennai, India
someshwaran.b.2019.ece@ritchennai.edu.in

S Sowmya
Department of Electronics and Communication Engineering
Rajalakshmi Institute of Technology, Chennai, India
sowmya.s.2019.ece@ritchennai.edu.in

S Ashwin Kumar
Department of Electronics and Communication Engineering
Rajalakshmi Institute of Technology, Chennai, India
ashwinkumar.s.2018.ece@ritchennai.edu.in

Abstract— Sign **language has long been a fundamental mode of communication for deaf and mute individuals, serving as a crucial tool for inclusivity and interaction. Nonetheless, communication barriers persist as many individuals outside of these communities struggle to comprehend and utilize sign language effectively. A ground-breaking system known as hand gesture recognition using image processing to audio conversion has emerged to address this issue. This innovative system aims to develop an application capable of translating sign language and hand gestures into text and audio, thereby facilitating communication between deaf and mute individuals and the wider society. This system enables the detection and classification of hand gestures by employing computer vision techniques, such as CNN algorithms for image processing and the MediaPipe framework for hand gesture identification in real-time video streams. Subsequently, the system utilizes audio signals to provide immediate feedback by converting the detected gestures into corresponding sounds. The feature extraction CNN algorithm is implemented in Python, while the execution takes place on a Raspberry Pi connected to an external camera utilizing OpenCV libraries. Through this comprehensive approach, the system endeavours to bridge the communication gap and enhance the inclusion of deaf and mute individuals in various social settings.**

Keywords—Hand Gesture recognition, Raspberry Pie, CNN-Convolutional neural network, Raspberry Pi, Real-time video streams, OpenCV libraries.

I. INTRODUCTION

Communication is a vital part of our lives, and it is especially important for people with hearing and speech impairments. Sign language is a visual and gestural language that is used by the deaf and dumb community, and it offers a means of communication that is rich, expressive, and culturally significant. However, the communication barrier between those who use sign language and those who do not poses significant limitations for inclusive interaction. In recent years, advancements in technology have opened new possibilities for overcoming this barrier and fostering inclusive communication. This research paper focuses on the development of a portable sign language conversion device, powered by Raspberry Pi and leveraging the MediaPipe framework. The objective is to create a compact and accessible solution that enables individuals with hearing and speech impairments to effectively communicate with individuals who do not understand sign language. The Raspberry Pi, a low-cost, credit-card-sized computer, provides a flexible platform for integrating computer vision and machine-learning capabilities. By utilizing the Raspberry Pi camera module and the MediaPipe framework, we aim to capture sign language gestures in real time and convert them into audio output that can be readily understood by non-sign language users. The proposed portable sign language conversion device holds great potential for enhancing the communication experience of individuals with disabilities. It not only empowers the deaf and dumb community to express themselves more freely but also enables a wider range of individuals to engage in inclusive conversations, fostering a more inclusive society. This research paper will explore the development process of the portable sign language conversion device, including the detection of sign language gestures using computer vision techniques, the recognition of these gestures using machine learning models, and the conversion of these gestures into meaningful audio output. The paper will also address design considerations for a user-friendly interface, device portability, and integration of necessary components to ensure seamless interaction between users. The significance of this research lies in its potential to break down communication barriers, fostering a more inclusive environment for individuals with hearing and speech impairments. By enabling real-time interpretation of sign language gestures into audio output, the proposed device opens up opportunities for greater social participation, educational integration, and professional engagement for individuals with disabilities.

2024 Third International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS) | 979-8-3503-6118-6/24/$31.00 ©2024 IEEE | DOI: 10.1109/INCOS59338.2024.10527627

II. RELATED WORK

Emphasizes the psychological depth of deaf and mute individuals and the importance of sign language as a means of expression. It highlights the significant communication challenges faced by this population due to limited access to certified interpreters. The proposed prototype assistive device aims to bridge this communication gap by enabling real-time recognition of sign gestures and converting them into speech [1]. Another system that offers a summary of the current methodologies and research in the field of Indian Sign Language recognition was put forth in the year 2020. It explores numerous strategies, including vision- and sensor-based ones, and draws attention to the difficulties and potential possibilities for this field of study [2]. In 2019, a Hadoop-based system was developed for recognizing hand gestures in Indian Sign Language (ISL). The paper discusses the use of computer vision techniques, including hand segmentation, feature extraction, and gesture classification, to accurately recognize and interpret ISL gestures. Experimental results and evaluations demonstrate the effectiveness of the proposed approach, highlighting its potential for facilitating communication between individuals who use ISL and others [4]. The system using MATLAB to apply image processing methods for gesture recognition goes over several image processing techniques used to extract useful information from gesture photographs, including thresholding, edge detection, and contour analysis [5]. The system developed in 2021 explores face detection technologies based on OpenCV in mobile augmented reality (AR) applications. It goes over how to use OpenCV to develop face detection algorithms and how mobile AR systems can use them. To show the efficacy and precision of the suggested approach in real-time face detection for mobile AR scenarios, it advances the industry by offering details on how to use face detection powered by OpenCV to improve mobile AR experiences [6]. A system that translates sign language movements into spoken words was put forth in 2016 to enhance communication between those who have hearing or speech difficulties and others. The system has modules for text-to-voice speech synthesis and image-processing techniques for sign language recognition [7]. The proposed solution is a two-way sign language translator that facilitates the real-time translation of speech into sign language and vice versa. To achieve this, each video frame is processed using Python's OpenCV Library, allowing for effective analysis and manipulation of the visual data. Additionally, the Gaussian Mixture-based Background/Foreground Segmentation Algorithm is utilized to remove the background from each frame, enhancing the accuracy and clarity of the translation process. By implementing this technology, the hearing-impaired community can experience increased integration into regular schools, leading to improved accessibility and affordability of education for them [8]. This paper proposes a method for representing sign language sentences using advanced image processing, graphics rendering, and signal processing techniques. The approach aims to bridge the communication gap between sign language users and non-signers by encoding gestures, handshapes, and facial expressions into a concise and standardized form. Experimental results demonstrate the effectiveness of the method, highlighting its potential for enhancing sign language communication[9]. Existing systems have shown promising results in sign translation, but occasional misclassifications still occur. To improve the accuracy of sign classification, it is important to detect and classify frequently misclassified signs. Our proposed system offers a communication interface that enables bidirectional sign language conversion. It employs Convolutional Neural Networks (CNNs) as the classification model, achieving a training accuracy of 94.86% [10]. Development of an interactive device aimed at assisting individuals who are unable to speak. The device utilizes a customized sign language system to facilitate communication between mute individuals and non-disabled individuals. By detecting finger movements through flex sensors and processing the corresponding signals, the device generates natural-sounding voice outputs. The customization feature of the device enhances its interactivity and adaptability to individual needs[11]. To facilitate communication, a glove equipped with sensors is utilized in this paper. These sensors detect the gestures made by the user and convert them into text data. An analog-to-digital converter on an Arduino Nano board assists in the conversion process. The converted text data is then transmitted wirelessly via a WIFI module to a cell phone. The WIFI module is responsible for sending the gesture information to a smartphone. Subsequently, Text-to-Speech software is employed to convert the incoming text message into an audible voice [12].

III. EVALUATION METHODOLOGY

step 1: Select the hardware platform. The Raspberry Pi is a good choice for this project because it is affordable, powerful, and portable.

Step 2: Install the necessary software. This includes OpenCV, MediaPipe, and a machine learning library like TensorFlow or PyTorch.

Step 3: Create a dataset of sign language gestures. This can be done by recording yourself or someone else signing the alphabet, numbers, and common words.

Step 4: Train a model to recognize sign language gestures. This can be done using a machine learning library like TensorFlow or PyTorch.

Step 5: Create a user interface for the device. This user interface should allow the user to sign gestures and hear the audio output.

Step 6: Implement real-time gesture recognition. Once you have trained your model, you need to integrate it with Raspberry Pi and OpenCV to perform real-time gesture recognition. This involves capturing video input from a camera connected to the Raspberry Pi, processing the frames using OpenCV, and using your trained model to classify the gestures.

Step 7: Enhance the user interface with feedback and interaction. To make the user experience more engaging, you can provide visual feedback on the screen or LEDs to indicate whether the detected gesture is correct or incorrect. Additionally, you can incorporate interactive features such as voice prompts or a display of recognized text corresponding to the signed gesture to further assist the user in learning sign language.

979-8-3503-6118-6/24/$31.00 ©2024 IEEE

IV. Structured Approach

A. Data Acquisition and Preprocessing:

1) Camera and Image Acquisition:

- Choose a suitable camera module for the Raspberry Pi, considering factors like resolution, frame rate, and power consumption.
- Implement code to capture continuous video streams using OpenCV libraries like cv2.VideoCapture.
- Address potential lighting challenges by utilizing OpenCV functions for brightness and contrast adjustments, noise reduction filters, and adaptive thresholding.

2) Preprocessing and Hand Region of Interest (ROI) Extraction:

- Apply background subtraction techniques to isolate the user's hand from the background in each frame.
- Implement skin tone detection algorithms using OpenCV functions like cv2.inRange to further refine the hand ROI.
- Employ morphological operations like erosion and dilation to remove noise and improve hand contours.

B. MediaPipe Hand Landmark Detection and Pose Estimation:

- Utilize MediaPipe's hand detection model (media pipe.solutions.hands) to identify key hand landmarks (wrist, knuckles, fingertips) in real-time.
- Extract relevant pose information based on landmark positions, such as finger angles, palm orientation, and hand movement trajectories.
- Consider training a custom MediaPipe hand model for improved accuracy on specific sign language gestures.

C. Gesture Recognition and Classification:

- Utilize machine learning algorithms for classifying captured hand poses and gestures.
- Train a suitable model (e.g., Support Vector Machines, Random Forests) on a labeled dataset of sign language gestures.
- The dataset should include variations in hand size, signing speed, and lighting conditions.
- Implement real-time gesture recognition using the trained model and extracted MediaPipe pose features.
- Optimize the recognition algorithm for speed and accuracy on the Raspberry Pi hardware platform.

D. Text-to-Speech Conversion and Audio Output:

- Utilize a text-to-speech (TTS) engine like eSpeak or Festival to synthesize audio output from the generated text.
- Consider utilizing speaker calibration and noise reduction techniques for clear and intelligible audio output.
- Design an intuitive user interface for displaying recognized signs and synthesized speech on the device.

E. System Integration and Optimization:

- Integrate all processing modules for seamless data flow from image capture to audio output.
- Optimize the system for real-time performance and low latency on the Raspberry Pi platform.
- Implement power-saving measures like dynamic clock scaling and resource management.
- Design a user-friendly interaction system for intuitive device control and feedback.

F. Evaluation and Testing:

- Define evaluation metrics like accuracy, recognition speed, and user satisfaction.
- Conduct rigorous testing with diverse sets of users and sign language vocabulary.
- Analyze errors and limitations to identify areas for improvement.
- Consider inviting feedback from deaf and hard-of-hearing individuals for user-centered design improvements.

V. IMPLEMENTATION

Implementation of the overall Sign language and hand gestures are discussed above below

A. Block Diagram

Figure 1 shows the working of Raspberry Pi 3, Raspberry pi is used to detect the position of the hand gesture using a Raspberry Pi camera and process it to identify which sign language and translate that sign language to audio, we want to install Linux Os and we use Thonny as a Python Interpreter for Executing the program. For Testing and simulation purposes we have connected a USB camera, we can also use Arducam specially designed for Raspberry Pi for Portability. The Audio Speaker Is connected to implement the voice that has been converted from the text. We have connected a Rechargeable Battery as a Power Source for the Functioning of Raspberry Pi.

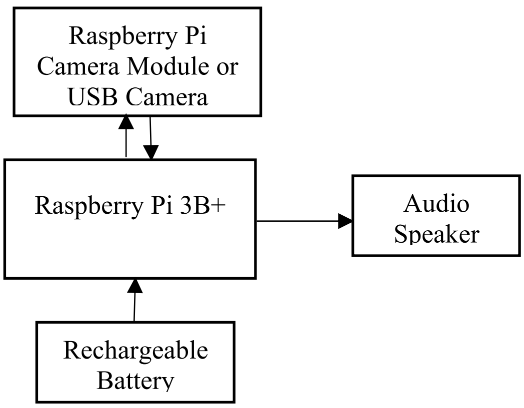


Figure 1. Block diagram of Hardware

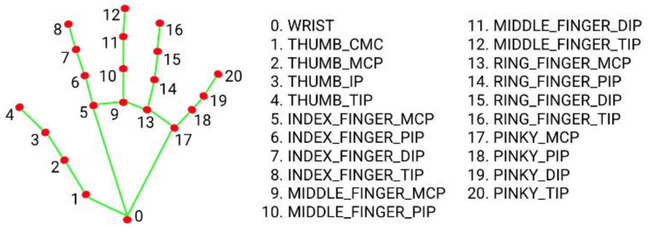
979-8-3503-6118-6/24/$31.00 ©2024 IEEE

B. *Detection of Fingers*

Figure 2 shows how the MediaPipe framework detects fingers using the two-step process.

Palm detection: The first step is to detect the palm. This is done using a machine learning model that identifies the characteristic features of a palm, such as the shape of the fingers and the curve of the wrist.

Finger landmark detection: The next step is to identify the individual fingers once the palm has been detected. This is done using a machine-learning model that identifies 21 landmarks in another hand, including the tips of the fingers, the knuckles, and the base of the palm.



In Figure 2

C. *Hand Tracking*

In Figure 3. The main function of hand tracking is to determine the gesture. We mark Respective fingers as an index, Middle, Ring, Little, and Thumb. By Determining which finger is Raised and which finger is folded we use an adding algorithm and operation to determine the hand gesture. This also helps to count the fingers and display text and audio. We can give predetermined logic and text for the hand gestures shown in the camera.

Table 1: Hand tracking with corresponding text to be displayed and converted to audio.

Fingers that have lifted and folded					**Text to be displayed**
Thumb	**Index**	**Middle**	**Ring**	**Little**	
True	True	False	False	False	I want
True	False	False	False	False	Why?
False	True	True	False	False	Peace
True	False	False	False	False	ok
True	True	True	True	True	Please
False	True	True	True	True	Thank you
True	True	True	False	False	No
False	True	False	False	True	I love you

In Figure 3

In figure 3 if the combination of fingers that are lifted and folded matches the corresponding table text is displayed and converted into audio.

D. *Results of Simulation*

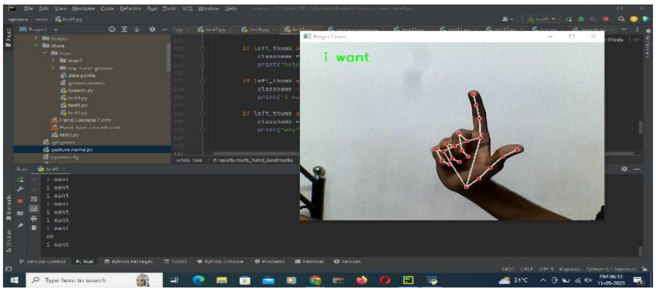


Figure 4. Hand Gesture

In Figure 4. The hand gesture has been Recognized the displayed text is “I want.”

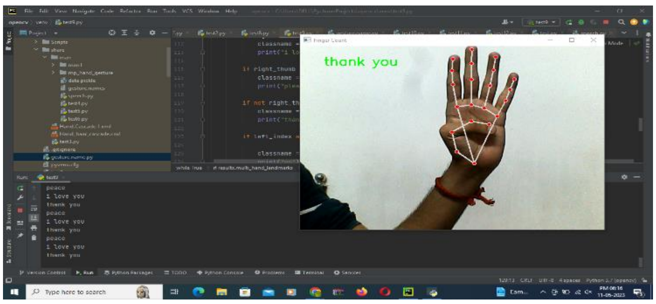


Figure 5. Hand Gesture

In Figure 5. The hand gesture has been Recognized the displayed text is “Thank You.”

979-8-3503-6118-6/24/$31.00 ©2024 IEEE

E. Hardware

Figure 6. Hardware Components

In Figure 6. It shows a Prototype of all the connected hardware components that are used to recognize hand gestures and translate them into audio.

F. Enhancing Portable Sign Language Conversion Device Performance

Hardware Optimization: Raspberry Pi offers various models with different specifications. Choose a Raspberry Pi model with higher processing power, such as the Raspberry Pi 4, which has a faster CPU, more RAM, and improved GPU capabilities. Upgrading to a more powerful model can enhance the overall performance of your device.

Software Optimization: Optimize your software implementation to make it more efficient on the Raspberry Pi platform. This includes leveraging hardware acceleration, such as using the GPU for parallel processing tasks. Explore libraries like OpenCL or TensorFlow Lite for utilizing the GPU effectively. Additionally, optimize code for better utilization of CPU cores, minimize unnecessary computations, and utilize efficient algorithms.

Frame Rate and Resolution: Consider reducing the camera frame rate or lowering the resolution if it does not significantly affect the accuracy of the gesture detection. This can help reduce the computational load and increase the system's overall performance on Raspberry Pi.

VI. CONCLUSION AND FUTURE WORK

In conclusion, sign language is essential for communication and inclusion for deaf and mute individuals. However, communication barriers exist due to the difficulty that many people face in understanding and using sign language. Developing a system for hand gesture recognition using image processing and audio conversion could significantly aid in bridging the communication gap between sign language users and non-sign language users. The system developed in this project can recognize hand gestures in real time and provide feedback in the form of audio signals, thus allowing for easy communication with sign language users.

Future Work of this project provides several ways that the device could be improved in the future. One way would be to increase the accuracy of the recognition algorithm. This could be done by collecting a larger dataset of sign language gestures and by using a more powerful machine learning model. Another way to improve the device would be to make it more portable. This could be done by using a smaller and lighter Raspberry Pi or by developing a mobile app that could run on a smartphone.

Overall, the device presented in this paper is a promising step toward making sign language more accessible to people who are deaf or hard of hearing. With further development, the device could become a valuable tool for communication and education.

The system could be enhanced to recognize more complex hand gestures, facial expressions, and body language, thus improving the overall accuracy of the translation. Additionally, the application could be further developed to support multiple languages, thus allowing for easy communication across different language barriers. Lastly, the system's accessibility could be improved by making it more user-friendly and accessible to a wider range of devices beyond Raspberry Pi.

Additional Future Scope Into the improvements mentioned above, there are several other ways that the device could be expanded in the future. For example, the device could be used to translate sign language into other languages. This would allow people who are deaf or hard of hearing to communicate with people who speak different languages. The device could also be used to provide feedback to people who are learning sign language. This feedback could be used to help people improve their sign language skills.

ACKNOWLEDGMENT

The authors deeply convey thanks of gratitude to their professors for their encouragement and support.

REFERENCES

[1] Boppana, Lakshmi & Ahamed, Rasheed & Rane, Harshali & Kodali, Ravi. (2019). Assistive Sign Language Converter for Deaf and Dumb. 302-307. 10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00071.

[2] Ruize Xu, Shengli Zhou, W. J. LiMEMS. Accelerometer Based Nonspecific-User Hand Gesture Recognition IEEE Sensors Journal 12(5):1166-1173

[3] Kumar, D & Madhukar, M & Prabhakara, Adarsh & Marathe, Archana & Manoj, & Bharadwaj, Shyam. (2019). Sign Language to Speech Conversion — An Assistive System for Speech Impaired. 272-275. 10.1109/ICATIECE45860.2019.9063849.

[4] M. Jose, V. Priyadharshni, M. S. Anand, A. Kumaresan, N. M. Kumar. Indian Sign Language (ISL) Translation System For Sign Language Learning International Journal of Innovative Research and Development.

[5] Chowdary, P & Babu, M & Subbareddy, Thadigotla & Reddy, Bommepalli & Vellaiappan, Elamaran. (2014). Image processing algorithms for gesture recognition using MATLAB. 1511-1514. 10.1109/ICACCCT.2014.7019356.

[6] Subedi, Basudev & Dorji, Kelzang & Wangdi, Pema & Dorji, Tshering & Muramatsu, Kazuhiro. (2021). Sign Language Translator of Dzongkha Alphabets Using Arduino. 1-6. 10.1109/i-PACT52855.2021.9696641.

[7] M. Aarthi, P. Vijayalakshmi Sign language to speech conversion. 2016 International Conference on Recent Trends in Information Technology (ICRTIT).

[8] K, Vishnu & S, Jayashri & V, Sivanjali & V, Sneha & K, ThamaraiSelvi. (2023). Two Way Sign Language for Deaf and Dumb using Deep Convolution Neural Network. International Journal for Research in Applied Science and Engineering Technology. 11. 3830-3835. 10.22214/ijraset.2023.51125.

[9] BM, Chethankumar & Siddappa, Nagendraswamy. (2016). LBPV for Recognition of Sign Language at Sentence Level: An Approach based on Symbolic Representation. H S, Nagendraswamy; B M, Chethana Kumara. Journal of Intelligent Systems.

979-8-3503-6118-6/24/$31.00 ©2024 IEEE

[10] Kannan, Rajeswari & Hugar, Naveen & Pulli, Rakesh & Rajput, Shubham & Tayade, Rushikesh & Vispute, Sushma. (2022). A Portable System for Bidirectional Sign Language Translation for Deaf and Dumb. 1-5. 10.1109/ICCUBEA54992.2022.10011048.

[11] Hasan, Md & Hasan, Md. (2022). Design and Development of a Sign Language to Speech Converter for Dumb People. International Journal of Research Publication and Reviews. 1939-1944. 10.55248/gengpi.2022.3.9.52.

[12] Patil, Shweta. (2019). Sign Language Interpreter for Deaf and Dumb People. International Journal for Research in Applied Science and Engineering Technology. 7. 354-359. 10.22214/ijraset.2019.9050.

[13] Khan, Saleh Ahmad & Joy, Amit & Asaduzzaman, S. & Hossain, Morsalin. (2019). An Efficient Sign Language Translator Device Using Convolutional Neural Network and Customized ROI Segmentation. 152-156. 10.1109/ICCET.2019.8726895.

[14] Grover, Yuvraj & Aggarwal, Riya & Sharma, Deepak & Gupta, Prashant. (2021). Sign Language Translation Systems for Hearing/Speech Impaired People: A Review. 10-14. 10.1109/ICIPTM52218.2021.9388330.

979-8-3503-6118-6/24/$31.00 ©2024 IEEE