JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Speech to Sign Language Translator using Python

[1]K Poornima Kamath, [2]Murugesh M S
[1,]Department of Electronics and Communication Engineering,
[1] BMS College of Engineering, Bangalore, INDIA.

Abstract : A vital form of communication for those who are deaf or hard of hearing is sign language, enabling effective expression and interaction. Communication barriers persist due to limited accessibility and understanding of sign language. This system addresses these challenges using Python and Vosk for speech recognition, with Matplotlib for visualization. Google MediaPipe and OpenCV are employed to process and save keypoints during training, which are stored in JSON files for easy access. These keypoints are used to generate sign language actions. The model, trained with a dataset of spoken words mapped to sign language gestures, allows for real-time translation of speech to sign language. This integration provides immediate visual feedback through 2D animation of the sign output, enhancing communication and accessibility.

IndexTerms - **Sign Language Translation, Computer Vision, Google MediaPipe, Vosk, speech recognition, Matplotlib, 2D animation ploting, Python**

I. Introduction

Indian Sign Language (ISL) is a visual-gestural language for the deaf and hard of hearing community in India. It is the primary mode of intraction for individuals who are deaf or have hearing impairments across various regions and linguistic backgrounds within the country. ISL is distinct from spoken languages such as Hindi, English, or regional languages, and it has its own grammar, vocabulary, and syntax.

ISL is not a universal language; rather, it varies regionally within India, with different signs and dialects existing across the country. However, efforts have been made to standardize ISL through initiatives like the Indian Sign Language Research and Training Centre (ISLRTC), which was established in 2015 under the aegis of the Ministry of Social Justice and Empowerment.

ISL is a visual language primarily conveyed through handshapes, movements, facial expressions, and body postures. It employs a combination of manual signs, finger spelling, and non-manual markers to convey meaning and express concepts. Similar to spoken languages, ISL exhibits linguistic properties such as phonology, morphology, syntax, and semantics. The ISL plays its role as a medium for communication and expression, enabling deaf individuals to interact with each other also with the hearing community. It facilitates access to education, information, employment opportunities, and social integration for the hard of hearing population in India. Deaf and mute individuals encounter numerous challenges in their daily lives due to communication barriers and societal misconceptions. Simple tasks which doesn't require importance, like engaging in conversations, accessing information, or seeking assistance, become daunting obstacles for those who rely on sign language or written communication. The lack of widespread understanding and acceptance of sign language often leads to frustration and isolation, as communication breakdowns hinder their ability to express themselves, convey their needs, or comprehend spoken language. Furthermore, inaccessible environments, such as public spaces lacking sign language interpreters or multimedia content without captions or sign language translations, exacerbate the sense of exclusion and marginalization experienced by the deaf and mute community. These challenges restrict their educational and career prospects in addition to impeding their social relationships, perpetuating systemic inequalities and hindering their full participation in society. As a result, addressing these challenges through improved accessibility, inclusive policies, and greater awareness of the needs of deaf and mute individuals is essential to fostering a more equitable and inclusive society for all.

Present challenges faced by deaf and mute individuals encompass various aspects of life, including communication, education, employment, and social inclusion. Some of the key challenges include:

- Communication Barriers: Deaf and mute individuals often encounter barriers to effective communication due to limited awareness and understanding of sign language among the general population. This lack of communication accessibility hinders their ability to communicate with people and obtain information, and participate fully in social, educational, and professional settings.

- Limited Access to Education: Deaf and mute individuals may face challenges in accessing quality education that meets their unique needs. Schools and educational institutions may lack resources, trained educators, and appropriate accommodations, leading to disparities in learning outcomes and opportunities for academic and personal development.

- Employment Discrimination: Deaf and mute individuals frequently encounter discrimination and barriers to employment opportunities due to misconceptions about their abilities and communication preferences. Many employers may be reluctant to hire deaf or mute individuals, citing concerns about communication barriers or the perceived need for special accommodations.

- Limited Social Inclusion: Deaf and mute individuals may experience social isolation and exclusion due to communication barriers and societal stigma. They may struggle to participate fully in social activities, community events, and interpersonal relationships, leading to feelings of loneliness, alienation, and low self-esteem.

In this study, aims to contribute towards bridging the gap in sign language accessibility by providing an affordable and user-friendly solution for sign language detection, the key contributions of this study are as follows:

- Investigating and selecting appropriate computer vision algorithms and machine learning techniques for sign language recognition.

- Developing software modules for hand gesture detection, feature extraction, and classification using low-tech input devices.

- Designing a user-friendly interface for real-time visualization and interpretation of sign language gestures.

- Evaluating the system's robustness, accuracy, and speed in a variety of environmental circumstances.

- Assessing the usability and accessibility of the model through user testing and feedback collection.

II. RELATED WORK

We reviewed different approaches and designs for the sign language or gesture animation. Few of them are as listed below from which we have improved our project. Yoga Pose Detection for Health and Wellness Using Raspberry pi [1] system's core advantage lies in its utilization of MediaPipe, an open-source framework developed by Google, which efficiently identifies and tracks essential body landmarks over time. These landmarks, alongside Tensor Flow's capabilities, are pivotal in extracting pertinent features from images and videos. OpenCV, a renowned computer vision library, further enhances the system by processing visual data. The user interface is designed to offer user convenience and efficient interaction. By providing real-time feedback through voice commands, the system assists users in achieving proper yoga poses. The Raspberry Pi, equipped with its camera module, serves as the hardware foundation. Its affordability, adaptability, and compact size make it an ideal choice for on-device processing, ensuring low latency and responsive performance. Research on using convolutional neural networks (CNN) to recognize static ISL gestures demonstrates models' capability to accurately identify hand gestures from RGB-D images [2][3]. Real-time translation systems for ISL using dynamic hand gesture recognition have been developed. Key frame extraction and feature fusion play a crucial role in enhancing recognition accuracy [3][4]. Spatio-temporal frameworks for recognizing dynamic ISL gestures by analyzing Spatio-temporal features and integrating deep learning techniques improve the recognition of continuous sign language sequences [5]. Systems that eliminate co-articulation effects in sign language recognition, making them independent of individual signers, enhance the robustness and accuracy of the recognition process [6]. Automated speech to sign language conversion systems using Google's speech recognition API and NLP techniques aim to bridge communication gaps for the hearing impaired [7]. Active learning platforms for sign language education, even when focused on other sign languages, provide methodologies adaptable for ISL. These platforms use deep learning to offer interactive sign language education [8]. LabVIEW-based hand gesture recognition systems for the deaf and dumb offer valuable frameworks and recognition techniques for ISL applications [9].

III. SOFTWARE INFORMATION

3.1 Python (Version 3.8.10)

Python is a high-level programming language suitable for developing the project. Version 3.8.10 was utilized as the primary programming language to implement various functionalities, including data preprocessing, model development, and system integration.

3.2 PyCharm

PyCharm is an integrated development environment (IDE) used for coding, debugging, and testing Python applications. It provided a user-friendly interface and advanced features such as code completion, syntax highlighting, and version control integration, facilitating efficient development and collaboration.

3.3 Google MediaPipe

Google MediaPipe is an open-source framework developed by Google Research that empowers developers to build real-time multimedia processing pipelines for a diverse range of applications. MediaPipe leverages machine learning techniques, including deep neural networks, to analyze and interpret multimedia data streams, such as video and audio, in real-time. With its modular architecture and customizable components, MediaPipe enables developers to create sophisticated perception-based applications with minimal effort, facilitating tasks like augmented reality, gesture recognition, sign language interpretation, and immersive experiences. The framework's flexibility, efficiency, and cross-platform compatibility make it well-suited for deployment on various devices, including smartphones, tablets, desktops, and embedded systems. MediaPipe's comprehensive documentation, community support, and integration with popular machine learning frameworks such as TensorFlow ensure accessibility and scalability for developers seeking to harness the power of multimedia processing in their applications.Pre-trained Models: Media Pipe provides pre-trained deep learning models trained on large datasets of annotated images and videos. These models are trained using convolutional neural networks (CNNs) and other techniques to accurately detect and localize hands in various poses and orientations.

3.3.1 Google MediaPipe Keypoints Location for Building Sign Database

The Google MediaPipe Landmarkers are used in this project to save keypoints of Hand, Pose and Face outline into json file. The json files are used during the translation of speech to 2-dimensional sing language using matplotlib.

3.3.1.1 Hand Landmarks Model

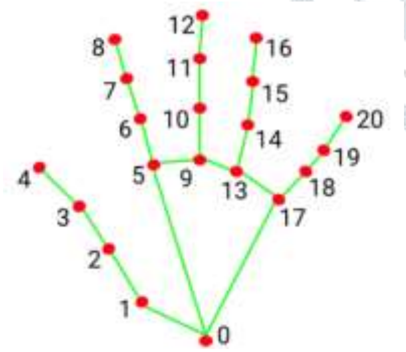


FIGURE 1 Hand Landmark Model

0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMP_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_FINGER_MCP
18. PINKY_FINGER_PIP
19. PINKY_FINGER_DIP
20. PINKY_FINGER_TIP

The hand landmarker model bundle contains a palm detection model and a hand landmarks detection model. The Palm detection model locates hands within the input image, and the hand landmarks detection model identifies specific hand landmarks on the cropped hand image defined by the palm detection model.Since running the palm detection model is time consuming, when in video or live stream running mode, Hand Landmarker uses the bounding box defined by the hand landmarks model in one frame to localize the region of hands for subsequent frames.

3.3.1.2 Pose Landmarks Model

0. NOSE
1. LEFT_EYE_(INNER)
2. LEFT_EYE
3. LEFT_EYE_(OUTER)
4. RIGHT_EYE_(INNER)
5. RIGHT_EYE
6. RIGHT_EYE_(OUTER)
7. LEFT_EAR
8. RIGHT_EAR
9. MOUTH_(LEFT)
10. MOUTH_(RIGHT)
13. LEFT_ELBOW
14. RIGHT_ELBOW
15. LEFT_WRIST
16. RIGHT_WRIST
17. LEFT_PINKY
18. RIGHT_PINKY
19. LEFT_INDEX
20. RIGHT_INDEX
21. LEFT_THUMB
22. RIGHT_THUMB
23. LEFT_HIP

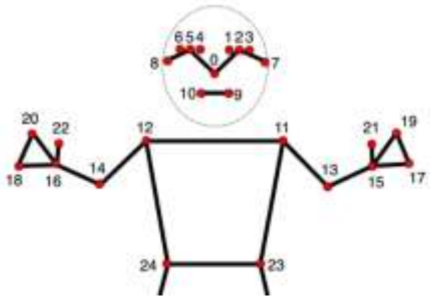


FIGURE2 Hand Landmark Model

11. LEFT_SHOULDER
12. RIGHT_SHOULDER
24. RIGHT_HIP

Landmarks in real-time video streams. These landmarks correspond to various critical points on the human body, facilitating the accurate tracking and analysis of human poses. Below is a detailed explanation of how the Pose Landmark Model operates, for this project, since the landmarks after the Hips are not required for the Sign Language, as hand and face landmarks with upper body pose landmarks are enough, so only first 24 keypoints are used.

3.3.1.3 Face Landmarks Model

The MediaPipe Face Landmarker task detects face landmarks and facial expressions in images and videos. This task identifies human facial expressions, apply facial filters and effects, and create virtual avatars. This task uses Machine Learning (ML) models that can work with single images or a continuous stream of images. The task outputs 3-dimensional face landmarks. For this project, only the outline of face, eyes and mouth are used to present the 2-dimensional face in speech to sign language translation model. The reason for using only the face outline instead of whole face mesh is that the total face mesh has around 465 keypoints including the 3-dimensional data, so including all those keypoints in the speech to sign language translation model will increase the model latency and also since the output is 2-dimensional only face outline keypoints will serve enough data to create output.

3.4 Vosk Offline Speech Recognition

Vosk is an offline speech recognition toolkit that allows for accurate and reliable transcription of spoken language without requiring an internet connection. In the context of this project, Vosk is utilized for the speech-to-sign language translation component. Here's a detailed explanation of its working principle

The first step involves capturing speech input using a microphone or any audio recording device. In this project, the speech input is captured using a webcam with an integrated microphone, providing a simple and accessible setup.

Once the audio is captured, it undergoes preprocessing to enhance the quality and prepare it for recognition. This includes noise reduction, normalization, and segmenting the audio into manageable chunks. Vosk's built-in preprocessing capabilities handle these tasks efficiently.

Vosk uses deep learning models trained on large datasets of spoken language. These models are designed to recognize and transcribe speech into text accurately. The project utilizes Vosk's pre-trained models suitable for the target language (e.g., English or any other relevant language).

The processed text is then mapped to corresponding sign language gestures. This mapping involves using a predefined dictionary of text-to-sign language translations. Each word or phrase in the text is matched with its equivalent sign language gesture, creating a sequence of signs representing the spoken input.

3.5 Matplotlib

Matplotlib is a comprehensive library in Python for creating static, animated, and interactive visualizations. In this project, Matplotlib is used to create 2D animations of sign language gestures, enabling users to see visual representations of the detected speech forms. Here's a detailed explanation of its working principle

The first step involves preparing the data for visualization. In this project, the data consists of sign language gestures represented as sequences of keypoints. These keypoints are extracted using computer vision techniques, such as hand tracking with Google MediaPipe.

Matplotlib's configuration options allow for customizing the appearance of plots and animations. This includes setting up the plot area, defining axes, and choosing appropriate color schemes and markers to represent the sign language gestures effectively.

For static visualizations, Matplotlib plots the keypoints on a 2D graph, showing the positions of hand and body landmarks. This helps in analyzing and understanding individual gestures in a static context. The static plots can be saved as images for documentation or further analysis.

For dynamic visualizations, Matplotlib's animation module is used. The FuncAnimation class from Matplotlib's animation module facilitates creating animations by updating the plot at regular intervals. In this project, each frame of the animation represents a different position of the hand and body landmarks, showing the movement over time.

3.6 PyQt6

PyQt6 is a set of Python bindings for the Qt libraries, which are widely used for developing cross-platform applications. The Qt framework offers a comprehensive suite of libraries for creating rich, modern user interfaces, making it a popular choice for both desktop and embedded systems development.

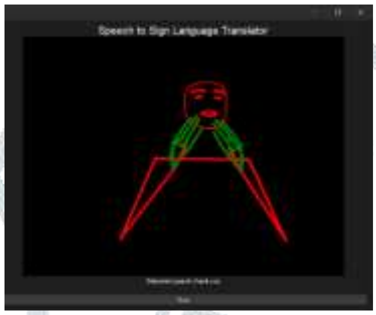


FIGURE 3 GUI for Speech to Sign Language Translator

Uses Matplotlib to display a 2D animation of the recognized sign corresponding to the detected speech. This animation helps users understand the sign language representation of the spoken words. Shows the text of the detected speech, confirming to the user what has been recognized and translated.The “Start” button initiates the translation process, allowing users to start the detection and recognition workflows for speech-to-sign language functionalities. Provides real-time updates on the system's status, such as "Listening for speech" or "Sign detected: word."

4. Working of The Model

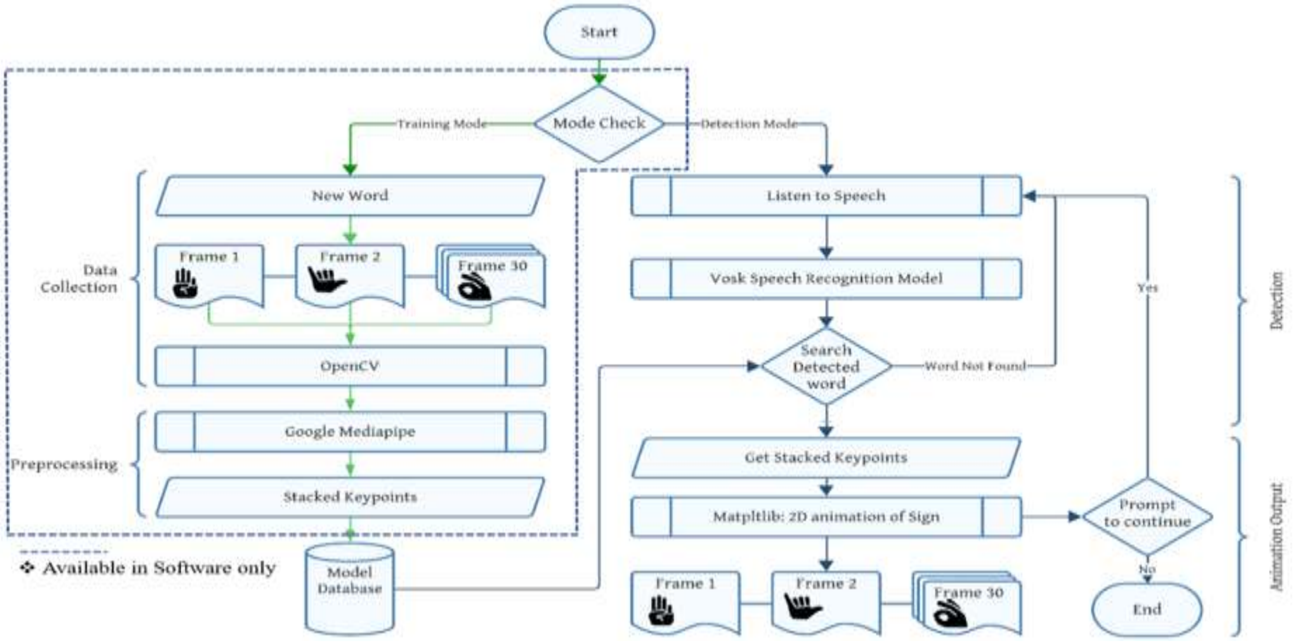


FIGURE 4 Flow Chart of Speech to Sign Language

4.1 Data Collection

In the data collection phase, frames are extracted from a video feed utilizing OpenCV. Each action is represented by a comprehensive set of 30 frames. During training mode, the videos are meticulously collected to encompass a wide spectrum of visual scenarios, contributing to effective training outcomes. Following collection, the frames undergo pre-processing to enhance quality and standardize features, facilitating subsequent stages of analysis and model development.

4.2 Preprocessing

In the pre-processing phase, Google's MediaPipe library is harnessed to detect facial features, as well as landmarks for the right and left hands, along with overall body pose in the video sourced from OpenCV. These landmarks are meticulously identified and augmented with bounding boxes for enhanced contextual understanding.

MediaPipe typically detects 21 landmarks for each hand, capturing the intricate movements and positions of the fingers, palm, and wrist. MediaPipe identifies 33 landmarks to represent the body pose, including key points on the torso, arms, legs, and joints, enabling comprehensive tracking of human posture and movements, along with it also captures the outline of face landmarks including the eyes and mouth. Subsequently, the stacked keypoints, comprising the detected landmarks and keypoints for each frame, are stored in a json format which are easy to read during the detection phase, the saved json files will serve as a database for the speech to sign language system.

4.3 Detection

The system initialization begins the process. All necessary components, such as the microphone, webcam, and software libraries, are prepared for operation. The system uses the microphone of the connected webcam to capture the audio input. This is the first step in translating spoken language into sign language. The captured audio input is processed using the Vosk offline speech recognition model. Vosk is a versatile speech recognition toolkit that converts spoken words into text, enabling the system to interpret the speech.

4.4 Search Detected Word

The recognized text (converted speech) is then searched within a pre-defined database of words. This database contains words and their corresponding sign language representations. If the word is found in the database, the system retrieves the associated sign language keypoints. If the word is not found, the system will prompt the user to continue or repeat the process.

4.5 Detection

The system verifies if it's in detection mode, proceeding to search the model database for gesture recognition. Upon detection, the corresponding word is displayed based on the search outcome, facilitating seamless interaction.

4.6 Animation Output

If the word is found in the database, the system retrieves the stacked keypoints corresponding to the sign language representation of the word. These keypoints represent the hand and body positions needed to perform the sign. The system uses Matplotlib to create a 2D animation of the sign language gesture. Matplotlib is a plotting library in Python that can be used for creating static, animated, and interactive visualizations. In this context, it is used to visualize the sequence of keypoints as a coherent animation of the sign.

4.7 Dataset Preparation for Training

In the data collection phase, frames are extracted from a video feed using OpenCV. Each action is meticulously captured, with each action represented by a comprehensive set of 30 frames. During the training mode, videos are collected under diverse visual scenarios to ensure effective training outcomes.

Once the videos are collected, the frames undergo preprocessing to enhance their quality and standardize features, facilitating subsequent stages of analysis and model development. MediaPipe is used to detect landmarks and keypoints, identifying 21 landmarks for each hand, which capture intricate movements and positions of the fingers, palm, and wrist. Additionally, MediaPipe identifies 33 landmarks for the body pose, including key points on the torso, arms, legs, and joints, and captures the outline of the face, including eyes and mouth. This comprehensive detection enables detailed tracking of human posture and movements.

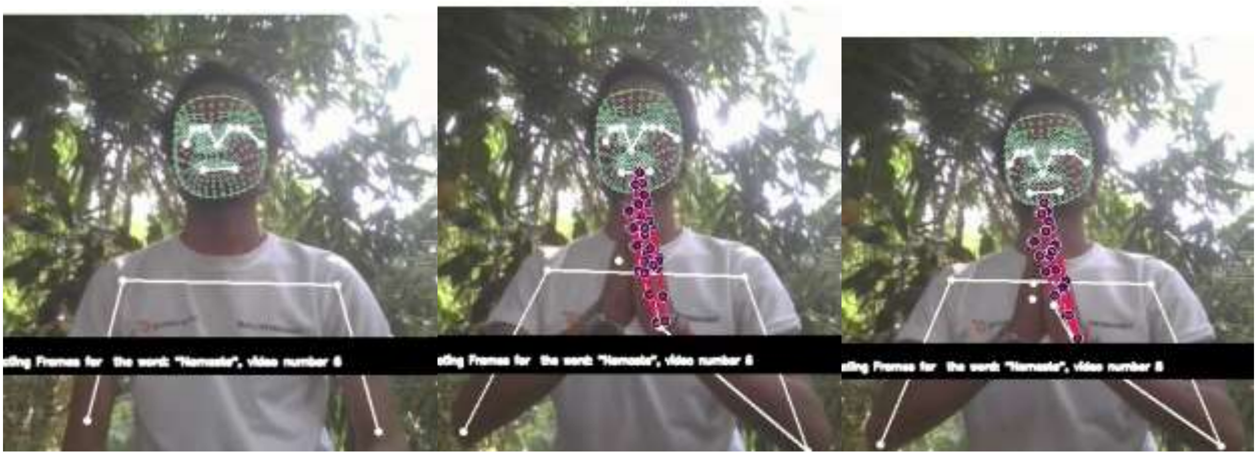


FIGURE 5 Dataset Words Training

Following the detection of landmarks and keypoints, the data is structured into a stacked format for each frame. These stacked keypoints, encompassing the detected landmarks for hands, body, and face, are stored in JSON format. This format is chosen for its ease of readability and accessibility during the detection phase.

The JSON files are saved with the filename structured as the word followed by "_Keypoints". For example, the keypoints for the word "thank you" would be saved as "thank_you_Keypoints.json". This systematic naming convention facilitates easy retrieval and management of the keypoint data.

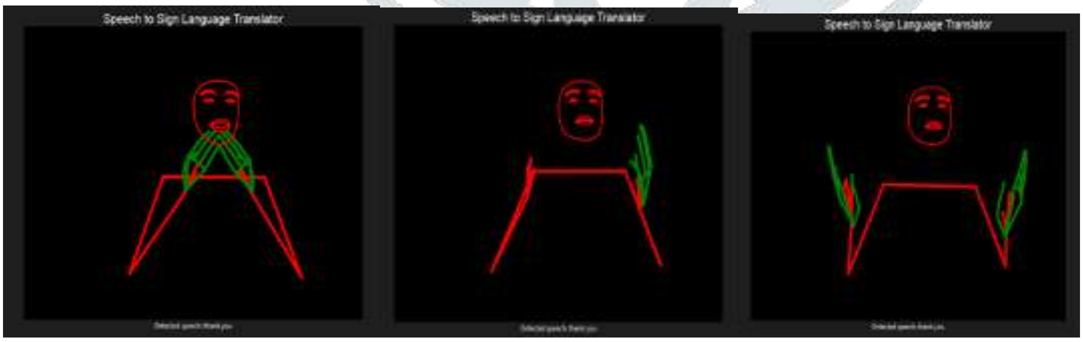
The saved JSON files serve as a robust database for the speech to sign language system. This database, containing detailed keypoint information for each frame, is essential for accurately translating speech into sign language gestures. The database is then deployed into the hardware model, which in this project involves a Raspberry Pi. This deployment enables the system to function efficiently in a real-time environment, ensuring accurate and responsive sign language translation.

5. Results and Discussion

In our study, we developed a speech-to-sign language translation system. The system integrates real-time speech recognition with a database of pre-recorded sign language gestures. This section outlines the system's components, performance, and results. The speech-to-sign language translator uses the Vosk speech recognition model for converting speech to text, Google MediaPipe for detecting and tracking body and hand landmarks, and Matplotlib for rendering 2D animations of sign language gestures based on the detected keypoints. We also integrated a hardware setup for the sign language to speech translator.

In the data collection phase, frames are extracted from a video feed using OpenCV. Each action is represented by a set of 30 frames. These frames are collected to encompass a wide spectrum of visual scenarios, contributing to effective training outcomes. The frames undergo preprocessing to enhance quality and standardize features, facilitating analysis and model development. MediaPipe detects 21 landmarks for each hand and 33 landmarks for the body pose, including key points on the torso, arms, legs, and joints. The keypoints are stored in JSON format, named based on the word followed by "_Keypoints" (e.g., "thank_you_Keypoints.json").

The system was tested with trained words to ensure accuracy and reliability. The performance was evaluated based on the clarity of the rendered sign language animations and the accuracy of the speech-to-text conversion. Below are examples of the 2D animations for the words "Thank You" and "Sorry" showing three key frames for each word.



(a)

(b)

FIGURE 5 2D animations for the detected words (a) Thank You (b) Sorry

These results demonstrate the effectiveness of our speech-to-sign language translator in providing accurate and real-time translations. The combination of advanced speech recognition, precise landmark detection, and efficient rendering of sign language gestures makes this system a valuable tool for bridging communication gaps in various settings.

6. Conclusion

The development and execution of the speech to sign language translation system represent a significant advancement in assistive technology aimed at enhancing communication and accessibility for individuals with hearing impairments or those who use sign language as their primary mode of communication. In this section, we will summarize the key achievements.

- Speech Detection: The core of the project lies in the accurate detection and recognition of speech using offline speech recognition system. By leveraging the capabilities of Google MediaPipe and Matplotlib, we have successfully developed a model have the ability of performing hand and pose landmarks with key frames, enabling real-time interpretation of sign language gestures.

- User-Friendly Interface: The system features a user-friendly interface that facilitates easy interaction and operation. With intuitive controls and feedback mechanisms, users can input sign gestures, receive instant recognition feedback, and hear the corresponding spoken words, enhancing communication and comprehension. Also, the toggle mode feature helps the user to interactively add any new word and train the model on the go.

- Scalability and Flexibility: The project architecture is designed to be scalable and adaptable to accommodate a greater number of sign gestures and vocabulary. With proper training data and model optimization, the system can support an extensive of sign language words and phrases, catering to diverse user needs and preferences.

7. ACKNOWLEDGMENT

The authors acknowledge all researchers who give their discussion and comments on this manuscript.

REFERENCES

[1] Lavanya, Y. N., Rajalakshmi, N. N., & Sumanth, K. (2023). A Novel Approach for Developing Inclusive Real-Time Yoga Pose Detection for Health and Wellness Using Raspberry pi. In Proceedings of the 2023 7th International Conference on Computation System and Information Technology for Sustainable Solutions (CSITSS). IEEE **[2]** Basu, S. 1997. The Investment Performance of Common Stocks in Relation to their Price to Earnings Ratio: A Test of the Efficient Markets Hypothesis. Journal of Finance, 33(3): 663-682.

[2] SpringerLink. (2021). Recognition of Indian Sign Language (ISL) Using Deep Learning Model. Retrieved from SpringerLink.

[3] SpringerLink. (2021). Real-Time Translation of Indian Sign Language to Assist the Hearing and Speech Impaired. Retrieved from SpringerLink.

[4] SpringerLink. (2021). A Spatio-Temporal Framework for Dynamic Indian Sign Language Recognition. Retrieved from SpringerLink.

[5] Journal of King Saud University. (2019). A Signer Independent Sign Language Recognition with Co-Articulation Elimination from Live Videos. Retrieved from Journal of King Saud University.

[6] MDPI. (2021). Translating Speech to Indian Sign Language Using Natural Language Processing. Retrieved from MDPI.

[7] arXiv. (2020). Speak with Signs: Active Learning Platform for Greek Sign Language, English Sign Language, and Their Translation. Retrieved from arXiv.

[8] IJESI. (2018). Labview Based Hand Gesture Recognition for Deaf and Dumb People. Retrieved from IJESI.

[9] Indian Sign Language, A Secondary Course by NATIONAL INSTITUTE OF OPEN SCHOOLING (An autonomous institution under Ministry of Education, Govt. of India) A-24/25, Institutional Area, Sector -62, NOIDA -201309 (U.P.)

[10] https://indiansignlanguage.org/ Online portal for Indian Sign Language Dictionary by RKMVERI, Faculty of Disability Management and Special Education, Coimbatore IEEE.