https://doi.org/10.29070/nfke6755

A study of the Sign language identifying using the OpenCV, MediaPipe, and Scikit-learn modules

Sanvi Agarwal[1]*, Aarjav Jain[2], Darshika Jallan[3], Krishna Agarwal[4]

[1,2,3,4] Class -12th, Sanskriti The Gurukul, Guwahati, Assam, India

[1] Email: sanvismd@gmail.com

[2] Email: aarjav18jain@gmail.com

[3] Email: darshika.jallan@sanskritithegurukul.in

[4] Email: Krishna.rohit3727@gmail.com

Abstract - Sign language is manual communication commonly used by people who are hard of speaking and hearing. These languages use the visual-manual modality to convey meaning, instead of spoken words. Sign languages are expressed through manual articulation using hand gestures, facial expressions, and body language to describe the intended message as well as some non-manual markers. This paper proposes a novel approach to interpreting sign language using the camera of a phone or a laptop breaking the communication barrier between a mute person and a person who does not know sign language. In this approach, the model has been trained to identify some signs using the OpenCV, MediaPipe, and Scikit-learn modules. The hand landmarks from the image data set were extracted using the media pipe module to train the model. The model can identify signs and once the sign has been identified it can play the corresponding sign in the form of audio.

Keywords- Sign language, MediaPipe hand landmark detection, Python, Audio,OpenCv library

- X -

1. INTRODUCTION

The World Health Organization (WHO) stated that approximately 70 million people in the world are deaf-mutes. The World Health Organization also estimates that there are approximately 466 million people worldwide with disabling hearing loss, which is about 6.1% of the world's population. Of these individuals, 34 million are children, and most live in low- and middle-income countries. This emphasises how important it is to break the barrier between us given so much of our population is faced with this communication barrier.

These people depend on sign language as their main means of communication. Sign language uses body language and posture to convey the intended message. Because of a lack of awareness and knowledge of interpreting sign languages, the deaf and mute frequently run into a few serious issues when they communicate with the general public. Here we propose the usage of a Sign Language Interpreter with an integrated camera to convert the signs detected by the camera to text and audio to ease communication between us.

American Sign Language is used in the model as it is one of the widely used sign languages. The interpreter is portable and accessible so that it can be used by anyone and also so that anyone who is not familiar with sign languages can comprehend its usage. Misreading the signing motion might potentially lead to issues in some situations causing unnecessary confusion. Therefore, it can be said that there is a barrier to sign language communication.

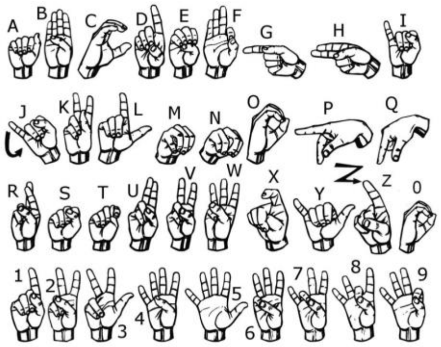


Figure 1: Different signs in the ASL

The interpreter uses a camera to capture the signs shown to it, then uses Google Media Pipe to extract landmarks. This landmark data is processed through the pre-trained train to identify the signs and processes it to convert it into text and audio for the user.

The model proposed here has several advantages ranging from portability to accessibility. The model uses a camera to detect signs shown and then processes it to convert it to audio to ease communication between two users and dismantle the communication barrier.

2. MEDIA PIPE

MediaPipe Solutions provides a suite of libraries and tools to quickly apply artificial intelligence (AI) and machine learning (ML) techniques to various applications. It is a cross-platform pipeline framework to build custom machine-learning model solutions for live and streaming media. The framework was open-sourced by Google and is currently in the alpha stage.

In this project, hand recognition is performed using the Mediapipe hand landmark detection module. The MediaPipe Hand Landmarker task detects the landmarks of the hands in an image. The task locates key points of hands and renders visual effects on them. It operates on image data with a machine learning (ML) model as static data or a continuous stream and outputs hand landmarks in image coordinates, hand landmarks in world coordinates, and handedness (left/right hand) of multiple detected hands.

The hand landmark model bundle detects the keypoint localization of 21 hand-knuckle coordinates within the detected hand regions. The model was trained on approximately 30,000 real-world images, as well as several rendered synthetic hand models imposed over various backgrounds.

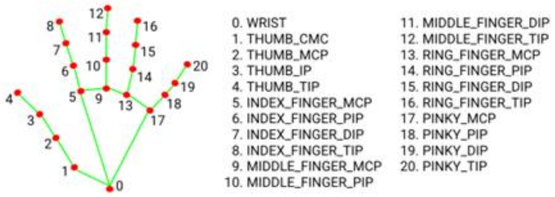


Figure 2: Mediapipe hand landmark detection module.

In our model, we have used media pipe landmarks to train the model instead of image classification and training the model with images of different signs because the media pipe task identifies the hand landmarks quite accurately as the task itself has been trained using a very large dataset. This approach helps the model to work in any background even with a lot of noise in the background.

3. IMAGE COLLECTION

The initial phase of the project involved collecting a dataset of images representing various signs, which were then organised into separate folders. A larger dataset increases the model's efficiency in identifying signs. For this project, 13 signs from American Sign Language (ASL) were chosen. This selection included individual letters such as “A,” “B,” and “C,” as well as commonly used phrases like "hello," "goodbye," and "I Love You."

To create the dataset, 100 images per sign were gathered. This task was facilitated by a local NGO, which provided pictures and videos of different signs from individuals who use sign language in their daily lives.

Figure 3: dataset of images representing various signs

These real-life examples ensured the authenticity and variability needed for an effective dataset. Each set of 100 images was stored in a folder named after the corresponding sign, making it easier to manage and retrieve the data. The data for each of the 20 reference points on the hands will be extracted from these images using the Mediapipe library.

4. DATA EXTRACTION

The Mediapipe hand landmark detection module detects the coordinates of the hand landmarks in the X, Y, and Z axes with reference to each other. For this project, only the X and Y axis data will be used because, in the approach used for this project, the data in the Z axis will be insignificant.

For each sign, all the images will be processed, and the hand landmarks from these images will be extracted using the Mediapipe library. This extracted data will be stored in two different arrays, and the sign name will be extracted from the folder name and stored in a variable. Now, this data will be stored in a final NumPy array along with a label column containing the sign name. NumPy was chosen because the NumPy package offers methods that make data manipulation very easy, as it is the fundamental package for scientific computing with Python.

This process will be repeated for all of the different signs in the collected image dataset. Once all the data is extracted, it is stored in a .pickle file. Using this type of format makes data storage and accessibility during the training process simple and swifter.

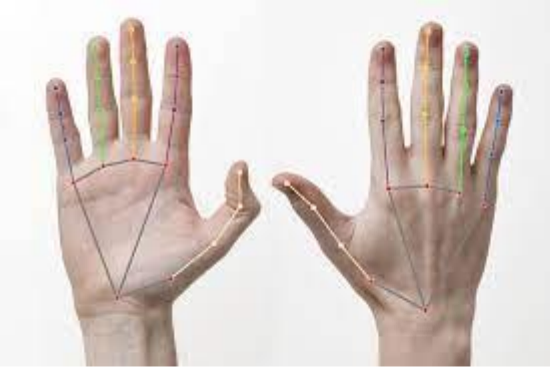


Figure 4: Mediapipe hand landmark detection module detects the coordinates of the hand landmarks in the X, Y, and Z axes

5. TRAINING

To develop a model that converts sign language into audio, the Random Forest algorithm from Scikit-learn will be used to build a robust and accurate machine-learning model for gesture recognition. The data, consisting of numerical arrays collected and stored in a .pickle file, will be used to train the model.

These numerical arrays, each corresponding to a specific sign language gesture, will train the Random Forest algorithm using sci-kit-learn. During the training phase, Random Forest will build numerous decision trees and learn how to categorise gestures according to their unique characteristics. This ensemble approach combines the predictions of all the individual trees to increase accuracy and reduce the risk of overfitting.

After training, any new gesture can be classified in real-time. The conversion of sign language to spoken words is eventually made possible by recognizing these gestures, converting them into text form, and then vocalising them through the TTS (Text to Speech) system. The gesture recognition challenge in any environment always benefits from high-dimensional data with complex and nonlinear interactions. Scikit-learn's Random Forest is well-suited for this purpose, as high-dimensional data with complex and nonlinear relationships align well with the task of gesture recognition.

6. IDENTIFYING THE SIGNS

First, the camera is accessed using OpenCv library and a frame is captured .OpenCV, short for Open Source Computer Vision Library, is an open-source computer vision and machine learning software library. Originally developed by Intel, it is now maintained by a community of developers under the OpenCV Foundation.

This frame which is captured is processed so that the landmarks can be identified using the mediapipe module. From this processed data only the X axis and the Y axis coordinates are used and are stored in an array. From this array the data is taken, once at a time, and is processed through a code which feeds it in the model we had trained.

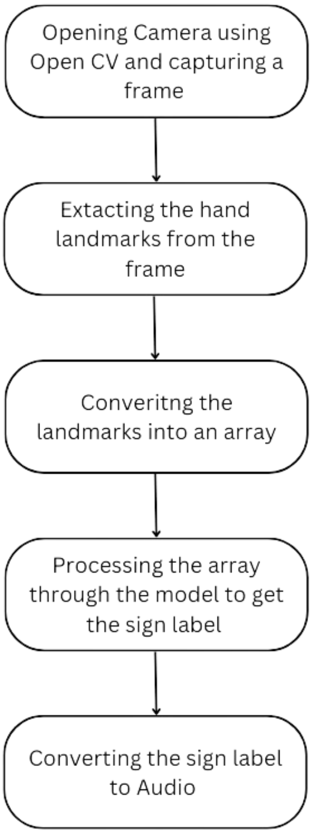

Opening Camera using Open CV and capturing a frame
Extacting the hand landmarks from the frame
Converitng the landmarks into an array
Processing the array through the model to get the sign label
Converting the sign label to Audio

Figure 5: Procedure to identify the hand landmarks using OpenCV

Once, the sign is identified after comparing it with data the model has been trained with, the label containing the sign name is returned. This process is performed again and again continuously so that the signs can be identified in real time using the camera.

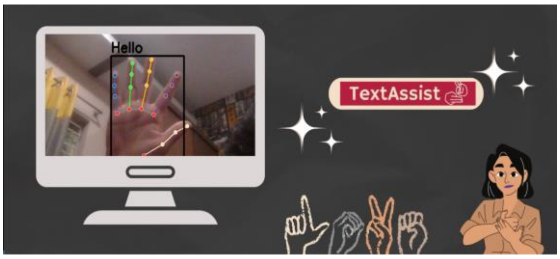


Figure 6: sign is identified after train the data

7. CONVERTING THE SIGNS TO AUDIO

Since reading the signs on the phone screen is not always feasible when the front camera is facing the person showing the signs, and to make the conversation between a person who is hard of hearing and a hearing person feel more like a natural conversation, the text label containing the sign name should be converted into audio. This can be achieved using the PYTTSX3 module.

PYTTSX3 is a text-to-speech conversion library in Python. Unlike alternative libraries, it works offline and is compatible with both Python 2 and 3. The PYTTSX3 module supports two voices: the first is female, and the second is male, which is provided by “sapi5” for Windows. For the current project, we have used the female voice.

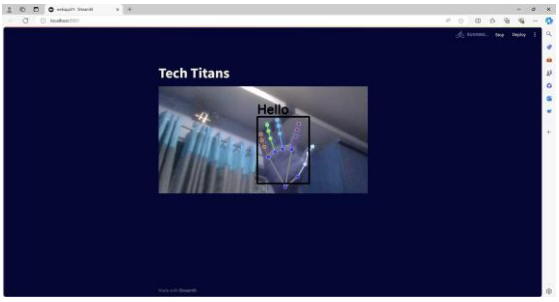
The label obtained after the sign is identified is passed through another method. This method, using the functions of the PYTTSX3 module, converts the sign name into audio.

The process of identifying the signs is performed continuously, capturing 24 frames per second. This means the program will read out the sign 24 times per second, which can be too fast for the human brain to comprehend. Therefore, another method is used that reads the sign out only once for each new sign identified.

8. DEPLOYMENT

After the model was ready, there was a requirement for a user interface to enable the capturing of sign language and convert it to text and audio. The use of Streamlit comes into play here as it is an open-source Python library that makes it possible for developers to rapidly create and share interactive web applications, even without an in-depth understanding of web development. It enabled the construction of a web application for capturing sign language input through a webcam, processing the input through a program accessing the trained model, converting it into text, and producing audio output.

Using Streamlit, a website was made to process the sign language input in real-time, ensuring that the text and audio output was generated quickly and accurately. The interface was also updated instantly to reflect the processed input, providing a seamless and interactive experience for users. A temporary GUI was also created for user testing as it provided a simple and intuitive interface for users to interact. This temporary GUI was designed to gather user feedback and ensure the interpreter worked accurately and efficiently in real-world scenarios.



Tech Titans

Hello

Figure 7: temporary GUI created for user testing

The final deployment of the sign language interpreter will involve creating a website and an app that can be accessed directly from both laptops and phones. This approach ensured a consistent and accessible experience for users, regardless of their device or platform.

9. TESTING

The model can currently identify 13 signs including English alphabets and some commonly used phrases. The interpreter was tested thoroughly and some of the signs interpreted in real time are shown below.

A) Letter ‘A’

When the sign for the character ‘A’ has been shown to the camera. The program first extracts a frame. This frame is then processed so that the position of hand landmarks can be detected using the mediapipe module. Then this extracted data is runned through the model which identifies the sign and returns the sign label. This sign label is then converted into audio and the letter ‘A’ is read aloud .

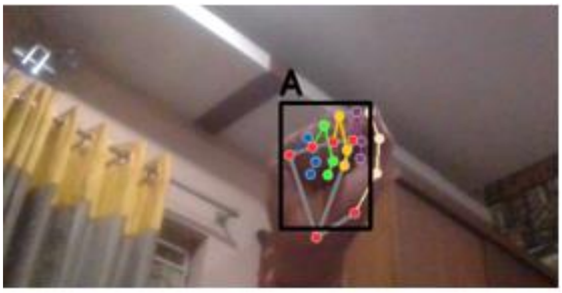


Figure 8: sign converted into audio and the letter ‘A’ is read aloud

B) Phrase ‘I Love You’

Similarly, when the sign for the phrase ‘I Love You’ has been shown to the camera. The program first extracts a frame. This frame is then processed so that the position of hand landmarks can be detected using the mediapipe module. Then this extracted data is runned through the model which identifies the sign and returns the sign label. This sign label is then converted into audio and the phrase ‘I Love You’ is read aloud .

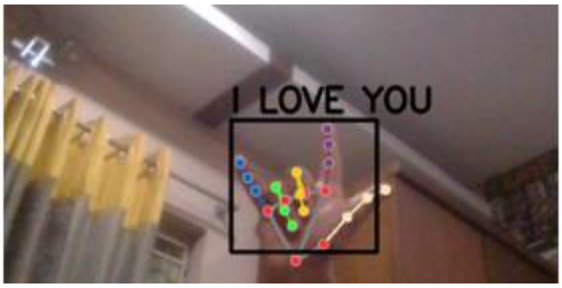


Figure 9: sign label is converted into audio and the phrase 'I Love You'.

10. FUTURE PROSPECTS

The model currently facilitates communication by converting sign language into text and audio. However, there are several future advancements in sign language interpretation that could be implemented. Right now, the interpreter is in the prototype phase and it shows the basic ability to turn sign language into text and audio, but there's a lot of room to make it better and add more features.

The current database contains 100 images for each sign, but the aim is to expand it to 1,000 pictures per sign down the road. This big jump in database size will boost interpreters' ability to accurately spot and grasp various sign language movements making the interpreter more productive and reliable overall. Currently, the model can understand 13 signs, but the dataset can be increased allowing more signs to be interpreted. This will make the interpreter more adaptable and useful.

The current implementation of the sign language interpreter primarily focuses on American Sign Language (ASL), but we can explore the possibility of training the interpreter to recognize and interpret different sign language systems such as Indian Sign Language (ISL), British Sign Language (BSL), Australian Sign Language (Auslan), and others. By doing so, the interpreter will be more inclusive and accessible to a global audience.

Currently, the use of a camera is to watch and figure out signs. But this doesn't always work well because of things like bad lighting in the background or too much noise in the background. Holding a camera and showing signs to it may not be feasible for a person all the time. This can be fixed using alternatives like electromyography (EMG) sensors. These sensors can pick up muscle movements, from the forearm region for instance, which can give a better way to spot signs without the issue of noise or lighting. With the addition of EMG sensors, it can recognize signs better and then turn them into sound. This might work better than using cameras in places where there's a lot of noise or the light keeps changing.

11. CONCLUSION

In this paper , the camera recognition approach suggested to convert the sign language into audio, is a prototype. The prototype is developed to aid deaf and mute people communicate with other people and to break this communication barrier. The main idea of using this camera based approach is to make the interpretation portable and easily accessible. In this approach firstly, a model is trained with some data consisting of different sign gestures. Then, using this model, signs can be identified in real time and converted to audio so that communication barriers can be broken and the deaf and mute people can also move along with this fast moving world .

REFERENCES

[1] Wikipedia contributors. (2024, July 16). Sign language - Wikipedia. https://en.wikipedia.org/wiki/Sign_language

[2] MediaPipe Solutions guide. (n.d.). Google for Developers. https://ai.google.dev/edge/mediapipe/solutions/guide

[3] Hand landmarks detection guide. (n.d.). Google for Developers. https://ai.google.dev/edge/mediapipe/solutions/vision/hand_landmarker

[4] Boesch, G. (2024, June 21). MediaPipe: Google's Open Source Framework (2024 Guide). viso.ai. https://viso.ai/computer-vision/mediapipe/#:~:text=for%20your%20organization.-,What%20is%20MediaPipe%3F,currently%20in%20alpha%20at%20v0

[5] OpenCV. (2024, July 12). OpenCV - Open Computer Vision Library. https://opencv.org/

[6] GeeksforGeeks. (2024, April 15). What is OpenCV Library? GeeksforGeeks. https://www.geeksforgeeks.org/opencv-overview/

[7] GeeksforGeeks. (2022, April 18). Python Text to Speech by using pyttsx3. GeeksforGeeks. https://www.geeksforgeeks.org/python-text-to-speech-by-using-pyttsx3/

[8] pyttsx 3. (2020, July 6). PyPI. https://pypi.org/project/pyttsx3/

[9] Grzejszczak, T., Kawulok, M., & Galuszka, A. (2016). Hand landmarks detection and localization in color images. *Multimedia Tools and Applications*, *75*, 16363-16387.

[10] 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*.

[11] Sánchez-Brizuela, G., Cisnal, A., de la Fuente-López, E., Fraile, J. C., & Pérez-Turiel, J. (2023). Lightweight real-time hand

segmentation leveraging MediaPipe landmark detection. *Virtual Reality*, *27*(4), 3125-3132.

[12] Luna-Jiménez, C., Gil-Martín, M., Kleinlein, R., San-Segundo, R., & Fernández-Martínez, F. (2023, October). Interpreting sign language recognition using transformers and MediaPipe landmarks. In *Proceedings of the 25th International Conference on Multimodal Interaction* (pp. 373-377).

[13] Priya, K., & Sandesh, B. J. (2023, March). Hand Landmark Distance Based Sign Language Recognition using MediaPipe. In *2023 International Conference on Emerging Smart Computing and Informatics (ESCI)* (pp. 1-7). IEEE.

[14] Bora, J., Dehingia, S., Boruah, A., Chetia, A. A., & Gogoi, D. (2023). Real-time assamese sign language recognition using mediapipe and deep learning. *Procedia Computer Science*, *218*, 1384-1393.

[15] Samaan, G. H., Wadie, A. R., Attia, A. K., Asaad, A. M., Kamel, A. E., Slim, S. O., ... & Cho, Y. I. (2022). Mediapipe's landmarks with rnn for dynamic sign language recognition. *Electronics*, *11*(19), 3228.

[16] Luna-Jiménez, C., Gil-Martín, M., Kleinlein, R., San-Segundo, R., & Fernández-Martínez, F. (2023, October). Interpreting sign language recognition using transformers and MediaPipe landmarks. In *Proceedings of the 25th International Conference on Multimodal Interaction* (pp. 373-377).

[17] Remiro, M. Á., Gil-Martín, M., & San-Segundo, R. (2023). Improving Hand Pose Recognition Using Localization and Zoom Normalizations over MediaPipe Landmarks. *Engineering Proceedings*, *58*(1), 69.

[18] Priya, K., & Sandesh, B. J. (2023, March). Hand Landmark Distance Based Sign Language Recognition using MediaPipe. In *2023 International Conference on Emerging Smart Computing and Informatics (ESCI)* (pp. 1-7). IEEE.

[19] 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*.

[20] Subashini, V., Someshwaran, B., Sowmya, S., & Kumar, S. A. (2024, March). Sign Language Translation Using Image Processing to Audio Conversion. In *2024 Third International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS)* (pp. 1-6). IEEE.

Corresponding Author

Sanvi Agarwal*

Class -12th, Sanskriti The Gurukul, Guwahati, Assam, India

Email: sanvismd@gmail.com