International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2024: 7.101

Bridging The Communication Gap: A Machine Learning Approach to Sign Language Recognition

A. Amulya[1], Afifa Butool[2], Sedmaki Aishwarya[3]

[1]Professor, Mahatma Gandhi Institute of Technology, Hyderabad

[2,3]UG Student, Mahatma Gandhi Institute of Technology, Hyderabad

Abstract: ***Sign language is the main and most effective communication tool that is through touch, among the hearing-impaired and the deaf. However, those that do not know it (through sign language) at all, therefore, a communication barrier exists in this situation. This project aims to build a strong system for sign language recognition based on Media Pipe, OpenCV, and Scikit-learn. Hand tracking in Media Pipe is used for visual hand detection and extraction of the pupil, which represents hand motion accurately and discriminately. OpenCV serves to offer comprehensive image processing such as scaling, normalization, and image data augmentation. This is done only after the preprocessing operation has ensured data best practices for training. The features that were acquired are then used to train the implemented machine learning models in Scikit-learn, for instance, Random Forests or Support Vector Machines for the classification of gestures.***

Keywords: Hearing-impaired, Media Pipe, OpenCV, Random Forest, Gesture, Normalization

1. Introduction

In enormous research conducted in diverse domain names, it turned into determination that impairments such as hearing-impairment, vocal-impairment or the ineptitude to express oneself causes loss of opportunities for such people when compared to able people. Not only does it lead to this but also hinders day-to-day activity of an individual such as normal conversations. According to MoSPI, Govt. of India [1], in 2002 about 30.62 lakh of the then population were suffering from hearing disorder and 21.55 lakh of the then population were suffering from speech disorder. Another 2001 Census [2] states that around 21 million Indian citizens (which constituted 12.6 million males and 9.3 million females approximately), that is, about 2.1 per cent of the then population of India, were facing certain disabilities. People with speech disability accounted for the 7.5 per cent while those with hearing disability accounted for 5.8 per cent of these 21 million people in total. These statistics also show evidence of the problems and discrimination faced by these people. Additionally, they also provide us with a wealth of facts about specific kinds of disabilities, the number of humans tormented by these disabilities and the barriers they face in their life. One of the foremost boundaries a disabled Individual faces in his existence is incapable of talking with an everyday man or woman. So, with our knowledge of technology, we hope to help such people through our project so that they are able to communicate normally with others.

2. Literature Survey

Sign language recognition using a Modified-LSTM Model for continuous sign language recognition employs a Leap Motion device for capturing hand movements. This model achieves average accuracies of 72.3% for signed sentences and 89.5% for isolated sign words. By improving the learning process through increased training data, the recognition performance can be further enhanced. The model effectively handles continuous sign language recognition while maintaining reasonable accuracy with improved data efficiency [1].

Another approach, Hand Gesture Recognition for Sign Language Using 3DCNN, adopts a 3D convolutional neural network that outperforms four out of six other state-of-the-art methods in performance evaluations. Despite its accuracy, this model does not currently support live video feeds, limiting its real-time application potential [2].

A Deep Cascaded Model for video-based isolated hand sign language recognition enhances accuracy and processing speed in uncontrolled environments involving rapid hand motions. While the model effectively handles fast-moving gestures, expanding the dataset could significantly improve detection accuracy [3].

The Depth-Based Indian Sign Language Recognition Using Microsoft Kinect method achieved an impressive average recognition accuracy of 71.85% with 100% accuracy for some signs. However, this model struggles with environmental inconsistencies, occasionally resulting in incorrect translations [4].

In contrast, a Signer Independent Sign Language Recognition Model incorporates coarticulation elimination techniques to enhance recognition from live videos. This economical model supports mobile cameras, improving accessibility for general users. However, its performance declines in cluttered backgrounds or poor lighting conditions [5].

The Efficient Binarized Neural Network for recognizing two-hand Indian sign language gestures in real-time delivers an exceptional accuracy of 98.8%, outperforming many conventional approaches. However, the system occasionally misclassifies similar-shaped gestures like M, N, and E. Additionally, the model is restricted to a limited number of gesture classes [6].

A Deep Learning-Based Indian Sign Language Recognition System by Sruthi C. J and Lijiya A (2021) leverages deep learning techniques to recognize Indian Sign Language gestures with impressive accuracy. The system achieved a remarkable training accuracy of 99.93% and a testing and validation accuracy of 98.64%, indicating its robustness in handling various hand gestures during controlled conditions. This model's high accuracy demonstrates its potential for real-world applications such as educational tools for the hearing-impaired or communication aids in professional environments. Integrating facial recognition and context analysis could significantly improve the model's overall performance and reliability in practical scenarios [7].

The American Sign Language (ASL) Recognition Using RF Sensing system developed by Sevgi Z. Gurbuz et al. (2021) presents an innovative approach to recognizing sign language gestures without requiring a camera or visual input. Instead, the system utilizes radio frequency (RF) sensing technology to detect hand movements and gestures, making it a viable solution for contactless ASL recognition. This method is particularly effective in ASL-sensitive smart environments, where visual input may be obstructed or unavailable. Moreover, the system demonstrates remarkable robustness by functioning efficiently even in low-light or dark conditions, a significant advantage over traditional vision-based systems. Developing this extensive dataset is essential for refining the system's accuracy and ensuring meaningful interpretations in real-world settings [8].

3. System Architecture

The module in fig.3.1 is mainly about processing data, training, and categorizing movements. It takes the raw data of images from storage and applies some preprocessing methods, like resizing, normalization, and augmentation. The module also carries out training of a machine-learning model or deep learning model for gesture recognition by the model.py class that contains the architecture. The train_model.py script utilizes the pre-processed data to train the model which will then be available in the Models database for classification later. The module further performs gesture recognition with user-given inputs, and the test_data.py script processes the input through the trained model that is stored in the Models database. The system, in turn, creates an output prediction that identifies the user's gesture.

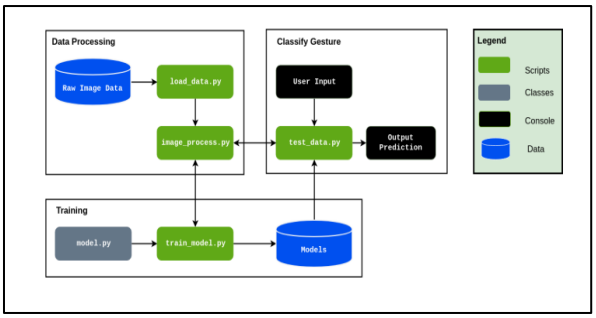

Figure 3.1: Architecture of the translation model

4. Implementation

4.1 Gathering Dataset

The first step in any machine learning problem is to gather the data. The data can either be taken from some opensource datasets from websites such as Kaggle or you can prepare your own dataset. In our case, we created our own dataset from scratch. For the data gathering process, we took x- and y-coordinates of 42 hand key points (21 for each hand) using the Media Pipe and OpenCV libraries. For each gesture, the following x and y key points were collected:

- Wrist
- Thumb
- Index finger
- Middle finger
- Ring finger
- Pinky finger

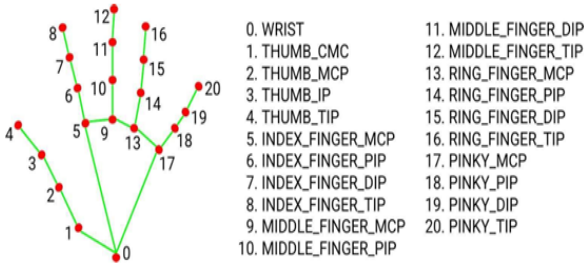

Figure 4.1.1: Hand Landmarks

Figure 4.1.1 shows the 21 hand landmarks or key points as we are considering them in our project. Each of the points above represents a key point which is a key factor in deciding the gesture. While joining the key points we consider both x-axis and y-axis coordinates for better understanding and better mapping. For our system, we have collected few frequently used hand gestures from the American Sign Language as out Dataset.

Figure 4.1.2: Sample Dataset

4.2 Creating the Training Data:

Table 4.2.1: Sample values from Dataset

	0	3400	2600	4200
CLASS	a	r	n	v
WRIST.x	0.519844	0.648427	0.410579	0.614857
WRIST.y	0.631178	0.940012	0.724817	0.878059
THUMB_CMC.x	0.438433	0.568895	0.329412	0.53715
THUMB_CMC.y	0.598432	0.936431	0.714727	0.847619
THUMB_MCP.x	0.379219	0.509991	0.282891	0.502831
THUMB_MCP.y	0.494121	0.836858	0.588538	0.772382
THUMB_IP.x	0.365754	0.550817	0.330221	0.596222
THUMB_IP.y	0.388703	0.742219	0.46443	0.770321
THUMB_TIP.x	0.350171	0.603992	0.367002	0.678162
THUMB_TIP.y	0.305574	0.6626	0.379596	0.737245
INDEX_FINGER_MCP.x	0.416051	0.523074	0.259804	0.528556
INDEX_FINGER_MCP.y	0.393216	0.664789	0.47045	0.560293
INDEX_FINGER_PIP.x	0.390594	0.513809	0.212547	0.494616
INDEX_FINGER_PIP.y	0.370394	0.537262	0.450001	0.411368
INDEX_FINGER_DIP.x	0.402981	0.506091	0.230641	0.472861
INDEX_FINGER_DIP.y	0.449733	0.445233	0.551565	0.314119
INDEX_FINGER_TIP.x	0.416766	0.501942	0.258426	0.458727
INDEX_FINGER_TIP.y	0.509701	0.365516	0.619552	0.218974
MIDDLE_FINGER_MCP.x	0.468968	0.578839	0.322361	0.597762
MIDDLE_FINGER_MCP.y	0.378663	0.63971	0.438229	0.561901
MIDDLE_FINGER_PIP.x	0.447741	0.538673	0.278005	0.622411
MIDDLE_FINGER_PIP.y	0.376232	0.524995	0.387824	0.417274
MIDDLE_FINGER_DIP.x	0.466829	0.506799	0.28388	0.64093
MIDDLE_FINGER_DIP.y	0.481974	0.436372	0.501866	0.31755
MIDDLE_FINGER_TIP.x	0.484939	0.487269	0.307072	0.659745
MIDDLE_FINGER_TIP.y	0.554076	0.362243	0.569292	0.22418
RING_FINGER_MCP.x	0.527718	0.639073	0.395642	0.651466
RING_FINGER_MCP.y	0.382022	0.651393	0.430524	0.598375
RING_FINGER_PIP.x	0.505762	0.623854	0.374539	0.684657
RING_FINGER_PIP.y	0.378085	0.652502	0.406852	0.638069
RING_FINGER_DIP.x	0.518674	0.616305	0.369548	0.665789
RING_FINGER_DIP.y	0.482942	0.768619	0.54245	0.730897
RING_FINGER_TIP.x	0.53095	0.614971	0.378459	0.647558
RING_FINGER_TIP.y	0.554198	0.842028	0.600908	0.772569
PINKY_MCP.x	0.588687	0.695562	0.466656	0.687371
PINKY_MCP.y	0.397633	0.683408	0.440557	0.652676
PINKY_PIP.x	0.566689	0.667793	0.443367	0.703334
PINKY_PIP.y	0.379503	0.702079	0.44557	0.690174
PINKY_DIP.x	0.560907	0.650071	0.43254	0.684527
PINKY_DIP.y	0.453977	0.793078	0.533916	0.755706
PINKY_TIP.x	0.562171	0.641488	0.435452	0.665548
PINKY_TIP.y	0.509735	0.856231	0.555947	0.795826

For creating the training data, we took x- and y-coordinates of 21 hand key points using the Media Pipe and OpenCV libraries. For each gesture, the following x and y key points were collected: wrist (WRIST), thumb (THUMB_CMC, THUMB_MCP, THUMB_IP, THUMB_TIP), index finger (INDEX_FINGER_MCP, INDEX_FINGER_PIP, INDEX_FINGER_DIP, INDEX_FINGER_TIP), middle finger (MIDDLE_FINGER_MCP, MIDDLE_FINGER_PIP, MIDDLE_FINGER_DIP, MIDDLE_FINGER_TIP), ring finger (RING_FINGER_MCP, RING_FINGER_PIP, RING_FINGER_DIP, RING_FINGER_TIP) and pinky (PINKY_MCP, PINKY_PIP, PINKY_DIP, PINKY_TIP).

4.3 Hardware and Software Interfaces

1) Visual Studio Code: Visual Studio Code is a source code editor made by Microsoft for Windows, Linux and MacOS. Most popular programming languages are supported in Visual Studio Code on a basic level. This fundamental support consists of configurable snippets, code folding, bracket matching, and syntax highlighting.

2) Media Pipe: The Media Pipe Framework is used to create machine learning pipelines for processing time-series data, including audio, video, and other types. This cross-platform Framework works in Desktop/Server, Android, iOS, and embedded devices. Media Pipe Toolkit comprises the Framework and the Solutions. Media Pipe Hands employs machine learning to deduce 21 3-dimensional landmarks of the hand using just a single frame.

3) OpenCV: OpenCV (Open-Source Computer Vision Library) is an open-source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being an Apache 2 licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

5. Results

When the user displays a sign, the model processes the hand frame and displays the output. The model works on multiple hand gestures whose dataset is stored and trained in the system.

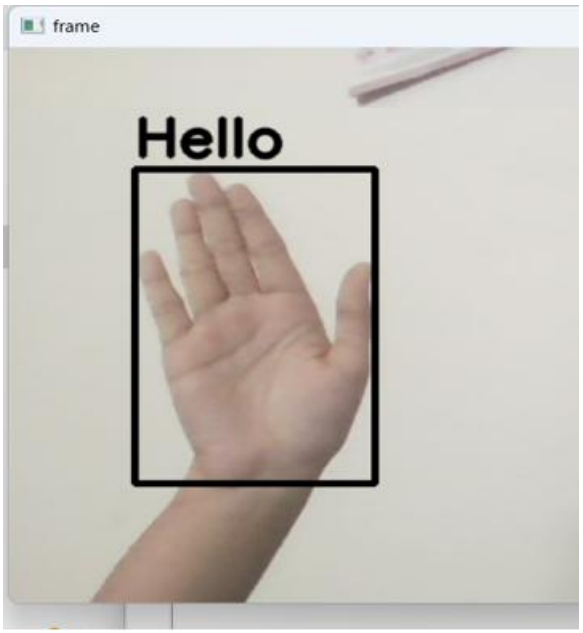

Figure 5.1: Application working on gesture ‘Hello’

Figure 5.2: Application working on gesture 'Emergency'

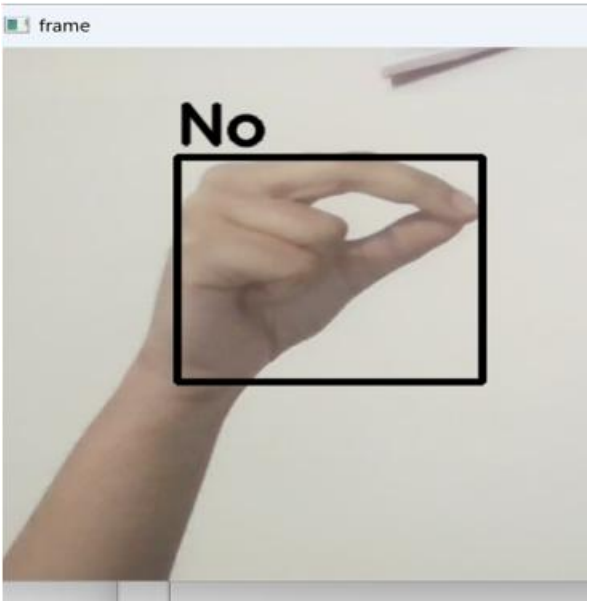

Figure 5.3: Application working on gesture 'No'

6. Conclusion and Future Scope

The Sign Language Recognition project utilizing OpenCV and Python has demonstrated promising capabilities in translating manual gestures into meaningful digital representations. Through the implementation of computer vision techniques, we have successfully captured and processed real-time hand gestures, enabling the recognition of sign language expressions. The use of Media Pipe and OpenCV for sign language recognition represents a significant advancement in the accessibility of communication for the deaf and hard-of-hearing communities. By leveraging Media Pipe's hand-tracking capabilities along with OpenCV's image processing, this project effectively recognizes and classifies hand gestures for alphabets and numbers, facilitating basic communication without the need for complex hardware. This project's success in accurately interpreting diverse sign gestures highlights the potential for technology to bridge communication gaps and enhance accessibility for individuals with hearing impairments. The robustness of the OpenCV library, coupled with Python's flexibility, has proven effective in creating a reliable and efficient system.

The future work that can improve our proposed work is listed as follows:

1) Understanding and applying a better algorithm to account for background noise and better separation of the foreground from the background.
2) Extend the model to recognize dynamic gestures or continuous sign language phrases, not just static signs.
3) Comparing the results when Transfer Learning is used instead of basic Neural Networks.

References

[1] "Dynamic hand gesture recognition of Arabic sign language by using deep convolutional neural networks."-Ismail M.H., Dawwd S.A., Ali F.H.(2023)
[2] Data Glove with Bending Sensor and Inertial Sensor Based on Weighted DTW Fusion for Sign Language Recognition" - Lu, C, Amino S,Jing L(2023)
[3] "An Efficient Two-Stream Network for Isolated Sign Language Recognition Using Accumulative Video Motion", H. Luqman, (2022)
[4] "Development of an End-to-End Deep Learning Framework for Sign Language Recognition, Translation, and Video Generation", B. Natarajan et al., (2022)
[5] "Machine learning methods for sign language recognition: A critical review and analysis. "- Ankita Wadhawan, Parteek Kumar (2021)
[6] Z. R. Saeed, Z. B. Zainol, B. B. Zaidan and A. H. Alamoodi, "A Systematic Review on Systems-Based Sensory Gloves for Sign Language Pattern Recognition: An Update From 2017 to 2022,"(2022)
[7] B. Joksimoski et al., "Technological Solutions for Sign Language Recognition: A Scoping Review of Research Trends, Challenges, and Opportunities,"(2021)