DOI: https://doi.org/10.52756/ijerr.2024.v41spl.002 *Int. J. Exp. Res. Rev.*, Vol. 41: 11-18 (2024)

Original Article | Peer Reviewed | Open Access

International Journal of Experimental Research and Review (IJERR)
© Copyright by International Academic Publishing House (IAPH)
ISSN: 2455-4855 (Online)
www.iaph.in

Enhancing Sign Language Understanding through Machine Learning at the Sentence Level

Ch Sekhar[1], Juthuka Aruna Devi[2], Mirtipati Satish Kumar[3], Kinthali Swathi[4], Pala Pooja Ratnam[4] and Marada Srinivasa Rao[5]*

Check for updates

[1]Department of Computer Science and Engineering (AI&ML), GMRIT, Rajam, Andhra Pradesh, India; [2]Department of Computer Science and Engineering, Vignan's Institute of Information Technology, Visakhapatnam, Andhra Pradesh, India; [3]Department of Computer Science and Engineering, Centurion University of Technology and Management, Vizianagaram, Andhra Pradesh, India; [4]Department of Computer Science and Engineering, Gayatri Vidya Parishad College of Engineering (A), Visakhapatnam, Andhra Pradesh, India; [5]Department of Information Technology, MVGR College of Engineering, Vizianagaram, Andhra Pradesh, India

E-mail/Orcid Id:

CHS, sekhar.ch@gmrit.edu.in, https://orcid.org/0000-0001-5603-1453; *JAD,* jarunadevi2003@gmail.com, https://orcid.org/0009-0004-0003-7257; *MSK,* satishkumarsrk@icloud.com, https://orcid.org/0009-0000-2920-7812; *KS,* swathi@gvpce.ac.in, https://orcid.org/0009-0006-2695-8227; *PPR,* poojaratnam@gvpce.ac.in, https://orcid.org/0009-0005-4451-178X; *MSR,* srinivas.marada22@gmail.com, https://orcid.org/0000-0001-6849-0286

Article History:

Received: 26th Apr., 2024
Accepted: 13th Jul., 2024
Published: 30th Jul., 2024

Keywords:

Sign Language, Feature Extraction, Machine Learning, Random Forest

How to cite this Article:
Ch Sekhar, Juthuka Aruna Devi, Mirtipati Satish Kumar, Kinthali Swathi, Pala Pooja Ratnam and Marada Srinivasa Rao (2024). Enhancing Sign Language Understanding through Machine Learning at the Sentence Level. *International Journal of Experimental Research and Review*, *41*(spl.), 11-18.
DOI: https://doi.org/10.52756/ijerr.2024.v41spl.002

Abstract: The visual language of sign language is based on nonverbal communication, including hand and body gestures. When it comes to communicating, it is the main tool for those who are deaf or hard of hearing all around the globe. Useful for both hearing and deaf persons, this can translate sign language into sentences in real-time or help those who are hard of hearing communicate with others. This work focuses on developing a sentence-level sign language detection system utilizing a custom dataset and Random Forest model. Leveraging tools such as Media Pipe and TensorFlow, we facilitate gesture detection. Through continuous detection of gestures, we generate a list of corresponding labels. These labels are then used to construct sentences automatically. The system seamlessly integrates with ChatGPT, allowing direct access to generate sentences based on the detected gestures. Our custom dataset ensures that the model can accurately interpret a wide range of sign language gestures. Our method helps close the communication gap between people who use sign language and others, with an accuracy of 80%, by merging machine learning with complex language models.

Introduction

Human existence is not possible without communication. The exchange of information and expertise is facilitated via communication. People are able to express themselves more freely and form new relationships as a result. Those who are unable to communicate via their native tongues, such as the deaf and dumb, encounter constant challenges to Yavanamandha et al. (2023). Verbal language, which includes speaking, reading, and writing, and nonverbal language, which includes facial expressions and sign language, play crucial roles in communication. Because of this, the only option for the Deaf and the Dumb is to communicate through "Sign language" stands for "non-verbal communication by Tambuskar et al. (2023).

The majority of individuals who do not have any kind of hearing or speech impairment communicate primarily through vocalisations and other forms of spoken language. Nevertheless, those who are deaf or have speech impairments rely on non-verbal means of communication, mostly signs and gestures, to convey their thoughts and feelings. Consequently, the common

*Corresponding Author: srinivas.marada22@gmail.com

people and those who are deaf or hard of hearing use separate channels for communication by Godage et al. (2021). This obstacle prevents the two groups of people from effectively communicating with one another.

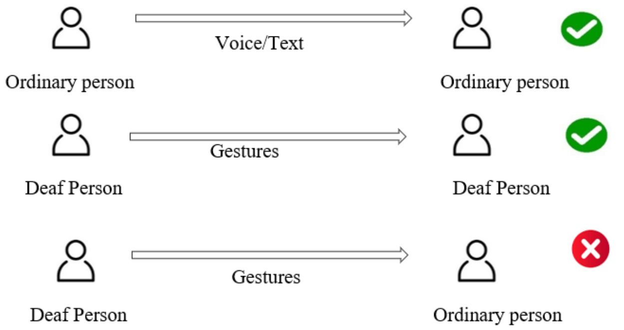

Figure 1. Communication methods.

It is difficult for a deaf person and a hearing person to communicate, as seen in Figure 1. Those who are deaf rely on sign language while hearing people communicate either vocally or through text. Both of the prerequisites for fruitful dialogue have already been touched upon. Both hearing-impaired and hearing-normal people can communicate with one another in this way by Gireesh Babu and Thungamani (2022). Czczmc Yet, the second requirement of utilising a shared communication platform is not met by deaf people communicating with hearing people. Pictured above is an effort to communicate by sign language, which the deaf can comprehend but the hearing impaired cannot. Failure to utilise a common communication platform results in their communication falling flat by Kasapbaşi et al. (2022).

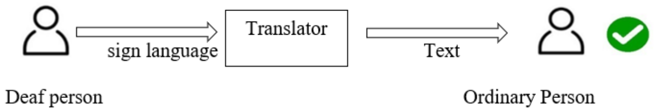

Figure 2. The communication method of the proposed solution.

Figure 2 demonstrates our system's ability to translate sign language gestures into text or voice for non-signers to understand easily. Our focus is on translating continuous sign language sentences into Text, enabling communication between deaf individuals and non-signers.

Existing System

Existing methodologies for sign language detection encompass a diverse range of approaches, each addressing the unique challenges of gesture recognition and communication for individuals with hearing impairments. One such approach involves leveraging YOLOv5 (Jayakumar and Peddakrishna, 2024), a lightweight and efficient deep learning architecture, for gesture detection. Training and evaluating the model on the MU Hand Images ASL dataset achieves high precision (95%), recall (97%), and mean Average Precision scores 98%, indicating its suitability for real-time gesture recognition tasks. YOLOv5's effectiveness stems from its efficient architecture, incorporating modern computer vision techniques and state-of-the-art activation functions, hyperparameters, and data augmentation techniques by Rao et al. (2022). Its small size and low power consumption make it an ideal choice for use on low-end devices like mobile phones, expanding the audience for sign language recognition software by Dima and Ahmed (2021) and Rao et al. (2023). Another notable approach involves the development of a wearable device capable of translating sign language into speech and text. This device integrates sensors, including accelerometers and flex sensors, to capture hand and finger movements corresponding to American Sign Language (ASL) gestures. Through the interaction of these sensors, specific ASL gestures are recognized and translated into speech and text using a mobile phone application. Early trial results demonstrate the feasibility of the device, with an average translation time of 0.6 seconds for converting sign language into speech and text, showcasing its potential for practical application in real-world scenarios by Abougarair (2022). Sign language recognition also makes use of deep learning methods like Gated Recurrent Units (GRU) and Long Short-Term Memory (LSTM). Modelling sign language movements is a good fit for these architectures because they can capture long-term interdependence in sequential data. We use various datasets and preprocessing techniques to train the model to make them more accurate. After that, we used evaluation metrics to see how well the model performed. We also see that LSTM and GRU have great promise for making sign language recognition systems more accurate by Chakraborty et al. (2023).

Proposed System

Our proposed system aims to develop a robust hand sign detection system leveraging Python, Media pipe, OpenCV, and Scikit Learn while incorporating a Random Forest model sentence-level prediction. To achieve this, we will start by collecting a diverse dataset comprising labelled hand sign images paired with associated sentences to facilitate supervised learning. We will enhance image quality and extract relevant features for model input by preprocessing techniques utilising Media pipe and OpenCV. Subsequently, the Random Forest model will be trained using Scikit Learn, utilizing the extracted features and corresponding sentence labels. The model's performance will be evaluated using various metrics, such as accuracy, with cross-validation

DOI: https://doi.org/10.52756/ijerr.2024.v41spl.002

techniques to ensure robustness. The model will be seamlessly integrated into the system upon successful training to enable real-time hand sign detection and sentence prediction. For sentence prediction, we are accessing the chat GPT to accurately generate the automated sentence using the list of gesture labels that have been detected.

Related Work

The use of hand movements in sign language is crucial for persons who have problems hearing or speaking, whether they are deaf or not. Sign language systems are important but not necessarily user-friendly or cost-effective. A model that recognises sign language automatically will help deaf and hard of hearing people communicate with society. This model teaches a convolutional neural network to extract features from static images with ten samples per sign. Images are often processed to find fingertips and convert them to text. Musthafa and Raji (2022) demonstrate that the Sign Language Recognition System can do real-time picture recognition by identifying sign language gestures during testing.

Alsaadi et al. (2022) suggested a system for real-time Arabic Sign Language that takes video captured by a camera and feeds it into the system. The hand was tracked in the video frames using a Haar-like algorithm. The region of interest was extracted using preprocessing techniques such as skin identification and size normalisation. After converting the images to the frequency domain, the feature vectors are obtained by applying Fourier Transform to the resulting images. An accuracy of 90.55% was attained by the system when employing the k-Nearest Neighbour (KNN) algorithm for classification.

Gangadia et al. (2020) proposed a method to aid communication for the hearing and speech impaired, particularly focusing on Indian Sign Language (ISL). The system aims to recognize ISL gestures in real-time using a Hybrid-CNN model and feature extraction techniques. It converts gestures to text and speech, facilitating efficient communication. Additionally, it employs a Rule-Based Grammar and Web Search query for generating sentences, augmented by a Multi-Headed BERT grammar

Table 1. Comparison of Existing models.

Sl. NO	Author Name	Methodology	Limitations	Accuracy
1	Tambuskar et al., 2023	CNN	does not address dynamic gestures	95%
2	Bhagat et al., 2019	EMG and IMU	Data availability, model complexity	80%
3	Dima and Ahmed, 2021	YOLOV5	Data availability	94%
4	Alsaadi et al., 2022	flex sensor, MIT, smart glove	Letter distinction, physical constraints, prototype focus.	80%
5	Chakraborty et al., 2023	LSTM and GRU (RNN)	Limited dataset, user variations.	85%
6	Kasapbaşi et al., 2022	CNN	Feature extraction, environmental conditions.	82%
7	Ilanchezhian et al., 2023	KNN	Feature extraction, hardware-free, recognition rate.	90.55%
8	Kasapbaşi et al., 2022	CNN, RNN	Spatial-temporal data, model complexity	95.21%
9	Godage et al., 2021	HMM	User-independent system, feature complexity.	91%
10	Gangadia et al., 2020	Hybrid-CNN	Feature complexity, data variety.	90%
11	Musthafa and Raji, 2022	CNN	Skin tone, lighting conditions.	83%
12	Raval and Gajjar, 2021	CNN, LSTM	Data synthesis, transfer learning	96%-CNN 98%-LSTM

DOI: https://doi.org/10.52756/ijerr.2024.v41spl.002

corrector for accurate outputs. This approach addresses the absence of a suitable communication medium for individuals with speech and hearing disabilities, providing a practical solution for effective interaction.

Raval and Gajjar (2021) addressed the need to recognize sign language gestures to facilitate communication for individuals with speech impairments. It combines image processing and machine learning to develop a real-time system capable of recognizing hand gestures. Image processing is utilized to pre-process images and isolate hands from the background. The Convolutional Neural Network (CNN) is trained on a dataset containing 24 English alphabet gestures and tested on both custom and real-time data, achieving an accuracy of 83%.

Bhagat et al. (2019) proposed a real-time hand gesture recognition system utilizing Microsoft Kinect RGBD camera developed for speech-impaired communication. Computer vision techniques facilitated accurate segmentation of gestures from background noise. Convolutional Neural Networks (CNNs) achieved 98.81% accuracy on 36 Indian Sign Language (ISL) static gestures, while Convolutional LSTMs reached 99.08% accuracy on 10 dynamic ISL word gestures. The model showcased adaptability by achieving 97.71% accuracy in recognizing American Sign Language (ASL) gestures through transfer learning. This system presents promising potential for enhancing gesture-based communication for speech-impaired individuals.

Deaf and mute people could converse better with gesture detection and translation, according to Ilanchezhian et al. (2023). The goal is to interpret camera-captured hand motions. The system grasps sign language expressions by analysing finger configuration, hand orientation, and relative locations. LabelImg, a Python object detection tool, collects and labels sign pictures. Using these photographs, a model is trained using the Tensor Flow object detection API. The system is able to detect and show the interpreted sign language on-screen by accessing the webcam using OpenCV-python and loading the trained model. This allows for real-time sign language recognition.

Materials and Methods

The proposed model for detection of sign and their corresponding sentence framing is explained in this section. The step-by-step detection process of hand gestures and sentence framing is highlighted in Figure 3.

Step 1: The process starts with capturing an image of a hand gesture using a camera or webcam.

Step 2: The captured image undergoes preprocessing steps to enhance its quality and prepare it for gesture recognition.

Step 3: Features relevant to hand gestures are extracted from the pre-processed image. These features might include hand position, shape, and movement.

Step 4: The extracted features are used to recognize and classify the hand gestures present in the image. Techniques such as machine learning algorithms, specifically Random Forest in this case, are applied for

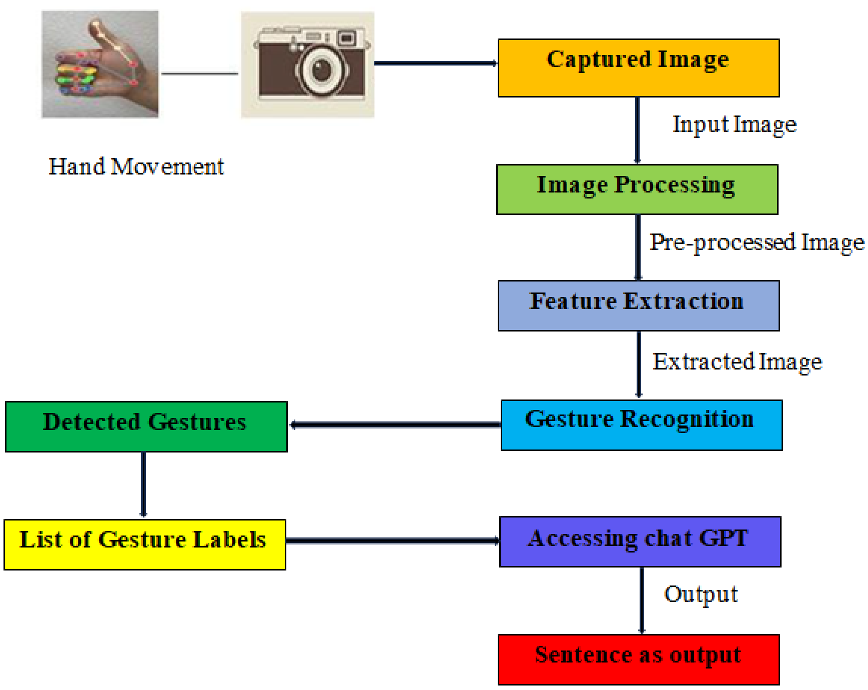

Figure 3. Process of Sentence Level Sign Language Detection.

DOI: https://doi.org/10.52756/ijerr.2024.v41spl.002

gesture recognition.

Step 5: Once recognized, the system identifies the specific gestures present in the image. Each gesture is associated with a unique sign language label representing a word or phrase.

Step 6: The recognized gestures are mapped to a list of corresponding labels. These labels represent the meaning conveyed by each gesture.

Step 7: The list of labels is passed to ChatGPT, an AI language model. ChatGPT generates natural language sentences based on the detected gestures.

Step 8: ChatGPT constructs coherent sentences in natural language using the provided labels. The sentences convey the meaning of the gestures captured in the image.

As a last step, choose the predicted outcome that received the most votes.

GPT-3.5 Integration

Integrating GPT-3.5 into a Python application involves steps.

Sign up for OpenAI API Access:

If you have not already, sign up for access to the OpenAI API and obtain your API key from the OpenAI website. Securely Store Your Account Credentials.

The secret key needs to be kept secret! Otherwise, other people can use it to access the API, and you will pay for it.

i. Locate Integrations in your workspace and click on it.

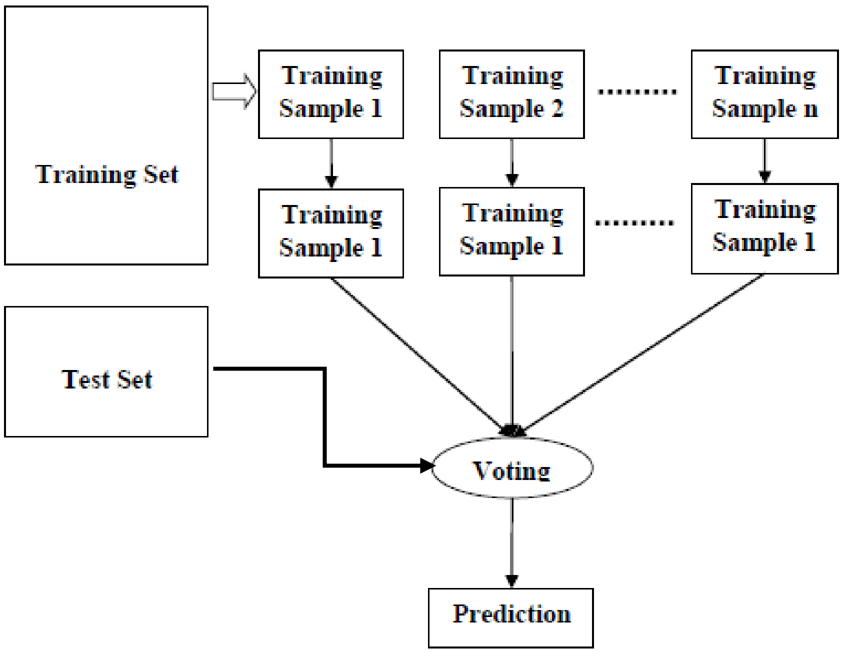

Figure 4. Random Forest Model.

Algorithm

Random forest models are like a team of decision-makers working together to understand sign language gestures. They're good at recognizing different hand movements and shapes, even when things get messy or complicated. They're used in apps or devices that help people communicate through sign language, ensuring they can understand and respond to gestures accurately and quickly.

Here are the steps shown in Figure 4:

- The first step is randomly choosing a subset of the data or training set.
- Its second step is to build a decision tree for each training data set.
- Third, the decision tree will be averaged before voting.

ii. To add an integration, choose "Create integration" (+).

iii. Choose a "Environment Variables" integration.

iv. Enter "OPENAI" into the "Name" column. Copy and paste your secret key v into the "Value" field.

v. Choose "Create" and then link the new integration.

Setup an OpenAI Developer Account: To use the API, you need to create a developer account with OpenAI. You'll need to have your email address, phone number, and debit or credit card details handy.

Make API Requests: Use the functions provided by the OpenAI SDK to make requests to the GPT-3.5 API. By utilizing methods like openai. Completion.create(), developers can send prompts to GPT-3.5 and receive text completions.

DOI: https://doi.org/10.52756/ijerr.2024.v41spl.002

Handle the API Response: Process the response returned by the API call. The response contains the generated text or other relevant information:

generated_text = response.choices[0].text.strip()

print(generated_text)

Test and Iterate: Test your integration thoroughly and iterate on your code as needed to optimize performance and address any issues.

Deploy Your Application: Once you're satisfied with your integration, deploy your Python application to your desired environment.

Monitor API Usage: Monitor your OpenAI API usage to ensure you stay within your usage limits and consider implementing rate limiting or caching mechanisms if necessary.

Result and Discussion

This section highlights various test cases by considering the real images to the developed application. All those test cases and generated sentences are highlighted.

Test case 1:

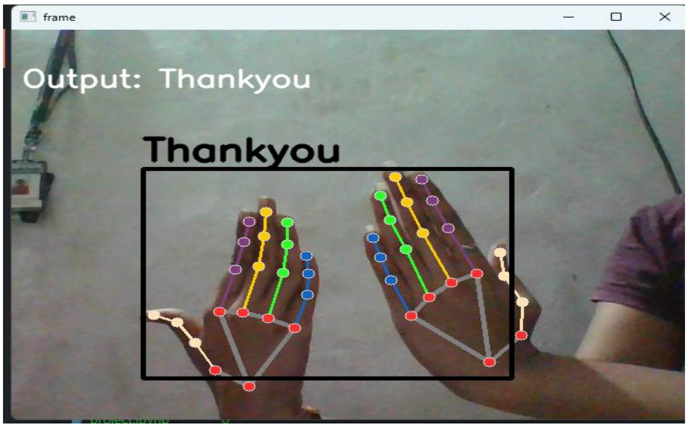

Figure 5. Thank you gesture is detected.

In Figure 5, we are trying to detect hand gesture "Thank you" with the help of the media pipe, once it is detected properly the respective label of the detected gesture will be displayed on the frame.

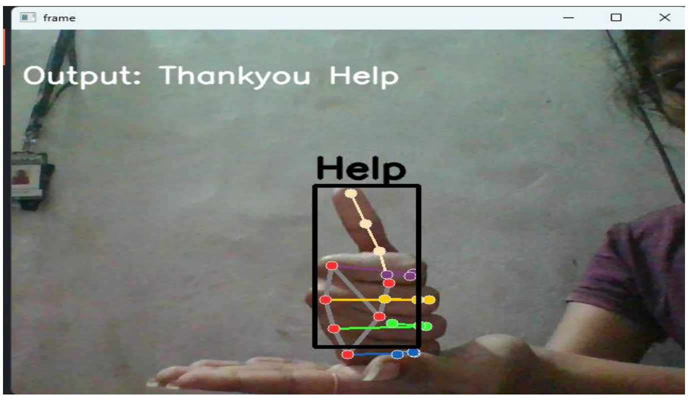

Figure 6. Help gesture is detected

In Figure 6, we are trying to detect another hand gesture "Help", once it is detected properly label will be displayed in the frame.

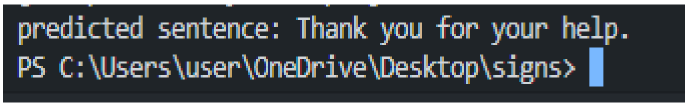

Figure 7. Displaying sentence for corresponding detected words

In Figure 7, AI will take words as prompts and generate sentences accordingly.

Test case 2:

Figure 8. Detecting hand gesture "please".

In Figure 8, we are trying to detect hand gesture "please" with the help of the media pipe, once it is detected properly it is displayed above automatically. After detecting the word properly, we will further detect other gestures.

Figure 9. Detecting Hand Gesture "call".

In Figure 9, we are trying to detect another hand gesture "call" with the help of the media pipe, once it is detected properly, it is displayed above automatically.

Figure 10. Detecting hand gesture "Me"

In Figure 10, we detect the hand gesture "me" with the help of the media pipe. Once it is detected properly, it is displayed above.

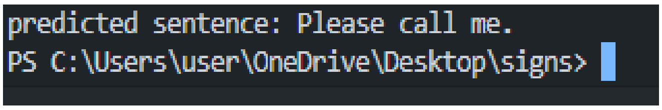

Figure 11. Generating Sentence.

In Figure 11, As Sign language gestures are detected, a related sentence is generated with the help of Gemini AI. In this work, we achieved an accuracy of 80%, which is not up to the mark of real-time application. This model is useful for dumb and deaf people in regular life, like in educational institutions, hospitals, working institutions etc. The limitation of this work is that we have experimented with this model for sample regular sentences, not for all sentences generally used by dumb and deaf people.

Conclusion and Future Scope

This work demonstrates the feasibility and potential of integrating machine learning techniques with computer vision for real-time hand gesture recognition and sentence translation. By leveraging custom datasets, the Random Forest model, and the Media Pipe library, we have successfully developed a system capable of accurately detecting and translating word gestures into sentences. This work holds significant promise in enhancing communication accessibility for individuals with hearing impairments and facilitating intuitive human-computer interaction. We have achieved 80% accuracy using random forest model.

Future scope for this work includes exploring more sophisticated machine learning models, such as deep neural networks, for improved gesture recognition accuracy. Integration of natural language processing techniques could enable the system to handle more complex sentence structures and improve translation accuracy. Additionally, enhancing the system's robustness to variations in lighting conditions, backgrounds, and hand orientations would be beneficial for real-world deployment. Moreover, expanding the vocabulary and supporting multiple languages could broaden the system's applicability and reach. Overall, continued research and development in this area hold immense potential for advancing assistive technologies and human-computer interaction paradigms.

Conflict of Interest

There is no conflict of Interest

References

Abougarair, A. J. (2022). Smart glove for sign language translation. *International Robotics & Automation Journal*, *8*(3), 109–117. https://doi.org/10.15406/iratj.2022.08.00253

Alsaadi, Z., Alshamani, E., Alrehaili, M., Alrashdi, A. a. D., Albelwi, S., & Elfaki, A. O. (2022). A real time Arabic Sign Language Alphabets (ARSLA) recognition model using deep learning architecture. *Computers*, *11*(5), 78. https://doi.org/10.3390/computers11050078

Bhagat, N. K., Vishnusai, Y., & Rathna, G. N. (2019). Indian Sign Language Gesture Recognition using Image Processing and Deep Learning. In *2019 Digital Image Computing: Techniques and Applications (DICTA), Perth, WA, Australia*. https://doi.org/10.1109/dicta47822.2019.8945850

Chakraborty, S., Paul, P., Bhattacharjee, S., Sarkar, S., & Chakraborty, A. (2023). Sign language recognition using Landmark Detection, GRU and LSTM. *American Journal of Electronics & Communication*, *3*(3), 20–26. https://doi.org/10.15864/ajec.3305

Dima, T., & Ahmed, M. E. (2021). Using YOLOv5 Algorithm to Detect and Recognize American Sign Language. *International Conference on Information Technology (ICIT), 2021*, 603-607. https://doi.org/10.1109/icit52682.2021.9491672

Gangadia, D., Chamaria, V., Doshi, V., & Gandhi, J. (2020). Indian Sign Language Interpretation and Sentence Formation. In *2020 IEEE Pune Section International Conference (PuneCon), Pune, India. 2020*, 71-76. https://doi.org/10.1109/punecon50868.2020.9362383

Gireesh Babu C N, Thungamani M (2022). Effective communication between differently abled and normal people using speech to sign translation system. *Journal of Pharmaceutical Negative Results*, *13*(9), 1078–1084. https://doi.org/10.47750/pnr.2022.13.s09.129

Godage, I., Weerasignhe, R., & Sandaruwan, D. (2021). Sign language recognition for sentence level continuous signings. *Natural Language Processing*. https://doi.org/10.5121/csit.2021.112305

Ilanchezhian, P., Singh, I. a. K., Balaji, N., Kumar, A. M., & Yaseen, S. M. (2023). Sign language detection using machine learning. In *Lecture notes in Electrical Engineering*, pp. 135–143. https://doi.org/10.1007/978-981-19-7126-6_11

Jayakumar, D., & Peddakrishna, S. (2024). Performance Evaluation of YOLOv5-based Custom Object Detection Model for Campus-Specific Scenario.

International Journal of Experimental Research and Review, *38*, 46-60. https://doi.org/10.52756/ijerr.2024.v38.005

Kasapbaşi, A., Elbushra, A. E. A., Al-Hardanee, O., & Yılmaz, A. (2022). DeepASLR: A CNN based human computer interface for American Sign Language recognition for hearing-impaired individuals. *Computer Methods and Programs in Biomedicine Update*, *2*, 100048. https://doi.org/10.1016/j.cmpbup.2021.100048

Musthafa, N., & Raji, C. G. (2022). Real time Indian sign language recognition system. *Materials Today: Proceedings*, *58*, 504–508. https://doi.org/10.1016/j.matpr.2022.03.011

Raval, J. J., & Gajjar, R. (2021). Real-time Sign Language Recognition using Computer Vision. In *2021 3rd International Conference on Signal Processing and Communication (ICPSC), Coimbatore, India*, *2021*, 542-546. https://doi.org/10.1109/icspc51351.2021.9451709

Rao, M., Kumar, S., & Rao, K. (2023). Effective medical leaf identification using hybridization of GMM-CNN. *International Journal of Experimental Research and Review*, *32*, 115-123. https://doi.org/10.52756/ijerr.2023.v32.009

Rao, M. S., UmaMaheswaran, S. K., Sattaru, N. C., Abdullah, K. H., Pandey, U. K., & Biban, L. (2022). A critical understanding of integrated artificial intelligence techniques for the healthcare prediction system. 2022, *2nd International Conference on Advance Computing and Innovative Technologies in Engineering* (ICACITE), *2022*, 728-731. https://doi.org/10.1109/icacite53722.2022.9823678

Tambuskar, H., Khopde, G., Ghode, S., Deogirkar, S., & Vaidya, E. M. (2023). A Survey Paper on Sign Language Recognition System using OpenCV and Convolutional Neural Network. *International Journal for Research in Applied Science and Engineering Technology*, *11*(2), 245–249. https://doi.org/10.22214/ijraset.2023.49007

Yavanamandha, P., Keerthana, B., Jahnavi, P., Rao, K. V., & Kumar, C. R. (2023). Machine Learning-Based Gesture Recognition for Communication with the Deaf and Dumb. *International Journal of Experimental Research and Review*, *34*(Special Vol.), 26–35. https://doi.org/10.52756/ijerr.2023.v34spl.004

How to cite this Article:

Ch Sekhar, Juthuka Aruna Devi, Mirtipati Satish Kumar, Kinthali Swathi, Pala Pooja Ratnam and Marada Srinivasa Rao (2024). Enhancing Sign Language Understanding through Machine Learning at the Sentence Level. *International Journal of Experimental Research and Review, 41*(spl.), 11-18.

DOI : https://doi.org/10.52756/ijerr.2024.v41spl.002