Sign Language Recognition for Hearing-Impaired People

1st **Shamitha S Hegde**
Department of MCA,
Mangalore Institute of Technology and Engineering,
Karnataka, INDIA
shamithahegde06@gmail.com

2nd **Ragesh Raju**
Assistant Professor,Department Of MCA,
Mangalore Institute of Technology and Engineering,
Karnataka, INDIA
ragesh@mite.ac.in

***Abstract*—Communication gaps remain between hearing and deaf populations despite major technology developments, especially when reliable, real-time sign language communication is required. Effective communication is hindered by the availability, speed, and naturalness of traditional means like text-based communication and human translators. The objective for this research is to develop a dependable and accurate Sign Language Detection system which can convert gestures through written language in real-time, thereby overcoming this major communication barrier. The suggested system processes video or sensor data and uses sophisticated machine learning as well as computer vision to identify hand gestures, face expressions, and body language. OpenCV, which processes images and videos, and MediaPipe, which accurately tracks hands and recognizes gestures, are important technologies. With the application using advanced image processing algorithms, data augmentation, adaptive learning, and resilience, the framework contains engineered to function dependably in a variety of lighting and environment scenarios.**

To figure out the complex patterns of sign language movements, algorithms for machine learning, such as Random Forest Classifiers (RFC), K-Nearest Neighbors (KNN) as well as CNNs or convolutional neural networks are used. KNNs give a simple method for categorizing gestures based on closeness to known examples, RFCs offer strong ensemble learning approaches to increase prediction accuracy, and CNNs are especially good at extracting features and learning geographic structures.

The phases within the mechanism design are as follows: gathering data, preprocessing, extracting features, training the model, and evaluating it. People who are fluent in the target sign language provide video recordings of their motions. Preprocessing involves breaking up video data into frames, resizing and normalizing images, and using noise reduction techniques. Accurate hand landmarks, such as fingertips and finger joints, are provided by MediaPipe, which is utilized for hand segmentation and landmark recognition. These symbols are necessary in order to gesture detection.

Following preprocessing, models for machine learning undergo training on preprocessed facts for guarantee accurate recognition of sign language gestures. Characteristics such as remembering, reliability, exactness, and F1 score are employed to evaluate the model's performance, while cross-validation has become used to guarantee consistency. The ultimate purpose is that framework aims to offer a simple user experience that makes it easier for hearing and deaf persons have conversations with others encouraging accessibility and equality.

This research helps to generate a more connected community by resolving the technical issues and guaranteeing the system's responsiveness to real-world settings. By offering a dependable, automatic solution for real-time sign language interpretation, the proposed SLR system improves communication and the quality of life to assist the hard of hearing population.

Keywords—- Sign Language Recognition, Real-Time Interpretation, Computer Vision, Machine Learning, OpenCV, MediaPipe, Convolutional Neural Networks, K-Nearest Neighbors, Random Forest Classifier

I. INTRODUCTION

Communication gaps between hard of hearing and hearing population still remain despite enormous technical developments, especially when accurate, instant gestures interpretation is necessary. Regarding accessibility, rapidity, and naturalness of touch, traditional approaches—like text-based communication and human interpreters—often fall short. A dependable, automatic system that can swiftly convert written through the sign words.

The fundamental purpose of this study aims at construct a robust and accurate SLR system capable of translating gestures to written translation within actual time . This mission addresses a large communication barrier between deaf and hearing individuals by using advanced technology.

The main objective is to build a prototype that could quickly and accurately recognize and interpret real-time motions using gestures. Machine learning as well as computer vision are needed to process video or sensor data that includes hand movements, facial expressions, and body language. For example, OpenCV suitable to image and video processing tasks like background reduction and feature extraction, while the MediaPipe library suitable to accurate and efficient hand tracking and gesture recognition. Since achieving real-time performance is the aim, there shouldn't be any discernible lag in the system's capacity that remain attached in a real discussion concurrently.

Sign language recognition systems need to work reliably in a range of scenarios and lighting conditions. The major objective here is to make the system strong enough to handle these variances without compromising accuracy. Adaptive learning, data augmentation that raise its diversity of the training dataset, and advanced image processing with OpenCV are among techniques it might be applied to ensure accurate results. Furthermore, by efficiently addressing several different environmental situations, models for machine learning like k-Nearest Neighbors (KNN) and Random Forest Classifier from the scikit-learn toolbox can be utilized to increase the robustness of the system.

During the system's design phase, the end users should be taken into consideration. It should be simple to use for both those who are deaf and getting input from those persons who might not be familiar with it. This could mean having an interface that makes communication easy, having directions that are easy to follow, and having a simple setup process. By learning patterns of space CNNs can find patterns in the data greatly increase precision of gesture detection, improving the system's dependability and user-friendliness. The technology needs for being produced readily available and safe to ensure that be widely used.

By increasing equality and availability, the creation of an efficient sign language recognition system might greatly improve communication for the deaf community. Through addressing the technical challenges and guaranteeing the system's flexibility to actual situations, this study makes a positive impact regarding building something more united and responsive society.

II. Related Work

Hope Orovwode et all. [1] implemented CNN for alphabet sign recognition in ASL or American-Sign Language. Three convolutional layers and a SoftMax output layer make up the CNN model. The categorical cross-entropy loss function and the Adam optimizer works for its compilation. CNN's conceptual framework is trained using pre-processed pictures taken with the dataset.

I.A. Adeyanju et all. [2] discusses multiple algorithms utilized to recognize gestures with specific emphasis on the DeepConvLSTM model, including CNNs and Neural Networks with Recurrent Architectures with Long-Short Term Memory (LSTM). This model is noted for its effectiveness in handling the temporal dynamics of sign language sequences.

Prof. Radha S. Shirbhate et all.[3] employs SVM for the identification of signs. SVM is taught using a set of data where skin segmentation is first performed using a dataset from UCI. Feature extraction is done using methods like SIFT and HU's moments before feeding information inside the SVM for classification.

S.Saravana Kumar et all. [4] used is a SVM. The SVM framework processes video feeds frame by frame, identifying hand contours to classify sign language symbols. The framework is practiced using a dataset from ASL or American-Sign Language alphabets mapped to their English equivalents. The SVM is implemented using the scikit-learn library, and OpenCV is used for image processing to extract the contours from the video frames.

Mahalakshmi V et all. [5] has implrmented RFC, SVM, and K-Nearest Neighbors (KNN) as its three primary model training approaches. The model is trained using these algorithms to identify motions in sign language from video input that is recorded using a camera. The video data is processed by OpenCV and hand relevant factors are forecast by means of MediaPipe.

Akshatha Rani K et all. [6] has applied Artificial Neural Network (ANN) architecture is used to classify the alphabets used in gesture communication. Collected images are manually tracked by the system and preprocessed using MediaPipe before being sent into the ANN model for training and classification.

Table 1.Comparing existing systems.

Paper	Algorithms used	Accuracy
[1] Hope Orovwode et all.	Convolutional Neural Network algorithm	CNN algorithm gives best accuracy
[2] I.A. Adeyanju et all.	DeepConvLSTM model, which integrates CNN Recurrent Neural Networks with Long-Short Term Memory (LSTM).	DeepConvLSTM model has the best accuracy
[3] Prof. Radha S. Shirbhate et all.	SVM algorithm	For a general dataset, the accuracy of SVM is 53.23%
[4] S.Saravana Kumar et all.	SVM algorithm using a Hidden Markov Model	The accuracy rate is high when SVM method used Hidden Markov Model
[5] Mahalakshmi V et all.	RFC, SVM, and K-Nearest Neighbors (KNN) algorithms	The precision that the sign language recognition system is not specifically stated but it says Random Forest has the proportion of correct predictions in the test data.

III. Methodology

The method of the research includes feature extraction, preprocessing, data gathering as well as ML techniques for gesture communication identification. Video recordings of individuals who were skilled in the target sign language were used to gather data, capturing several different types of motions.

During preprocessing, video content was separated into frames, images were resized and normalized, and noise reduction methods were used. To separate hand areas and extract important features, Mediapipe was employed in the segmentation of hands and landmark recognition.

In order to recognize gestures, feature extraction concentrated on getting accurate hand landmarks, such as fingertips and fingers. Models for machine learning used these features as inputs.

Machine learning techniques, like Random Forest Classifiers (RFC), K-Nearest Neighbors (KNN), and Convolutional Neural Networks (CNNs), were used in the model training process. The preprocessed dataset was used to train and verify these models to ensure precise recognition of sign language motions.

The block diagram of the device is displayed in Fig.1, which also shows the different phases and parts within the system, such as collection of data, preprocessing of data, feature extraction, model training, and evaluation. This graphic representation shows the project's development and flow.

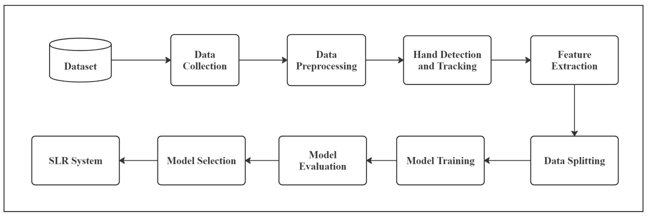

Fig. 1. **System Block Diagram**

A. *Dataset*

Acquiring a set of data for the identification of sign language involves using publicly available datasets or creating your own dataset. Publicly available datasets provide pre-existing videos and images using gestures. Alternatively, anybody are able to produce their own dataset. by recording videos of people performing sign language gestures, ensuring good lighting and background conditions, and accurately labeling the data.

B. *Data Collection*

A sign language recognition system's data collection process involves a number of essential processes designed to acquire a huge and accurate dataset of motions. Setting goals for the recognition system, specifying the sign language and gestures that will be included, and defining the scope and objectives are the first actions throughout the process. After that, participants who are fluent in the target sign language are chosen, making sure that a variety of signing styles are captured.

High-quality cameras, maybe improved by depth sensors, are employed to grab the gestures within a regulated setting with sufficient lighting and low outside noise. To allow for essential variances, each gesture is performed many times by the participants.

The following stage is to add labels to the gathered videos that match the particular indicators being executed. This annotation procedure, which frequently makes use of specialized software tools, guarantees that each gesture is consistently and accurately labeled.

C. *Data Preprocessing*

Data preprocessing in sign language recognition involves transforming raw video data into a format suitable for model training. This includes extracting frames, normalizing and resizing images, optionally converting to grayscale, reducing noise, segmenting the hand region, and extracting features like hand landmarks. Data augmentation increases diversity, while label encoding converts gesture labels into numerical form.

D. *Hand Detection and Tracking*

Hand tracking and detection are essential functions of a framework for recognizing sign language that allow the system to easily evaluate and categorize hand gestures. For these kinds of jobs, two popular libraries are Mediapipe and OpenCV.

Mediapipe provides pre-trained algorithms and APIs that they are specifically designed to track and detect hands. Mediapipe's strong and effective algorithms enable it accurately recognize hands in video frames and follow their movements over time. Mediapipe offers complete details about the shape and structure of the hand by identifying and extracting hand landmarks or keypoints from the identified regions, including the palm, fingers, and fingertips. The method for identifying gestures benefits greatly from having this data as input attributes.

However, OpenCV is a flexible library that is frequently used for tasks related to computer vision, such as tracking and hand identification. A variety of features and algorithms are available in OpenCV for hand detection, including optical flow and Kalman filters for monitoring hand movement across video frames, as well as HOG (Histogram of Oriented Gradients) descriptors and Haar cascades for hand detection. OpenCV is a flexible and powerful tool which improves Mediapipe's hand monitoring and identifying capabilities.

When combined, Mediapipe and OpenCV offer strong capabilities for hand tracking and detection in systems for identifying a symbol, allowing accurate hand gesture interpretation and efficient sign language communication.

E. *Feature Extraction*

Feature extraction defines that it is a process of extracting the important features from video frames into a system for recognization so which makes use of sign language motions can be accurately represented. Using tools like Mediapipe, this method involves detecting and separating hands, locating important landmarks, and extracting temporal and spatial data like hand motion and geometric arrangements. For these kinds of jobs, OpenCV and Mediapipe are vital since they provide precise landmark detection and hand tracking. Before the extracted features are utilized as input for ML methods are first normalized and, if necessary, decreased in dimensionality. In general, feature extraction is necessary to convert unprocessed video data transformed into usable information effectively utilized for accurate gesture recognition in sign language.

F. *Data Splitting*

Data splitting a system for recognizing sign language refers to partitioning the dataset into training and testing subsets. The set of testing validates the model's capacity for generalization on unobserved data, whereas the training dataset, including most data, is utilized to teach the model patterns and relationships. Methods such as stratified splitting or random splitting guarantee representative subgroups. Libraries such as scikit-learn in Python provide useful functions for data splitting implementation. To recognize and categorize the signs, the built system additionally made use of RFC algorithm from the scikit-learn package. All things considered, appropriate data separation guarantees objective model testing and efficient performance evaluation in sign language recognition systems.

G. Model Training

In the implemented system, model training incorporates the application of machine learning techniques such as CNNs, K-Nearest Neighbors (KNN), and Random Forest Classifier (RFC) to learn patterns and associations between hand gesture features and their corresponding labels. Here's how model training typically works for each algorithm:

1.Convolutional Neural Networks (CNN): Firstly, the CNN's architecture was created expressly for image-based work, with a particular focus on hand motion recognition. Typically, this architecture consists of layers called pooling layers that reduce the dimension of space and convolutional layers that collect characteristics from images.

The input data is prepared with care, including labels that indicate the gesture being done for the hand gesture photographs or extracted attributes such as hand landmarks.

Following that, machine learning methods like backpropagation and optimization algorithms like Adam or Stochastic Gradient Descent (SGD) are employed to instruct the CNN model. Over several cycles, the performance of the model gradually improves as it gains the capacity for correlate input characteristics together with the labels that go with them during training.

Tuning the hyperparameters is an essential phase in maximizing the way in which the CNN. To identify the combination that produces the greatest results, this means trying out different factors including learning rate, batch size, and network architecture.

To evaluate the performance of the model and avoid overfitting, validation is executed using an independent dataset. This guarantees that the system is not just learning the examples used in training but also performs well when applied to new data.

Lastly, the trained CNN's accuracy in identifying sign language motions is examined utilizing a separate testing set. This step offers information on the performance of the model and reliability additionally evaluates how effective it is in the actual world. All things considered, the CNN is a potent tool for precisely deciphering hand movements and enabling successful communication for those who utilise the sign language in a machine learning-based system of gestures.

2.KNN (K-Nearest Neighbors): The precision and robustness of the gesture system are improved by combining KNNs with Convolutional Neural Networks (CNNs).

Convolutional Neural Networks (CNNs) are a great tool for tasks like understanding of gestures made with hands in sign language due to their exceptional ability to extract complex patterns and characteristics from photos or videos. They might, however, miss the minute distance relationships between these characteristics, which are essential for exact classification.

Once feature representations are extracted by CNN, KNNs is activated. KNN compares newly made gestures to feature vectors of previously recorded gestures using the similarity principle. The correlations in space between the many items that the CNN collected are represented by these characteristics dimensions.

KNN locates the gestures that are most similar to recent input gesture by locating the k-nearest neighbors in the feature space. The positional details recorded in the feature vectors are considered in this comparison, which the CNN alone might probably not capable of completely represent.

The name on the new input gesture is decided by KNN using a majority voting process after the closest neighbors have been found. Accordingly, the label given to the new gesture is determined by utilizing the label that is most often adopted by its closest neighbors.

Overall, by taking into consideration positional details that CNN alone could miss, merging KNN with CNN improves accuracy in gesture identification. Through the utilization of CNN's feature extraction powers and KNN's space sensitivity, the system builds a strong foundation for accurately detecting a variety of signs language gestures.

3.Random Forest Classifier(RFC): The Random Forest Classifier (RFC) is a important component of this systems. This is a last classification stage and improves the total effectiveness that the system through the use of detailed feature representations that are extracted by CNNs and may be modified by K-Nearest Neighbors (KNN).

To ensure that extract rich and advanced feature representations that capture the key patterns and properties of each gesture, CNNs first process the input photos or videos of sign language gestures. These characteristics function as a three-dimensional visual illustration of hand gestures and forms utilized in a sign language.

KNN ablility can be applied to analyzing the distances between the extracted features to further enhance these features. New gestures can be classified according to how similar they are to known gestures in the feature space. This step focuses on the similarities and differences between movements, which improves the features' ability to discriminate.

The RFC serves as the final classifier once the features have been collected additionally refined. During the training stage of ensemble learning, RFC builds several decision trees. Every decision tree analyzes random subsets of features and is trained on a distinct random subset for the instructional data. The variation that this randomly brings to the trees is essential to the grouped ability to capture various parts of the data.

With the "bagging" or bootstrap aggregating technique used by the RFC, every tree is instructed using a distinct random portion of the input. This method ensures that every tree captures distinct patterns and changes in the data, hence reducing overfitting and improving the model's ability to adapt.

The RFC uses an entire set of decision-making trees to classify new gestures in the prediction phase. Based on the attributes of the new gesture, each tree independently predicts the future. Following the collection of these predictions by the RFC, the majority vote of the trees determines the final categorization. By using a majority vote, the final choice is guaranteed to be solid and less sensitive to the mistakes made by individual trees. There are various benefits connected to the RFC's ensemble character. First off, by merging the outcomes from several decision trees—each of which captures a distinct component of the data—accuracy is increased. With this diversity, the RFC is guaranteed to have the ability to recognize a greater variety of motions than it would occur in a single model. Second, because the collective decision-making reduces the impact of any one noisy or misleading feature, the RFC's robustness against faulty data is improved. Finally, the RFC gains more strength in its capacity to generalize to unknown data, which makes it a good fit for real-world scenarios where input data may differ.

H. Model Evaluation

To guarantee the precision and dependability of the model, a system for recognizing sign language must go through several important phases of model review. Three groups within the dataset are separated: test, validation, and training. Performance is evaluated using range of criteria, including accuracy, reliability, memory, and F1 score. Cross-validation gets used for verify consistency between several data divisions. We watch out for both overfitting and underfitting to make sure the model fits the information accurately. Throughout training, the performance of the model is regularly verified, and the set of tests is utilized in the final assessment. Error detection and comprehension are aided by methods such as ROC(Receiver Operating Characteristics) curves and confusion matrix analysis. Ultimately, the model is improved by real-world testing and a feedback loop, which strengthens its flexibility and conditional adaptability.

I. Model Selection

In order to guarantee maximum efficiency, a sign language recognition system's model selection process requires multiple crucial processes. First, specific standards and objectives are established, including robustness, efficiency, and correctness. The strengths of CNNs, KNNs, and RFCs in particular—feature extraction, simplicity, and robustness—are taken into consideration while evaluating proposed models. Normalization, augmentation, and dataset division into test sets, validation, and training are the steps taken to prepare the dataset. To guarantee consistent performance, each model is trained and validated, and hyperparameters are adjusted and cross-validation is applied.

Every model is evaluated using performance criteria such as processing time, accuracy, precision, recall, F1 score, and calculation time. These criteria are used to compare models and choose the best one. To ensure objectivity, the selected model is put through additional testing on an alternative test set.

The illustration is integrated into the system and speed and size optimized after final selection. In order to keep the model accurate and effective, it is continuously improved depending on input from the real world.

When selecting a model in gesture recognition, one must assess potential models according to predetermined standards, train and validate them, compare their results, and select the optimal model for integration and ongoing optimization.

J. Software Language Recognition(SLR) System

This system can function in batch or real-time mode after completing data collection, preprocessing, hand detection, feature extraction, data splitting, model training, evaluation, and selection. Using OpenCV and MediaPipe, it records video frames, preprocesses them by resizing and filtering them, and then uses these to identify and track hands. CNN has been set up to recognize key patterns in the movements extracts features. For gesture recognition, the collected features are next input into a chosen classification model, like a Random Forest Classifier. Smoothing is one of the post-processing strategies utilized to enhance the predictions for stable recognition. The gestures that are identified are output as speech, text, or actions. Through the incorporation of user feedback and the update of the prototype with fresh data, the framework is continuously improved. Lastly, it is ready for deployment across a range of hardware platforms.

IV. Outcome

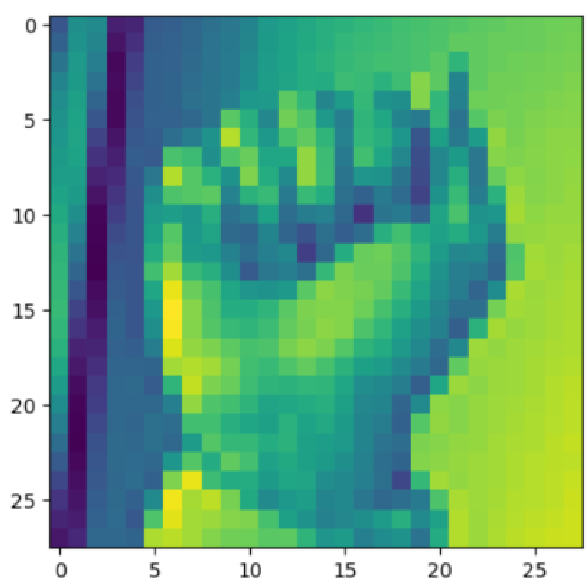

Fig. 2. **Sample Image from Training Set**

Among the most significant preprocessing steps before feeding the data into a CNN is reshaping the images to incorporate the single channel dimension (Fig 1). Multiple channels of data, usually three for RGB pictures (Red, Green, Blue), can be processed using CNNs. To prevent shape mismatch errors during model training and evaluation, the channel dimension must be explicitly included because the dataset consists of grayscale images.

Verifying the sample image visually aids in confirming that the reshaping was done successfully. After the preprocessing stages, it offers a visual assurance that the photographs are intact and structured correctly. Before moving on to model training, the most essential step in any machine learning workflow is to confirm the accuracy of the data.

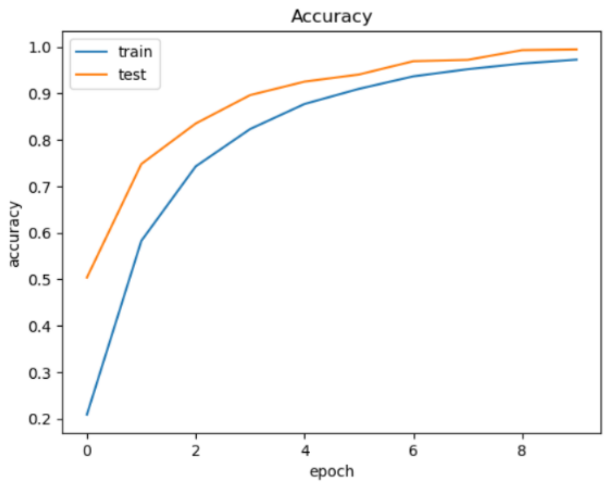

Fig. 3. **Train and Test data with Accuracy**

We can observe from Fig 3 that the prototype learns across several epochs, the training accuracy curve displays how well the model is doing regarding the instructions in dataset. The model is successfully learning patterns from the training information when there is a rising trend in training accuracy.

The model's ability to generalize to previously unseen data is shown by the validation accuracy curve. In a perfect world, the validation accuracy would rise and level off to show that the model is not overfitting and can adapt its learning to new sets of data.

In evaluating the model's learning process, tracking accuracy throughout both instruction and approval is essential: Overfitting is indicated if the validation accuracy either slows down or declines while the accuracy of the training keeps rising. Rather of learning to generalize, the model is simply learning to memorize the training set.

The accuracy charts aid in assessing the model's overall effectiveness. A model is deemed to be operating well if it exhibits high instruction and trained accuracy.

V. Conclusion

This work offers a thorough method for creating a real-time the system for Sign Language Recognition (SLR) system that will help deaf and hard of hearing individuals communicate more effectively. Through the utilization of modern machine learning and computer vision techniques, the framework is able to reliably translate movements from sign language into written. High accuracy and efficiency are guaranteed by the combination of MediaPipe for hand tracking and gesture detection with OpenCV for picture processing. Utilizing several types of machine learning models, such as CNNs, KNNs, and RFCs, improves the system's ability to adapt and capacity to adjust to changing environmental circumstances. The created SLR system has great promise for advancing equality and enhancing deaf community communication. It provides a workable solution for real-time interpretation of sign language, which is essential for smooth communication in daily life. The system's capacity to handle a range of sign language movements and its simple to use interface make it a valuable resource for promoting a more accessible and interconnected society.

VI. Future Work

Future research will concentrate on a number of important areas to enhance the functionality and potential of the SLR system. Expanding the dataset by adding more gestures and participant demographics is among the main goal. This will include gathering information from a wider range of signers to improve the system's capacity for generalization. Another crucial area is real-time deployment on mobile and edge devices, which requires the creation of lightweight models that are performance-optimized without compromising accuracy. A more comprehensive comprehension of sign language will result from the incorporation of multimodal data, such as body posture and facial expressions, greatly improving identification accuracy. Enhancements to the user interface are necessary to ensure that the system is functioning is usable and accessible for those who are deaf and hearing, with features like customized gesture sets and voice feedback. By putting continuous learning strategies into practice, the model will possess the ability to adjust and recover over time, keeping up with changing language trends. Finally, adding support for additional sign languages will broaden the system's use and require the training of models on a range of languages datasets. The SLR system will become more flexible, accurate, and accessible due to these developments, greatly improving deaf community communication and promoting an inclusive society.

References

[1] Orovwode, Hope Oduntan, Ibukun Abubakar, John. (2023). Development of a Sign Language Recognition System Using Machine Learning. 1-8.10.1109/icABCD59051.2023.10220456.

[2] I.A. Adeyanju, O.O. Bello, M.A. Adegboye, "Machine learning methods for sign language recognition: A critical review and analysis", Intelligent Systems with Applications, Volume 12, 2021, 200056, ISSN 2667-3053, https://doi.org/10.1016/j.iswa.2021.200056.

[3] Prof. Radha S. Shirbhate1, Mr. Vedant D. Shinde2, Ms. Sanam A. Metkari3, Ms. Pooja U. Borkar4, Ms. Mayuri A. Khandge5, "Sign language Recognition Using Machine Learning Algorithm", International Research Journal of Engineering and Technology (IRJET), Volume: 07 Issue: 03 — Mar 2020, e-ISSN: 2395-0056, p-ISSN: 2395-0072.

[4] S.Saravana Kumar1 , Vedant L. Iyangar2, "SIGN LANGUAGE RECOGNITION USING MACHINE LEARNING", International Journal of Pure and Applied Mathematics, Volume 119 No. 10 2018, 1687-1693, ISSN: 1311-8080 (printed version); ISSN: 1314-3395 (on-line version)

[5] Mahalakshmi V, Asst. Prof. Dr. E. Ranjith, "Sign Language Training Tool Using Machine Learning Techniques", International Journal of Research Publication and Reviews, Vol 4, no 6, pp 3488-3494 June 2023, ISSN 2582-7421

[6] Akshatha Rani K1, Dr. N Manjanaik2, "Sign Language to Text-Speech Translator Using Machine Learning", International Journal of Emerging Trends in Engineering Research, Volume 9. No.7, July 2021, ISSN 2347 – 3983

[7] V, Anjana T, Charulatha P, Dharishinie. (2023). Sign Language Recognition and Training Module. 10.21203/rs.3.rs-3057185/v1.

[8] Iftikhar Alam1, Abdul Hameed2 and Riaz Ahmad Ziar3 “Exploring Sign Language Detection on Smartphones: A Systematic Review of Machine and Deep Learning Approaches”, Hindawi Advances in Human-Computer Interaction Volume 2024, Article ID 1487500, 36 pages https://doi.org/10.1155/2024/1487500

[9] Sahoo, Ashok. (2014). Indian sign language recognition using neural networks and kNN classifiers. Journal of Engineering and Applied Sciences. 9. 1255-1259.

[10] Jayanthi P, Ponsy R K Sathia Bhama* B Madhubalasri , “Sign Language Recognition using Deep CNN with Normalised Keyframe Extraction and Prediction using LSTM”, Journal of Scientific Industrial Research Vol. 82, July 2023, pp. 745-755 DOI: 10.56042/jsir.v82i07.2375

[11] N. Subramanian, B., Olimov, B., Naik, S.M. et al. An integrated mediapipe-optimized GRU model for Indian sign language recognition. Sci Rep 12, 11964 (2022). https://doi.org/10.1038/s41598-022-15998-7

[12] Kothadiya, D.; Bhatt, C.; Sapariya, K.; Patel, K.; Gil-González, A.-B.; Corchado, J.M, “Deepsign: Sign Language Detection and Recognition Using Deep Learning.” Electronics 2022, 11, 1780. https://doi.org/10.3390/electronics11111780

[13] Pathan RK, Biswas M, Yasmin S, Khandaker MU, Salman M, Youssef AAF. Sign language recognition using the fusion of image and hand landmarks through multi-headed convolutional neural network. Sci Rep. 2023 Oct 9;13(1):16975. doi: 10.1038/s41598-023-43852-x. PMID: 37813932; PMCID: PMC10562485. https://rdcu.be/dI7jP

[14] Das, A., Gawde, S., Suratwala, K., Kalbande, D. (2018). Sign Language Recognition Using Deep Learning on Custom Processed Static Gesture Images. 2018 International Conference on Smart City and Emerging Technology (ICSCET). doi:10.1109/icscet.2018.8537248