A Survey of Machine Learning Techniques for Sign Language Translation

Sridevi G M
Department of Information Science and Engineering Dayananda Sagar Academy of Technology and Management Bengaluru, India sridevi.gereen87@gmail.com

Gourav Mishra
Department of Information Science and Engineering Dayananda Sagar Academy of Technology and Management Bengaluru, India gouravmishra42001@gmail.com

Aditi Patni
Department of Information Science and Engineering Dayananda Sagar Academy of Technology and Management Bengaluru, India aditiptn@gmail.com

Manish Kumar
Department of Information Science and Engineering Dayananda Sagar Academy of Technology and Management Bengaluru, India talk2manish.me@gmail.com

Ayush Kumar
Department of Information Science and Engineering Dayananda Sagar Academy of Technology and Management Bengaluru, India kumarayush0427@gmail.com

Abstract

Bridging the communication gap between deaf and non-verbal communities has long been a vital objective, with sign language recognition playing a crucial role. This research delves into the realm of automated sign language recognition, specifically focusing on American Sign Language (ASL) and leveraging the prominent ASL pickle data. Employing various machine learning algorithms, including Random Forest, Support Vector Machines, Convolutional Neural Network and K Nearest Neighbors, this study investigates key-point detection-based approaches to ASL recognition. The model's performance is analyzed in great detail through exhaustive testing, utilizing metrics such as F1 score, precision, and recall to determine the most effective approach. To improve user interaction, a user- friendly graphical user interface (GUI) is implemented, allowing for effortless interaction and prediction generation using the best machine learning model. Additionally, this paper provides a thorough review of the various techniques used for sign language translation.

Keyword-American *Sign Language(ASL), K Nearest Neighbors(KNN), Convolutional Neural Network(CNN), Random Forest(RF), Support Vector Machine(SVM)*

1. Introduction

Around the world, over 70 million people have hearing loss, and according to the World Federation of the Deaf, roughly 80% of them live in developing countries. For those with hearing impairments, sign language is the most preferred method of communication. Sign language recognition can be done using computer vision or other methods to varying degrees. Some people argue that sign language is made up of a set of movements, each with its own meaning. This meaning allows individuals to communicate with the outside world as well as other deaf and hard of hearing individuals. We propose a system that would improve sign language comprehension. To do this, we typically employ support vector machines (SVM), and for the classification challenge, we also utilized random forest algorithms (RF) and K-nearest neighbors (k

NN).Additionally, we are creating a training program for anyone interested in learning sign language. This will make it simpler for many families who cannot afford to send their children to school to learn the language and will also enable them to communicate with those who are hard of hearing without difficulty. This promotes the broader use and understanding of sign language.This paper's goal is to examine sign language recognition techniques and develop the best training tool for those with visual impairments.

1.1 ***Essential Features for Gesture Recognition***

Since hand motions are so distinctive in terms of shape variation, texture, and velocity, choosing the right features for gesture recognition is crucial. Separating features like finger orientations, fingers, skin color, and hand shapes for static hand identification makes it simple to determine hand posture. Those features are not consistently available or reliable due to the lighting and image background.

Other non-geometric characteristics, such as silhouette, color, and textures, are sufficient for recognition. The entire frame or converted image is used as the input because precisely characterizing features is challenging. The recognizer then implicitly and automatically finds the features.

1.2 ***Vision-Based Systems for Real-Time Sign Language Recognition***

One of the primary objectives of this technology is to create systems that can identify unique gestures and use them to transmit information or control equipment. However, hand postures are the static structure of the hand, whereas gestures are the dynamic movement of the hand, and gestures need to be represented in both the spatial and temporal domains.

The two primary techniques used to analyze hand gesture usage are:

Data glove: This method uses sensors placed on the hand to capture the movement of fingers and joints.

Vision-based technique: This method uses a camera to capture images of the hand and then analyzes those images to recognize the gestures.

Our main objective is to create a vision-based system capable of real-time sign language recognition. A vision- based system is preferable because it offers a clearer, simpler, and more realistic means of communication between a human and a machine. The system will accept the silhouette of the hand placed in front of it as the input. This allows it to recognize the shape of the hand, even if it is somewhat blurry.

Our top concern is that the silhouette of the hand shape might not always be sufficient to determine the character. We have therefore made every effort to decipher the person's intended gesture.

1.3 ***Basic Process of Sign Language Recognition***

Basic process of sign language recognition Opens in a new window

The basic process of sign language recognition (SLR) is quite straightforward, as seen in Figure 1. It adheres to the fundamental principles of machine learning algorithms. The SLR process is first trained with a dataset demonstrating how each hand gesture will be made to represent a character. This is then processed, and the features of the hand are used as a reference for the model to compare. When the input is obtained through video capturing, this is implemented. The input is taken by analyzing the hand's features. It then compares it to the dataset stored in the database and outputs the letter the person is displaying.

This process is broken down into the following steps:

Data Acquisition: This involves capturing images or videos of sign language gestures.

Preprocessing: This involves cleaning and enhancing the captured data.

Feature Extraction: This involves extracting key features from the data, such as hand shape, orientation, and location.

Classification: This involves using a machine learning algorithm to classify the extracted features into different sign language symbols.

Output: This involves displaying the recognized sign language symbol.

By following these steps, a vision-based system can be created that is capable of real-time sign language recognition. This system can be used to improve communication between people with hearing loss and those who can hear.

2. Literature Survey

A. Bar Bhuiya and colleagues introduced a methodology centered around deep learning-based convolutional neural networks (CNNs) to effectively model static signs within the realm of sign language recognition[1]. This research specifically applies CNNs to Hand Gesture Recognition (HGR), concurrently considering both alphabets and numerals of American Sign Language (ASL). The paper meticulously discusses the merits and demerits associated with employing CNNs in the context of HGR. The CNN architecture is constructed by modifying the AlexNet and VGG16 models for classification purposes. Feature extraction utilizes modified pre-trained AlexNet and VGG16 architectures, which are subsequently fed into a multiclass support vector machine (SVM) classifier. Evaluation metrics are based on diverse layer features to optimize recognition performance. Accuracy assessments for HGR schemes involve both leave-one-subject-out and a random 70–30 cross-validation approach. Additionally, the research delves into character-specific recognition accuracy and explores similarities among identical gestures. Notably, experiments are executed on a basic CPU system, eschewing high-end GPU systems to underscore the cost- effectiveness of the proposed methodology. Remarkably, the introduced system attains an impressive recognition accuracy of 99.82%, surpassing some contemporary state- of-the-art methods.

Sandrine Tor nay et al. proposed a technique delves into the realm of sign language recognition, focusing on the challenge of resource scarcity in the field[2]. The primary obstacle lies in the diversity of sign languages, each with its own vocabulary and grammar, creating a limited user base. The paper proposes a multilingual approach, drawing inspiration from recent advancements in hand shape modeling. By leveraging resources from various sign languages and integrating hand movement information through Hidden Markov Models (HMMs), the study aims to develop a comprehensive sign language recognition system. The research builds upon prior work that demonstrated the language independence of discrete hand movement subunits. The validation of this approach is conducted on the Swiss German Sign Language (DSGS) corpus SMILE, German Sign Language (DGS) corpus, and Turkish Sign Language corpus HospiSign, paving the way for a more inclusive and versatile sign language recognition technology.

Shirbhate et al. proposed a technique in his research paper addresses the vital role of sign language as a communication medium for the deaf and dumb community, emphasizing its significance for the 466 million people worldwide with hearing loss [3]. Focusing on Indian Sign Language (ISL), the study highlights the challenges faced in developing countries, such as limited educational resources and high unemployment rates among adults with hearing loss. The research aims to bridge the gap in sign language recognition technology, specifically for ISL, by utilizing computer vision and machine learning algorithms instead of high-end technologies like gloves or Kinect. The project's primary objective is to identify alphabets in Indian Sign Language through gesture recognition, contributing to the broader accessibility and understanding of sign languages in the context of Indian communication and education.

D.M.M.T. et al. in his paper studies the problem of vision-based Sign Language Translation (SLT), which bridges the communication gap between the deaf mute and normal people [4]. It is related to several video understanding topics that targets to interpret video into understandable text and language. Sign Language is a form of communication that uses visual gestures to convey meaning. It involves using hand-shapes, movements, facial expressions, and lip-patterns to communicate instead of relying on sound. There are many different sign languages around the world, each with its own set of gestures and vocabulary. For instance, ASL (American Sign Language), GSL (German Sign Language), and BSL (British Sign Language) are some examples.

ASL has around 6,000 gestures for common words, and fingers are often used to communicate obscure words or proper nouns. Sign Language is primarily used in the deaf community, including interpreters, friends, and families of the deaf, as well as people who are hard of hearing themselves. However, these languages are not widely known outside of these communities, which creates communication barriers between deaf and hearing individuals. Hand gestures are nonverbal communication methods for these people. It becomes very difficult and time consuming to exchange the information or feelings for a person who uses sign language to communicate with other non-users. One can make use of devices which translate sign language into words, but this is not an effective way as it can have huge costs and need maintenance. The main aim of our research is to form an effective way of communication between the people using sign language and the English language.

Mehreen Hurro et al. have proposed a technique for Sign Language Recognition using Convolutional Neural Network and Computer Vision [5]. We have also developed a similar system in which we use a 2D CNN model with a Tensorflow library. To extract important features from the input image, the convolution layers are applied that scan the images with a filter of size 3 by 3 and calculate the dot product between the frame pixel and the weights of the filter. After each convolution layer, pooling layers are then applied that

decrement the activation map of the previous layer and merge all the features that were learned in the previous layer's activation maps. This helps to reduce overfitting of the training data and generalizes the features represented by the network. Our convolutional neural network's input layer has 32 feature maps of size 3 by 3, and the activation function is a Rectified Linear Unit. The max pool layer has a size of 2x2, and the dropout is set to 50 percent. The layer is then flattened, and the last layer of the network is a fully connected output layer with ten units, and the activation function is Softmax. Finally, we compile the model by using category cross-entropy as the loss function and Adam as the optimizer.

3. Proposed System

The proposed system of our model contains the following modules:

- Data collection and pre-processing
- Algorithms Applied
- Model Training and Testing
- sign Prediction and Self Mode Training

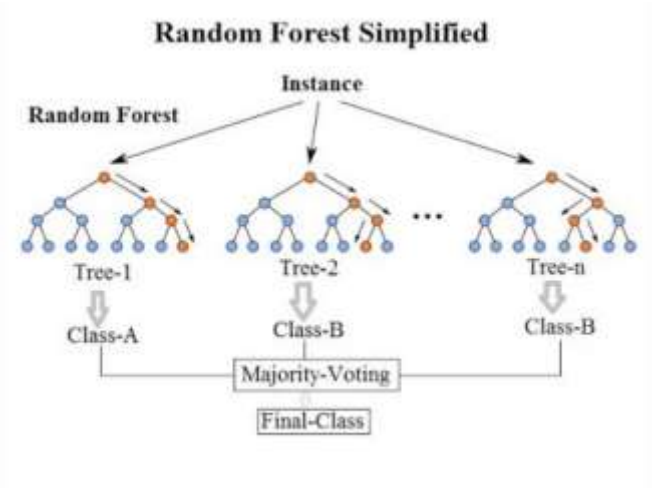
3.1 Data collection and pre-processing

This proposed system is passed on the dataset of ASL (American Sign Language). The main idea behind using ASL is that it is one of the most common Sign Language in the world. Due to its communication which translates to English language its known world-wide which leave us the perfect dataset for all people to understand and learn. Image processing is a method for converting a physical image to a digital one so that a person can edit, add to, or remove information from it. A video frame or photograph serves as the input for this type of signal distribution, and the output might either be another image or attributes associated to that image. In the system Feature Extraction becomes the main job for image pre- processing. For this first the image is captured using live cameras and taken as input which undergoes a major background check from BGR to RGB for OpenCV () to work. Then captured images were passed to a Hand tracking model from the Media Pipe framework which was deployed using the library. This is the main step where the feature of the model has been extracted. Using media pipe coordinates has been given for the images. These coordinates serve as the data values for further carrying out the task of Sign Language recognition.

3.2 Algorithms Applied

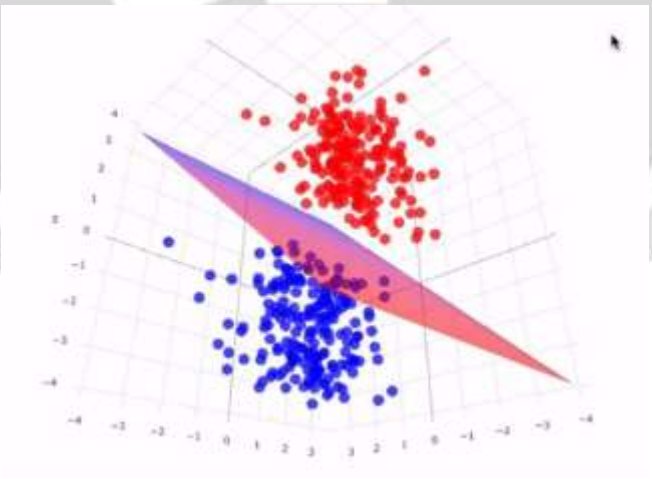
3.2.1 Random Forest

The algorithm constructs a collection of decision trees and combines their predictions to make a final prediction. Random Forest is a popular choice for sign language recognition due to its ability to handle complex, non-linear relationships between features and outputs. One advantage of Random Forest for sign language recognition is its ability to handle high-dimensional data, such as sign language data that includes multiple modalities. Performance of Random Forest, like any machine learning algorithm, will depend on the quality of the data and the appropriate selection of features and parameters. High recognition rates and accuracy are advantages of utilizing the Random Forest algorithm for sign language recognition. For instance, a study on Chinese Sign Language letter identification revealed that the random forest method outperformed artificial neural networks (ANN) with an average recognition rate of 95.48%. Another study suggested a pipeline that uses the random forest classifier to identify gestures in American Sign Language. The study built a sign language recognition model using three distinct algorithms, and the random forest classifier produced promising results. Overall, the Random Forest algorithm has demonstrated success in recognizing sign language, producing reliable and accurate results.



3.2.2 **Support Vector Machine**

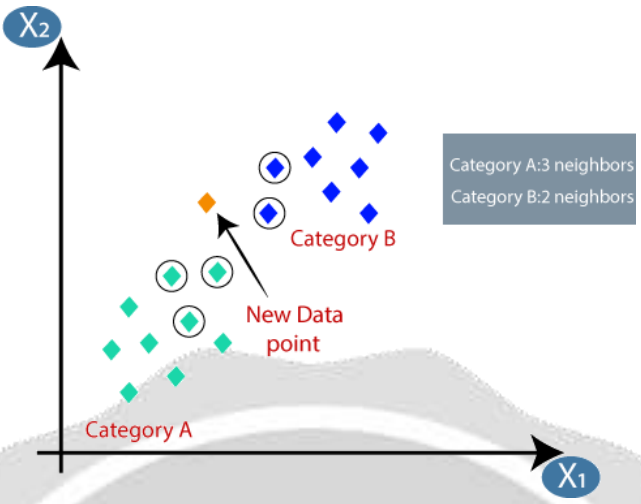
Support Vector Machine can be used to classify hand gestures into corresponding words or letters in sign language. The algorithm works by finding the optimal boundary (or hyperplane) between the classes in a high-dimensional feature space, and then classifying new data points based on which side of the boundary they fall on. The SVM method has many advantages for sign language identification, including excellent accuracy and the capacity to categorize input signs. One study, for instance, suggested classifying input signs into various categories using SVM classification to create a sign language recognition system. Another study recognized Indian Sign Language using the SVM algorithm with a good degree of accuracy using the photos. Similar experiments were conducted on the classification of sign language using Support Vector Machines and Hidden Conditional Random Fields. A hand gesture detection algorithm utilizing SVM and HOG was also presented in order to recognize gestures quickly and in real-time. Overall, the SVM algorithm has demonstrated effectiveness in the recognition of sign language, with excellent accuracy and the capability to categorize input signs into various classes.



3.2.3 **K Nearest Neighbor**

K-Nearest Neighbor (KNN) is a non-parametric, instance-based machine learning algorithm. In sign language recognition, KNN can be used to classify hand gestures into corresponding words or letters in sign language. The algorithm works by computing the similarity between a given test sample and the training samples, and then selecting the K training samples with the highest similarity. The KNN algorithm's capacity to categorize objects based on feature space and employ distance measurements as its classification criteria are two advantages of using it for sign language recognition. An Indian Sign Language identification system, for instance, was developed in a study employing the K-nearest neighbor (KNN) classifier, which categorizes objects based on feature space. KNN is one of the machine learning algorithms that uses distance measures as its classification criteria, according to another study. Similar to this, the KNN classifier was used to identify the signs in a sign language recognition system that was intended to help deaf-dumb persons. The K-nearest

neighbor (KNN) technique was utilized in a study to develop a classification method for Indian sign language recognition. Signs are grouped according to their features. Overall, the KNN algorithm has demonstrated success in classifying signs accurately based on feature space and distance measurements.



3.2.4 **Convolutional Neural Network**

Convolutional Neural Networks (CNNs) stand at the forefront of deep learning, particularly in the realm of computer vision, where they have revolutionized image and video analysis. These networks draw inspiration from the visual processing systems found in animals, emphasizing local connectivity and a hierarchical organization of visual information.

At the heart of CNN architecture are convolutional layers, designed to detect local patterns or features within input data. Convolutional filters, or kernels, traverse the input image, progressively capturing spatial hierarchies of features. This hierarchical learning allows CNNs to recognize simple elements like edges in initial layers and more intricate structures as depth increases.

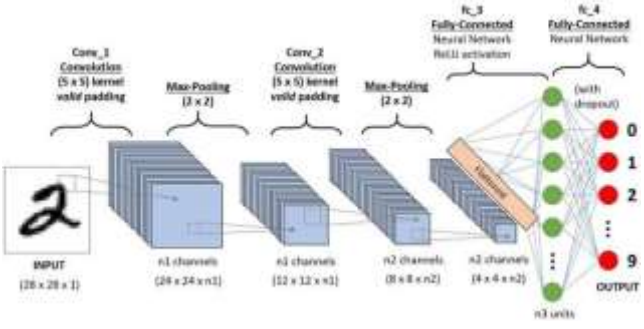
Pooling layers follow convolutional operations, reducing the spatial dimensions of feature maps while retaining critical information. Pooling, through operations like max pooling or average pooling, contributes to achieving translation invariance and lessening computational requirements.

Fully connected layers come into play after convolutional and pooling layers, amalgamating the learned features to make predictions. These layers establish connections between every neuron, facilitating the capture of global dependencies.

One notable feature of CNNs is weight sharing, enhancing parameter efficiency. By applying the same convolutional filters across different input regions, the network learns reusable features.

Transfer learning is a common practice with CNNs, leveraging pre-trained models on extensive datasets (e.g., ImageNet) and fine-tuning them for specific tasks. This approach proves particularly beneficial when dealing with limited labeled data.

CNNs find application across a spectrum of tasks, excelling in image classification, object detection, facial recognition, medical image analysis, and the development of autonomous vehicles. Despite their success, challenges such as sensitivity to input variations, computational demands, and the necessity for substantial labeled data persist.



3.3 ***Model Training and Testing***

Using the dataset, the model is trained. There will be two datasets namely training and testing. Usually training and testing datasets are separated. A training dataset is a starting set of data that is used to instruct machine learning models on how to recognize specific patterns or carry out a specific activity. The dataset that is utilized to train a machine learning model is rather vast. The machine learning model is trained or fitted using the training dataset, which is a crucial stage in creating machine learning models. Prediction models are taught to see patterns and generate predictions based on the data using the training dataset. Making sure that the training dataset is indicative of the issue being addressed is essential for the success of machine learning models. Once the model is trained it goes to the testing phase. This is done using a testing dataset. the testing dataset used to gauge its effectiveness. A testing dataset is a subset of data used to evaluate the performance and advancement of machine learningalgorithms and to adjust or improve their training for improved results. Unseen data is used in the testing dataset to provide a fair assessment of how well a model fits the training dataset while changing model hyperparameters. Every machine learning model requires the testing dataset, which helps to confirm the model's accuracy and dependability. The model's performance is assessed using the testing dataset, and it may be modified or optimized for better outcomes. By doing this training process we are creating numerous models which are used for the evaluation purposes. For this we are selecting the best model by comparing the model using their evaluation metrics and the time they have taken for computing the output. After selecting the best model, we can proceed with the step of testing which occurs when the input is provided by the means of a web camera. This input is compared with the provided dataset inside a model and the output is given to us in the form of alphabets For this we are selecting the best model by comparing the model using their evaluation metrics and the time they have taken for computing the output. After selecting the best model, we can proceed with the step of testing which occurs when the input is provided by the means of a web camera. This input is compared with the provided dataset inside a model and the output is given to us in the form of alphabets.

3.4 ***Sign prediction and self mode training***

The last step in the model that has been proposed is displaying the output. There are two steps in this. One is actual sign prediction and the other is self-mode training. The sign prediction is one of the models that has been created. The model to be displayed in the manner of English alphabets so that it will be an accurate description of the process. This will show us the perfect alphabet that we have been trying to predict. It also shows us the confusion matrix which will indeed result in the accuracy of the alphabet.

The model proposed has a 98 percent accuracy as the least and 100 percent accuracy on most of the alphabets. Once the model has been predicted training model comes in.The above model is connected with a web camera so that the system can take the live footage as the input for the model and the tool. By connecting the web camera, it will have the live camera footage for the model. The footage should be at least somewhat clear so that the device will provide accurate results. Taking input in the form of a live camera instead of a recorded video also provides a great benefit in the training tool as the person will know what mistake he has made and how to rectify it then and there instead of watching the video he made again and again. This makes a great difference in efficiency and it will help the person learn quickly and efficiently.

A raw footage of some letter is taken as the input of the model. This is achieved by reading the live camera feed which is acquired using a web camera. The next step in taking input is removing all the unnecessary background which is just an obstacle in the path of decoding. By removing the background, the model can achieve a more accurate shape of what -the shape of the hand is. Binarization is the method of converting any entity's data characteristics into vectors of binary values to improve the performance of classifier algorithms. By doing this step, it makes it efficient for the document to achieve a more accurate shape of the hand so that the output will be precise. The individual pixels in an image can be identified using a computer vision method called image segmentation.by doing this step we are making the image as a prefect input for the system as it can understand the language of decoding. By removing the background, the model can achieve a more accurate shape of what the shape of the hand is. Focuses more on details that are unwanted as the image is pixelated. This includes mainly the wrist but, in some cases, arms are also included if the whole arm is read by the system. The last main step in generating a perfect input for the model is centralization of the image. Once the small details are removed, there will be a more precise silhouette of the hand gesture. This is centralized for the model so that it can have a good input.

The next step in the model proposed is the comparison of the input to the trained model. Now this step occurs in the training model as shown in the above diagram. Once the input is entered this and follows the other step it will enter the chosen training model which has been selected with the help of the test and their speed. Once entered these inputs are compared with possible images from the dataset that the model finds quite alike to the

input. After this the images are made sure that they are compared with every possible scenario that the model believes may match the silhouette of the image. After this the model will take the decision about the letter that the found about being quite an accurate description of the shape the hand is making to display the output and the output will be an alphabet the model is predicted. The training model will have the following steps as above and will get in once the sign is predicted.

Using the model created above we are planning to build a tool which will help people to learn ASL. This tool will make sure that the people has kept the signs of alphabets quite accurately. By creating this tool, we will make sure that everyone can learn sign language with ease and it will also help the deaf people who can't afford a school. This training tool will be integrated with the web camera available on the system. This is made sure that it captures the correct shape of hand which the person is trying to make. Once it reads the video feed by the camera, it will segment the video into frames. It will go through the above process in the selected model where the model will compare the available dataset with the video frame. It will make sure that the video frame will be compared with every possible image that is available in the dataset. There is also a time limit where the people can keep track of time when they learned their symbol. This is very crucial as both the computer and the person can have the idea of how well they are doing. If the person does it in less time it will take it as a failure and it will push the person more. For e.g. If a person is training to make an "A", this training tool will make sure the person is making the sign quite accurate to the original and if the person comes for the second time it will take the previous time as example and will make sure the person does it in less time for it considering the learning as success.

After this only the person can move forward to the next letter. By doing this we can make sure that everyone learns sign language accurately.

4. Outcomes

Millions of people worldwide are hearing-impaired, often facing limited access to education and communication resources. This proposed system addresses this critical need by providing an accessible and

effective solution for both sign language learning and communication.

The system goes beyond simply translating hand gestures to letters. It empowers both learners and communicators through:

Accessibility: Anyone can learn ASL conveniently, overcoming barriers like location and financial constraints.

Accuracy: Advanced machine learning algorithms ensure precise sign recognition and insightful feedback.

Personalized Learning: The system adapts to individual learning styles and provides tailored feedback for optimal progress.

Bridging Communication Gaps: By facilitating communication between hearing and non-hearing individuals, the system fosters inclusivity and understanding.

The system offers comprehensive functionality for a holistic learning experience:

1. Real-time Recognition:

Captures live video feed through webcam integration.

Identifies and displays the corresponding English alphabet character for each sign performed.

Provides a confusion matrix for evaluating performance and identifying frequently misclassified signs.

2. Interactive Training:

Offers real-time feedback on the user's hand gestures.

Tracks the time taken to perform each sign, encouraging efficiency and mastery.

Compares the user's hand gesture with the ideal representation of the intended sign.

Provides personalized guidance and corrective feedback for improved accuracy.

Adapts the training difficulty based on individual progress, ensuring optimal learning pace.

While initially focused on ASL, the system has the potential to evolve into a comprehensive platform serving diverse needs:

Support for Additional Sign Languages: Expanding accessibility to a wider range of users and communities.

Advanced Gesture Recognition: Incorporating recognition of other gestures for broader applications.

Enhanced Feedback Mechanisms: Providing more detailed and interactive feedback for a richer learning experience.

Real-time Language Translation: Translating sign language to spoken language and vice versa for seamless communication.

This proposed system has the potential to revolutionize sign language learning and communication. By combining advanced technology with an intuitive interface, it empowers individuals, bridges communication gaps, and fosters a more inclusive and connected world.

Difference between various machine learning models

MODELS	***ADVANTAGES***	***ACCURACY***
KNN	Simple,effective,requires no training time	96.20%
CNN	Captures semantics of text better the KNN, robustness of distortions	98.10%
Random Forest	It produces a highly accurate classifier and learning is fast.	94.80%
SVM	In high dimensional spaces, it tends to be more effective.	98%

5. Conclusion

The development of sign language recognition technology has profound implications for the non-verbal and hearing- impaired community, paving the way for enhanced communication accessibility and social inclusion. This project successfully implemented a keypoint detection- based sign language recognition system for American Sign Language (ASL) by leveraging the power of machine learning algorithms. The project further developed a user- friendly training tool to capture user input and perform sign letter prediction, achieving an accuracy of 97% with the Random Forest algorithm, 96% with K-Nearest Neighbors, and 95% with Support Vector Machines.

The success of this project underscores the immense potential for future research to continue refining the accuracy and efficiency of sign language recognition. This advancement holds the promise of significantly impacting the lives of individuals who rely on sign language for communication by fostering greater access to information, opportunities, and connection within the larger community.

Abbreviations

Random Forest algorithm (RF), Support Vector machine (SVM), K-Nearest Neighbor (KNN), Convolutional Neural Network (CNN), American sign language (ASL), Sign language recognition (SLR)

References

[1] A.A. Bar Bhuiya, R.K. Karsh, R. Jain "CNN based feature extraction and classification for sign language" Multimedia Tools and Applications, 80 (2) (2021), pp. 3051-3069.

[2] Sandrine Tor nay, Marzieh Razavi, Mathew Magimai-Doss "Towards multilingual sign language recognition" Proceedings of the ICASSP, IEEE international conference on acoustics, speech and signal processing (2020), pp. 6309-6313

[3] Shirbhate, Radha S., Mr. Vedant D. Shinde, Ms. Sanam A. Metkari, Ms. Pooja U. Borkar and Ms. Mayuri A. Khandge. "Sign language Recognition Using Machine Learning Algorithm." Volume: 07 Issue: 03 Mar 2020.

[4] D. M. M. T. Ankur Verma, Gaurav Dubey, "Sign Language Translator", IJAST, vol. 29, no. 5s, pp.246 - 253, Mar. 2020.

[5] MehreenHurroo, Mohammad Elham, 2020," Sign Language Recognition System using Convolutional Neural Network and Computer Vision". INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT)
Volume 09, Issue 12,2020.

[6] Shruty M. Tomar, Dr. Narendra M. Patel, Dr. Darshak G. Thakore. A Survey on Sign Language Recognition Systems. IJCRT, Volume 9, Issue 3 March 2021.I. S. Jacobs and C. P.