Gesture based Sign Language Recognition system using Mediapipe

Laveen Chandwani (✉ chandwanila18.elec@coep.ac.in)
COEP Technological University

Jaydeep Khilari
COEP Technological University

Kunal Gurjar
COEP Technological University

Pravin Maragale
COEP Technological University

Ashwin Sonare
COEP Technological University

Suhas Kakade
COEP Technological University

Abhishek Bhatt
Symbiosis Skill and Professional University

Rohan Kulkarni
COEP Technological University

Research Article

Keywords: Machine Learning, Mediapipe, SVM, Sign Language, Gesture recognition, Assistive technology

Posted Date: June 28th, 2023

DOI: https://doi.org/10.21203/rs.3.rs-3106646/v1

Additional Declarations: No competing interests reported.

Gesture based Sign Language Recognition system using Mediapipe

[1]Laveen Chandwani, [1]Jaydeep Khilari, [1]Kunal Gurjar, [1]Pravin Maragale, [1]Ashwin Sonare,
[1]Suhas Kakade, [2]Abhishek Bhatt, [1]Rohan Kulkarni,
[1]College of Engineering, Pune, India; [2]Symbiosis Skill and Professional University ,Pune, India;
chandwanila18.elec@coep.ac.in; khilarijd18.elec@coep.ac.in; kunalsg18.elec@coep.ac.in; pravinrm18.elec@coep.ac.in; sonareap18.elec@coep.ac.in; smk.elec@coep.ac.in; abhishek.bhat@sspu.ac.in; rsk17.elec@coep.ac.in.

ABSTRACT

Sign language is a visual language that uses hand motions, changes in hand shape, and track information to convey meaning. It is the primary mode of communication for those with hearing and language impairments. The use of sign language for communication is limited, despite the fact that sign language recognition can help a large number of such persons deal with regular people. As a result, there is a need to create a more comfortable approach for people with hearing and language impairments to learn and work in order to improve their lives. Therefore, the basic idea behind this article is to make the communication between normal human beings and deaf people much easier. In order to recognize static gestures associated with sign language alphabet and a few commonly used words, we conducted a comprehensive research study employing the hand tracking technique Mediapipe and a gesture classification model based on Support Vector Machine (SVM). The results of the experiments are validated using Recall, F1 Score and Precision. Based on the validated results, we recommend the application of the discussed techniques for such communication. The suggested methods have high generalization qualities and deliver a classification accuracy of around 99 percent on 26 alphabet letters, numerical digits, and some regularly used words.

Keywords: Machine Learning, Mediapipe, SVM, Sign Language, Gesture recognition, Assistive technology.

1. Introduction

Humans communicate with one another using natural language channels such as words and writing, or by body language (gestures) such as hand motions, head gestures, facial expression, lip motion, and so forth. Comprehending sign language is equally as vital as understanding natural language [13].

People with hearing impairment use sign language as their preferred mode of communication. Without a translation, people with hearing impairments have difficulty speaking with other hearing people. As a result, implementing a system that understands sign language would have a substantial positive impact on the social lives of deaf people. According to the World Health Organization, 466 million individuals worldwide (more than 5 percent of the population) have impaired hearing, with 34 million of them being teens (WHO). According to studies, by 2050, these numbers will have surpassed 900 million. Furthermore, the majority of cases of profound hearing loss, which afflict millions of individuals, occur in low and middle-income nations [2].

Furthermore, the majority of cases of substantial hearing loss, which affects millions of individuals, occur in low- and middle-income nations. There are more than 135 distinct sign languages spoken worldwide, including American Sign Language (ASL), British Sign Language (BSL), and Indian Sign Language (ISL) [15].

Machine learning enables the development of systems that accurately interpret sign language, which can greatly improve communication and social lives of deaf people. These technologies are particularly important for those living in low and middle-income nations where the majority of hearing impairments occur. The growing prevalence of hearing loss worldwide highlights the urgent need for technological solutions to help bridge the communication gap between hearing-impaired individuals and the rest of society.

Machine learning is a branch of artificial intelligence that deals with the methods that let computers extract meaning from data and create AI applications. In the meanwhile, deep learning is a subset of machine learning that enables computers to resolve increasingly challenging issues [11]. As deep learning develops transferable answers, it is more powerful than traditional machine learning. Through neural networks, or layers of neurons/units, deep learning algorithms are able to produce transferable solutions [12]. Deep learning is a subset of machine learning where a computer program learns to carry out classification operations on complex input such as images, text, or sound. These algorithms are able to execute at a state-of-the-art (SOTA) level of accuracy and, in certain situations, even surpass humans. Numerous labeled data points and intricate neural network topologies are used to learn them. It is a vital part of modern innovations like self-driving cars, virtual assistants, and face recognition.

In our research, we have thoroughly examined the existing literature on Sign language recognition. We will now focus on the most notable research papers and discuss their methods for feature extraction, image pre-processing, and image classification, which employ a variety of algorithms including SVM, KNN, and CNN. Additionally, we have examined several image-processing techniques, including Canny-edge detection, Convex-hull algorithm, and Gaussian blur filter, among others.

A Microsoft Kinect camera was used to create a sign language recognition system in [6]. This was chosen to allow the whole programme to be independent of restrictions such as poor illumination, loud input, and so on. Depth and Motion were the two main feature capturing modules used in their methodology. In fact, a feature vector was calculated for each frame of the video series, and some preprocessing was applied to each frame to eliminate undesired noise and provide a clean image of the depth map. They used the Gaussian blur filter 15 and the Erosion filter to do this and also presented the depth information using a 256-bin histogram for a depth image. They were able to create the feature matrix for that particular video sequence or gesture using the combined array of feature vectors from all of the frames in the video sequence. Following this pre-processing, the feature matrix was given as an input to a multi-class SVM classifier to construct an appropriate Machine

Learning model for classification of the test files using kernel functions, with the linear and RBF kernels being specifically employed. The total accuracy achieved was between 81.48 and 87.67 percent. However, this work was unable to investigate other high-level characteristics such as optical flow information, motion gradient information, and so on, which may have improved accuracy performance.

A more precise real-time Hand Gesture Recognition (HGR) system based on American Sign Language is the primary goal of [8], which is to illustrate (ASL). The combination of K-curvature and convex hull approaches is proposed as a novel feature extraction technique. This method, known as the "K Convex Hull" technique, can recognize fingers with extreme precision. An ANN is used in this system together with feed forward and reverse propagation techniques to train a network with 30 feature vectors to accurately identify 37 indications of American alphanumeric letters, which is beneficial for HCI applications. The entire gesture recognition rate of this system in a real-time scenario is 94.32 percent.

The study described in [17] involved the use of a camera to capture images of Indian Sign Language (ISL) hand gestures. Before feature extraction from image, its pre- processing was done. In this work, a unique approach of the Canny Edge 16 Detection Algorithm was discussed. It was found that Canny edge detection algorithm is able to detect both strong and weak edges proving it to be more accurate than other techniques like Laplacian or Gaussian. Once the image's necessary elements have been extracted, it is matched with the data set, which is categorized using a CNN, and the appropriate text is generated. This text is then converted into a voice. Similarly, a regular person's vocal input is recorded and turned into text using a microphone. After that, the text is matched against the data set, and a corresponding sign is created. This method bridges the communication gap between hearing-impaired and non-hearing-impaired people. The overall accuracy obtained in this method is 98% tested for the set of 35 alphanumeric gestures of ISL.

[19] reviews and compares several algorithms and techniques for creating single hand gesture detection systems utilizing various vision-based methodologies. The research uses the hand's fundamental structure as well as properties like centroid to identify the pattern the fingers and thumb generate and assign code bits, i.e., changing each gesture into a set of 5 digits representation. Motion is recognized using centroid movement in each frame. The study uses techniques like K-means clustering or thresholding for background removal, Convex Hull or a custom peak identification algorithm, and text to voice API to translate gesture-related words and phrases into speech. The Convex Hull algorithm is used to identify the smallest convex polygon that contains every point from the frame.

2. Dataset

In this work, we have utilized the ASL dataset [20] consisting of 51 classes, with approximately 4000 images per class. The classes comprise the alphabet, numbers, and commonly used words such as ‘Hello’, ‘Help’, and ‘Stop’. The alphabet class enables the formation of new words through fingerspelling, where individual letters are used to represent words without a designated sign symbol.

A Python script was employed to efficiently convert the image class folders into a .csv file, which stores the (x, y, z) coordinates of all landmark points of each sign with their respective outputs. An 80:20 train-test split was implemented to improve the model's feature extraction process.

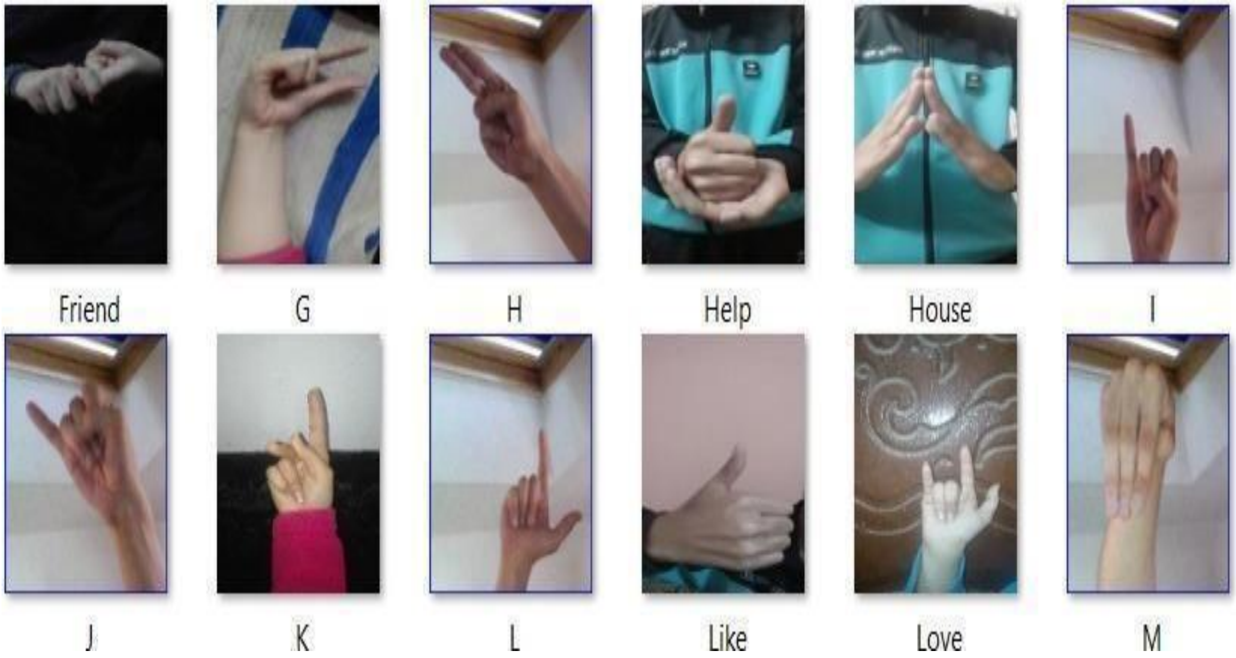


Fig. 0 Various Sign Symbols.

3. Mediapipe

Sign language recognition has the potential to improve the situation of a large number of disabled people while dealing with normal human beings but the use of sign language for communication is limited. As a result, there is a need to create a more convenient approach for persons with hearing impairments to learn and work in order to improve their lives.

Gesture recognition has been studied extensively utilizing traditional techniques such as body component tracking, different color glove-based tracking, Kinect depth sensor tracking, and skeleton tracking. Multiple methods have been used to solve this problem like modified CNN, image segmentation, SVM and deep learning.

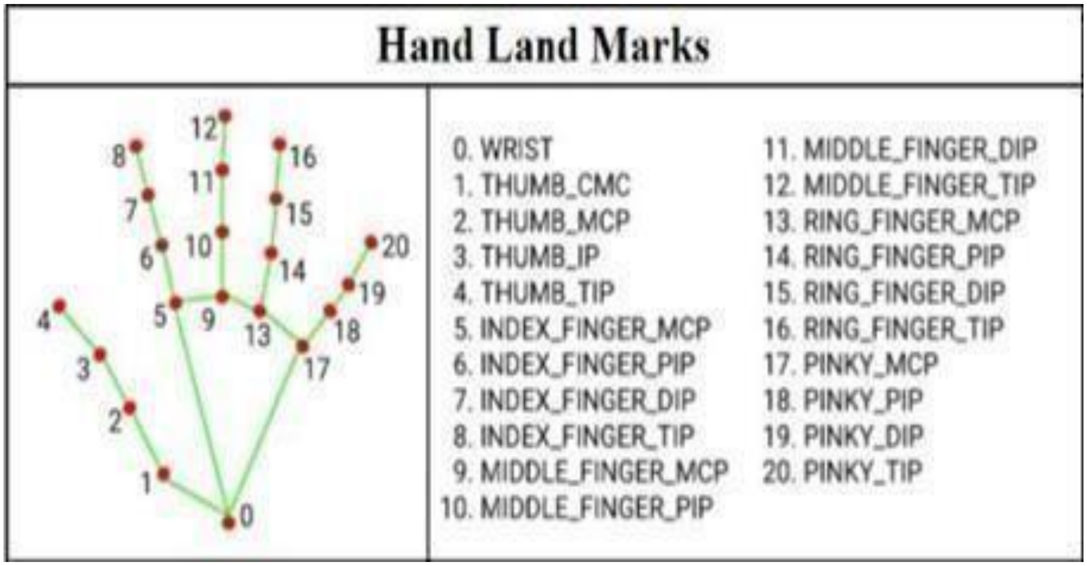
Many machine learning algorithms have been developed for hand gesture recognition so as to create AI-based applications. Out of them, MediaPipe can be used for hand gesture recognition. Google supported MediaPipe framework can be used for solving several problems like face-recognition, face-map, eye, hand, pose-estimator, holistic, hair, object-detection, box tracking and KIFT. With the help of the MediaPipe framework, we can develop an algorithm or model for the application, then help the application by providing results that can be cloned across different platforms.

The MediaPipe framework is composed of three major components: (1) performance evaluation, (2) a mechanism for collecting data from the sensor (3) an assembly of reusable parts. A graph consisting of all the parts called the calculators is known as pipeline, wherein every calculator is inter-connected by channels through which the data flows. Developers can create their required application by removing or delineating user defined calculators anywhere in the graph. This result of calculators and channels creates a data-flow diagram.

Hand gesture recognition with the MediaPipe framework is a dependable and high-fidelity hand and finger-tracking system. Mediapipe hands uses an integrated ML pipe of several models working together [18]: (1) A palm recognizer processes the captured hand image, (2) A hand landmark model takes processed image as input and returns hand with 3D key points as output. (3) A gesture recognition model which processes the 3D hand key-points and classifies them into a discrete set of gestures.

The palm detection model outputs a precisely cropped picture of the palm that is then sent to the landmark model. This method does away with data augmentation, which is used in deep learning models [5] to rotate, flip, and scale images. The technique of detecting hands is time-consuming and difficult since it involves working with different hand sizes, thresholding, and image processing. Prior to identifying hands with connected fingers, a palm detector is trained, which estimates bounding boxes around hard objects like fists and the palm. The second method is to utilize an encoder-decoder as an extractor for a larger scene context [14].

Hand Landmark model implements machine learning model to take 21 3-D key points of a hand from just a frame using regression which will directly produce the coordinate prediction. Even with faintly visible hands and self-occlusions, the model acquires a rigorously defined hand position representation [6]. It provides better real-time performance on devices compared to other algorithms and can be scaled for multiple-hands in a single frame.



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Fig. 1 Hand Landmark [Source: Ref [21]]

Hand-knuckles of the landmark have x, y, and z coordinates where x and y are normalized to [0, 1] as width and height of the image, while z represents the depth of the landmark. The closer the landmark to the camera, the value of z becomes smaller.

4. Experimentation

In order to achieve our desired objective, we have created an end-to-end web application that allows real-time communication between common people and deaf people without any use of hardware technologies like sensors, microcontrollers, etc. This website makes user interaction comfortable as it consists of combined application of Sign to Text and Text to Sign conversion, along with other essential features. To create this application, we have made use of multiple technologies and frameworks. HTML, CSS and JavaScript tools are used for Frontend and Flask (a Python web framework) is used for Backend. In Backend, the machine learning model is loaded in the form of a pickle (.pkl) file. This .pkl file allows easy serialization and deserialization of any ML model.

The functionality of our website is that it takes the webcam video as the input which captures our hand image. Later, Mediapipe technique is applied to this extracted image and key points are marked accordingly which then stores the (x, y, z) coordinates of the landmarks. Last but not least, this data is sent into the Support Vector Machine classifier, a supervised machine learning classifier (SVM). Regression and classification studies both use the SVM model. Finding the most important dividing line is done using it. The primary objective of this approach is to identify the best hyperplane for dividing and separating training vectors. Using gamma as the RBF parameter, SVC is an SVM classifier (Radial basis function kernel). To determine if a model is overfitting, underfitting, or providing the optimum fit, one uses the gamma value. The pickle (.pkl) module was utilized to load the two files, X and y, which are data files used for training the SVM model. The X file contains a list of image pixels, while the Y file contains labels for the list of pixels. After loading the dataset, it is passed to the model for training purposes.

The SVM model is represented by the equation:

f(x) = sign(sum(alpha_i * y_i * exp(-gamma * ||x_i - x||^2)) + b) ....(1)

where alpha_i are the Lagrange multipliers, y_i are the corresponding labels, exp(-gamma * ||x_i - x||^2) is the RBF kernel function, x represents the input feature vector, and b is the bias term. The hyperparameters C and gamma are typically determined through a grid search or cross-validation process. Once the model is trained, the webcam images are passed to the model for testing. The model recognizes the corresponding letter/word and outputs it on the screen.

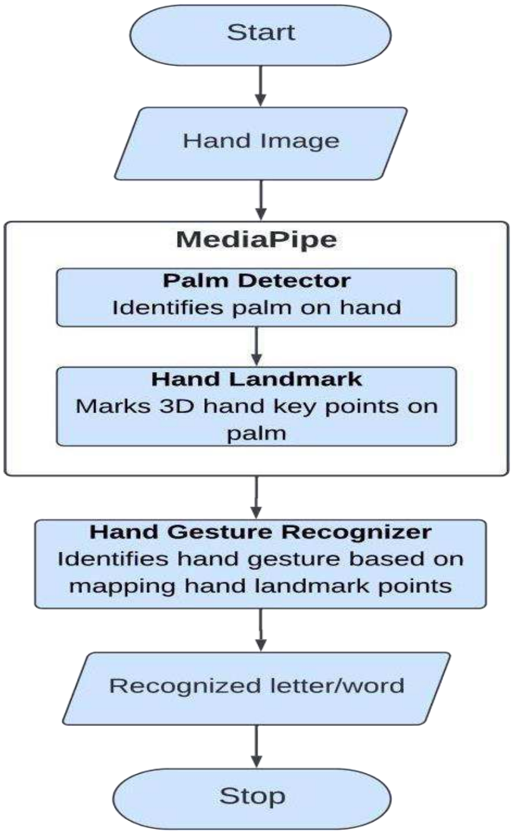
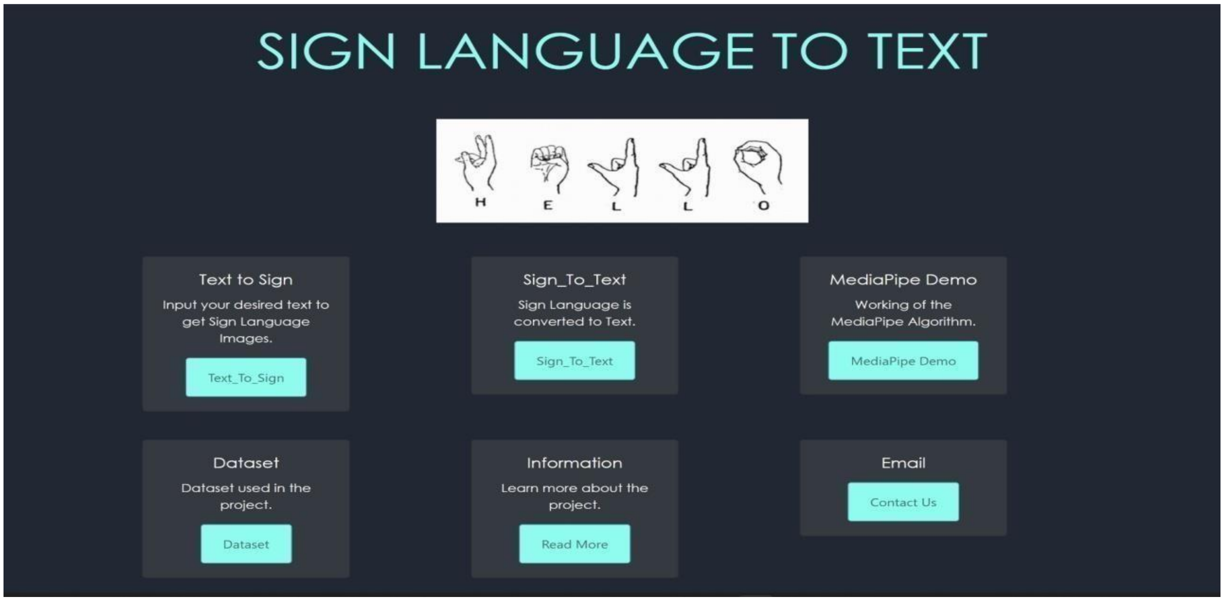


Fig. 2 Methodology

Some images of our integrated web application are shown below:



SIGN LANGUAGE TO TEXT

Text to Sign

Input your desired text to get Sign Language Images.

Sign_To_Text

Sign Language is converted to Text.

MediaPipe Demo

Working of the MediaPipe Algorithm.

Dataset

Dataset used in the project.

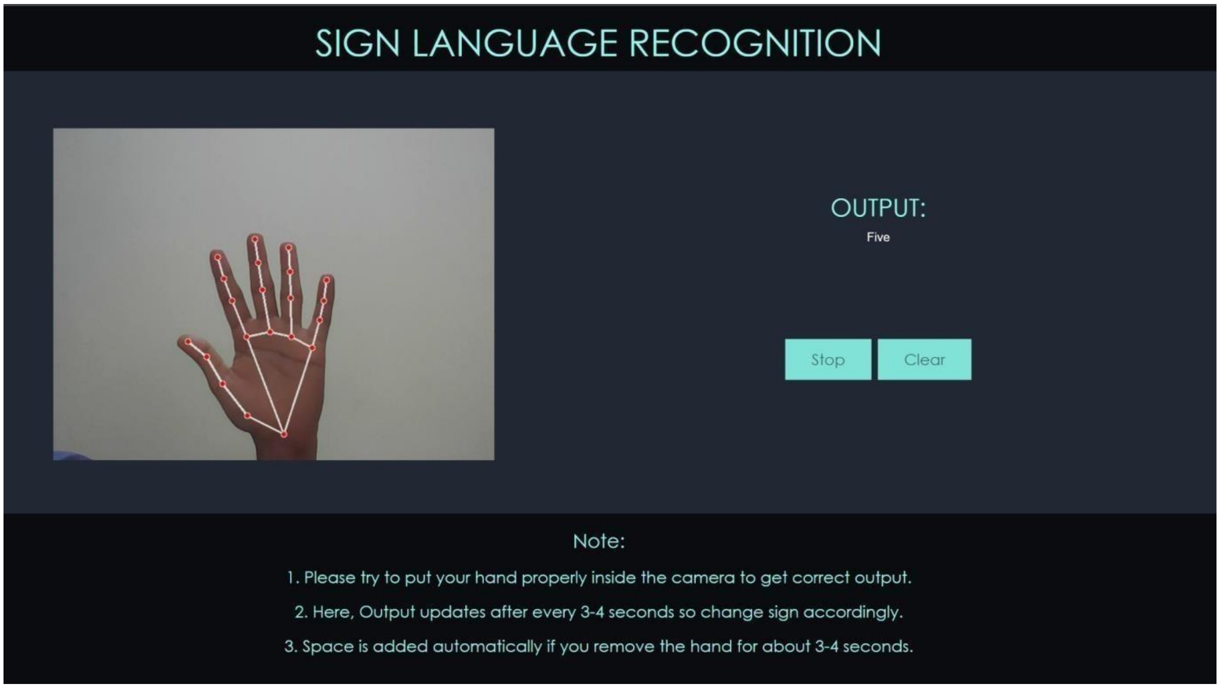
Information

Learn more about the project.

Email

Fig. 3 Home Page

The above image is the first page of our website. It contains the path to all the important sections present in the website like Sign to Text, About, Text to Sign, Dataset, etc.



SIGN LANGUAGE RECOGNITION

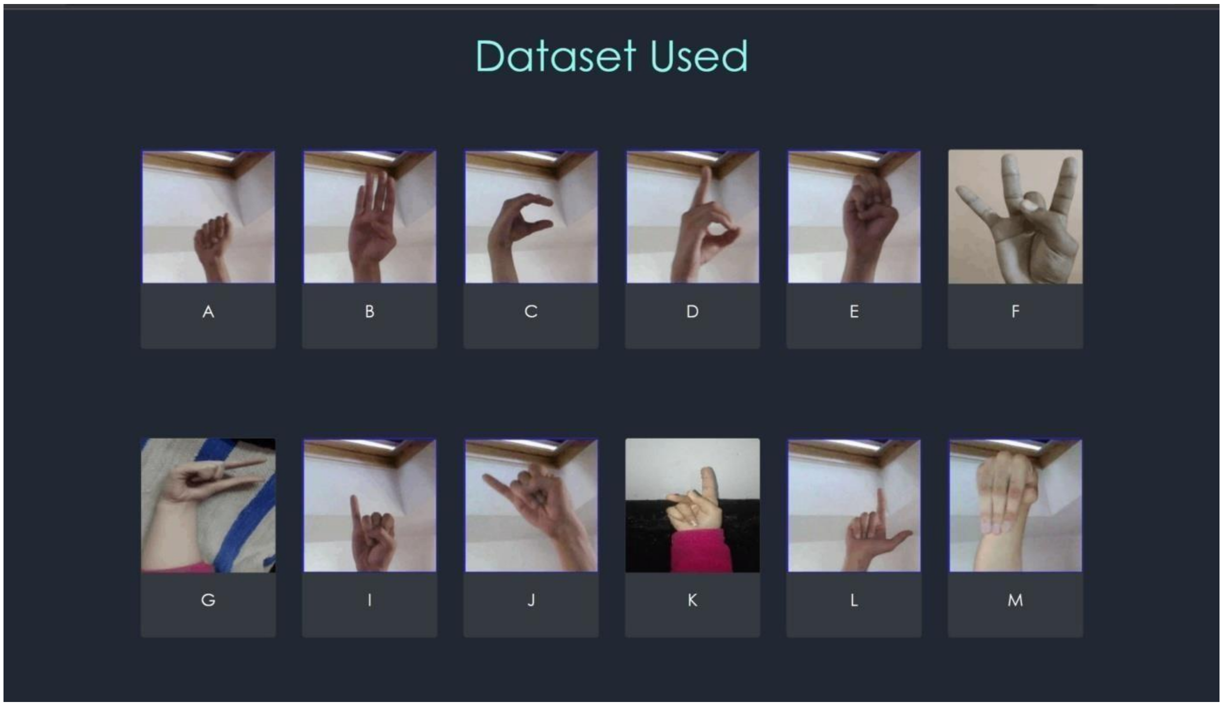
OUTPUT:
Five

Note:

1. Please try to put your hand properly inside the camera to get correct output.
2. Here, Output updates after every 3-4 seconds so change sign accordingly.
3. Space is added automatically if you remove the hand for about 3-4 seconds.

Fig. 4 Sign to Text Page

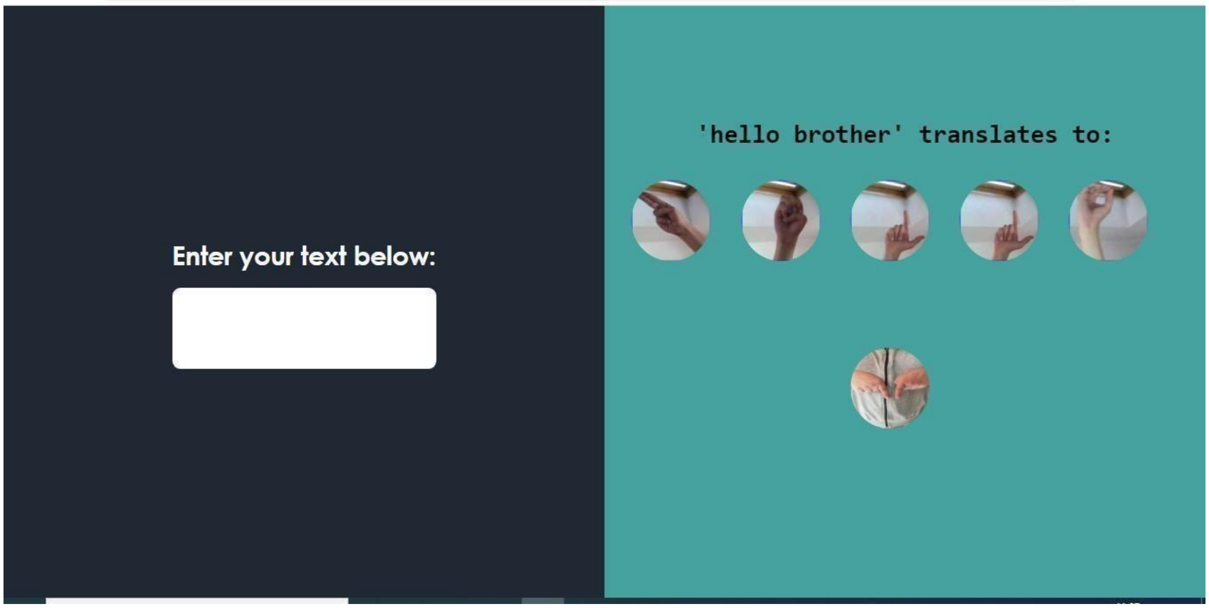
The above picture is the most crucial page in the web application since it shows the thorough implementation of our project.



Dataset Used

Fig. 5 Dataset Page [Source: Ref [20]]

The above image contains the dataset used for our project and its detailed information along with pictures of each sign. Anyone who wants to learn and understand the signs can refer to this page.



Enter your text below:

'hello brother' translates to:

Fig. 6 Text to Sign Page

The above picture depicts the reverse process. It converts the entered text to its respective signs.

5. Results

Various machine learning models are used for sign detection. These models are evaluated based on parameters like accuracy, recall, F1 score, etc. Among the utilized models, it is observed that SVM outperformed other machine learning techniques such as Naive Bayes, KNN, Decision Tree, etc. by achieving an accuracy of 98.65% (training) and 98.35% (testing) as shown in table 0. The reason it outperformed is because of its effectiveness in high- dimensional spaces where it draws a hyperplane boundary in order to classify the labels. It is also computationally less extensive and works well for image analysis tasks.

The below table shows the values of training and testing accuracy along with Recall, F1Score and Precision for different tried models:

Table 0 Results for various ML algorithms

Model	Train (%)	Test (%)	Recall	F1 Score	Precision
SVM	**99.70**	**98.975**	0.98	0.98	0.98
Random Forest	99.89	97.50	0.97	0.97	0.97
Decision Tree	99.89	91.52	0.91	0.91	0.91
Naïve Bayes	50.63	50.84	0.50	0.50	0.50
KNN	97.62	96.39	0.96	0.96	0.96

The below confusion matrix for SVM algorithm prints the correct and incorrect values in number count which gives us a good data visualization.

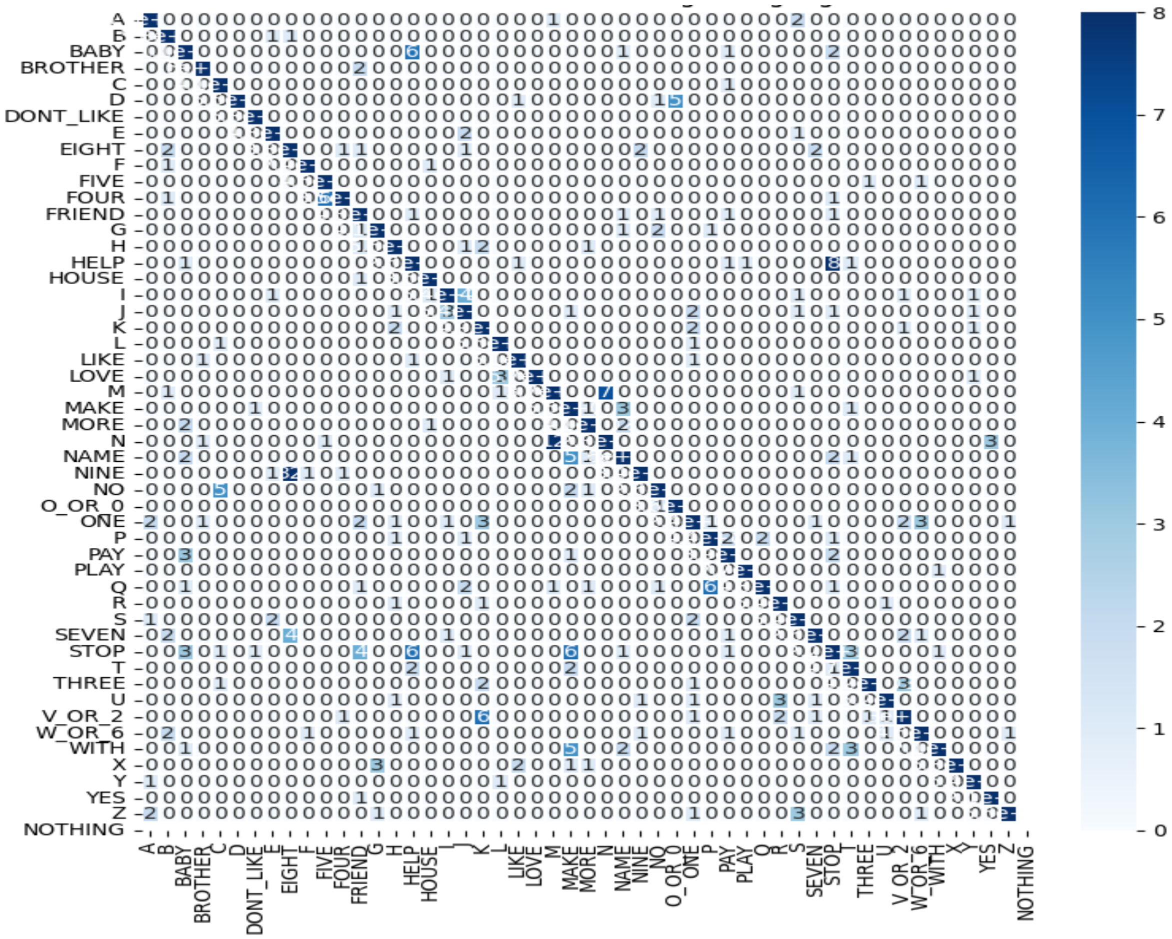


Fig. 7 Confusion Matrix for SVM Model

The output images captured for some of the real time inputs are shown below:

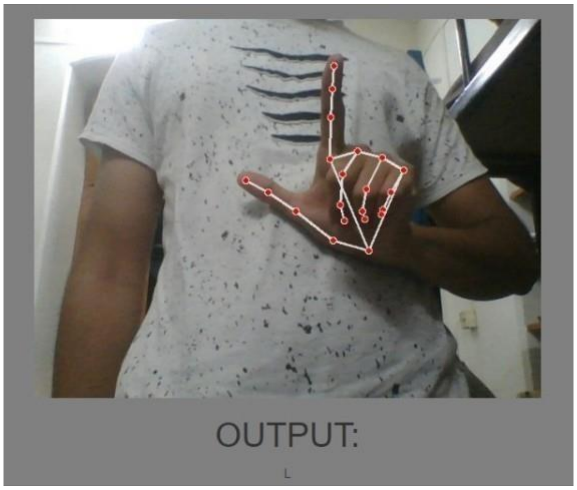


Fig. 8 Output – L

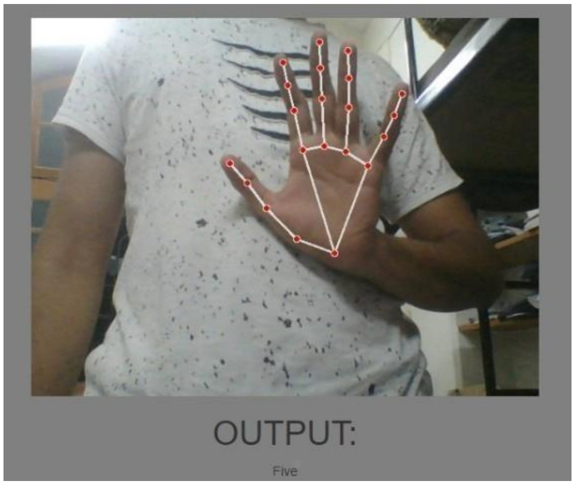


Fig. 9 Output - Five

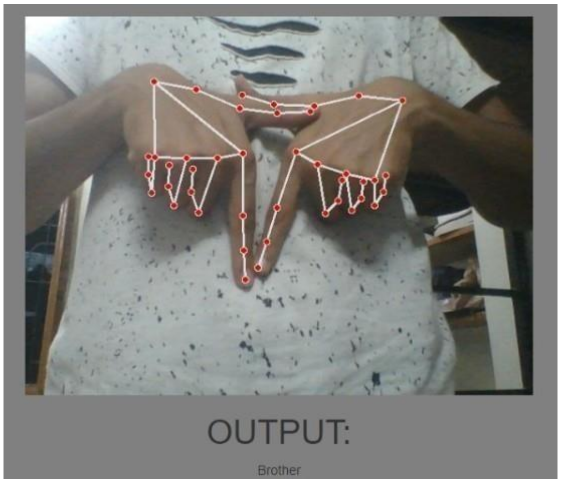


Fig. 10 Output - Brother

Results obtained in [12] and [5] are less accurate due to the use of ineffective feature extraction approaches and inappropriate models. Despite using the same dataset, accuracy mentioned in [22] is around 94.88%. Additionally, some of the research papers have attained an accuracy of about 99%, however these articles employ a small dataset with a small number of classes.

Our machine learning approach is suitable for use in mobile applications since the learned model is deliberately light. Our methodology's real-time sign language identification makes it quick, reliable, and especially flexible for smart devices. Mediapipe makes feature extraction simple by deconstructing and analyzing challenging hand-tracking data. This strategy uses less computer resources and takes less time to train the model than other cutting-edge approaches.

Table 1 Comparative Analysis of Accuracy for Various Models and Preprocessing Techniques with Simpler Datasets

Preprocessing & Algorithm	**Training Accuracy**	**Validation/Test Accuracy**
Convex Hull + CNN	99.54%	91%
Gaussian Blur + CNN	89.8%	91%
Gaussian Blur + VGG	84.66%	84.92%
Canny Edge Detection + VGG	93.71%	93.69%
Convex Hull + ResNet	94.03%	91.98%
Convex Hull + EfficientNet	90.68%	90%

This table compares the effectiveness of different preprocessing techniques and algorithms on our dataset and simpler datasets. It includes the preprocessing technique and algorithm used, along with the training accuracy and validation/test accuracy achieved by each technique. As shown in table 1, several preprocessing techniques were tested, including convex hull, Gaussian blur, and Canny edge detection. The algorithms used included CNN, VGG, ResNet, and EfficientNet. After analyzing the results from the experiments using the techniques and algorithms presented in the above table, we found that they did not yield satisfactory performance on our dataset. Therefore, we decided to discard these techniques and algorithms and explore other approaches to achieve better results.

Table 2 Performance Comparison with Similar Techniques

Type of Dataset	Our Accuracy	Existing/Others Accuracy
Alphabets only	**99.43%**	99.15% **[7]**
Alphabets, Numbers, and Words	**98.975%**	98.62% **[7]**

Based on our analysis, we could improve the accuracy of the model by adjusting the parameters. The improvement in accuracy was found to be around 0.28% to 0.35%. Our experiments also revealed that the model tends to overfit at higher values of C, and the choice boundary's curvature weight decreases with lower values of gamma. As a result, the areas separating different classes become more generic. After tuning the parameters, we were able to identify the optimal decision boundary for our dataset at C = 52 and gamma = 0.6.

6. Conclusion

Individuals with hearing disabilities often face significant challenges in communicating with people who can hear. One of the most effective ways for them to communicate is through sign language. However, for people who do not know sign language, understanding what is being communicated can be a significant challenge. This communication gap can have a detrimental impact on the social and emotional well-being of individuals with hearing disabilities, making it difficult for them to engage fully in society.

The proposed Sign Language Recognition system offers an innovative solution to the communication gap between individuals with hearing disabilities and those who can hear. The proposed system successfully recognizes sign language with high accuracy, with an SVM model achieving a classification accuracy of 98.975%. Moreover, the use of Google's MediaPipe palm detector method has made the system accessible to people without any special hardware, which is a significant advantage.

The proposed method's potential for practical applications is considerable, and it has the capacity to improve the quality of life for individuals with hearing disabilities, helping to bridge the communication gap between them and the rest of the world. Future work will expand the current system to add more indicators and create a complete and reliable system for mobile platforms. Additionally, the proposed method can be adapted for use in other Indian regional languages, such as Hindi, Marathi, Sindhi, Telugu, and more.

Although there are still some research gaps that need to be addressed, such as improving the system's accuracy in recognizing signs for complex phrases and developing a portable and affordable device for practical use in daily life, the proposed Sign Language Recognition system offers a promising step towards creating a more inclusive society. With further development and refinement, this system can play a significant role in breaking down communication barriers and facilitating greater accessibility and understanding for individuals with hearing disabilities.

CONFLICT OF INTEREST

The authors declare no conflict of interest.

REFERENCES

[1] Akoum, Alhussain, and Nour Al Mawla. "Hand gesture recognition approach for ASL language using hand extraction algorithm." Journal of Software Engineering and Applications 8.08 (2015): 419.

[2] Bandgar, Bapurao. "Implementation of Image Processing Tools for Real-Time Applications." International Journal of Engineering Research & Technology (IJERT) 10.07 (2021).

[3] Bazarevsky, Valentin, and G R Fan Zhang. "On-Device MediaPipe for Real-Time Hand Tracking." (2019).

[4] Brahmankar, Vipul, et al. "Indian Sign Language Recognition Using Canny Edge Detection." International Journal 10.3 (2021).

[5] Das, P., T. Ahmed, and M. F. Ali. "Static Hand Gesture Recognition for American Sign Language Using Deep Convolutional Neural Network." 2020 IEEE Region 10 Symposium (TENSYMP) (2020): 1762-1765.

[6] Devineau, G., F. Moutarde, W. Xi, and J. Yang. "Deep Learning for Hand Gesture Recognition on Skeletal Data." 2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018) (2018): 106-113.

[7] Halder, Arpita, and Akshit Tayade. "Real-Time Vernacular Sign Language Recognition Using MediaPipe and Machine Learning." International Journal of Recent Technology and Engineering (IJRTE) 10.2 (2021): 7421.

[8] Islam, M. M., S. Siddiqua, and J. Afnan. "Real-Time Hand Gesture Recognition Using Different Algorithms Based on American Sign Language." 2017 IEEE International Conference on Imaging, Vision & Pattern Recognition (icIVPR) (2017): 1-6.

[9] Jin, C. M., Z. Omar, and M. H. Jaward. "A Mobile Application of American Sign Language Translation via Image Processing Algorithms." 2016 IEEE Region 10 Symposium (TENSYMP) (2016): 104-109.

[10] Khan, Rafiqul Zaman, and Noor Adnan Ibraheem. "Hand Gesture Recognition: A Literature Review." International Journal of Artificial Intelligence & Applications 3.4 (2012): 161.

[11] Li, Y., X. Wang, W. Liu, and B. Feng. "Pose Anchor: A Single-Stage Hand Key Point Detection Network." IEEE Transactions on Circuits and Systems for Video Technology 30.7 (2020): 2104-2113.

[12] Martinez-Martin, Ester, and Francisco Morillas-Espejo. "Deep Learning Techniques for Spanish Sign Language Interpretation." Computational Intelligence and Neuroscience 2021 (2021).

[13] Pal, D. H., and S. M. Kakade. "Dynamic Hand Gesture Recognition Using Kinect Sensor." 2016 International Conference on Global Trends in Signal Processing, Information Computing and Communication (ICGTSPICC) (2016): 448-453.

[14] Raheja, J. L., Anand Mishra, and Ankit Chaudhary. "Indian Sign Language Recognition Using SVM." Pattern Recognition and Image Analysis 26.2 (2016): 434-441.

[15]. Shriram, S., Nagaraj, B., Jaya, J., Shankar, S., & Ajay, P. (2021). Deep Learning Based Real-Time AI Virtual Mouse System Using Computer Vision to Avoid COVID-19 Spread. Journal of Healthcare Engineering, 2021, 8133076.

[16]. Singha, J., & Das, K. (2013). Hand gesture recognition based on Karhunen-Loeve transform. arXiv preprint arXiv:1306.2599.

[17]. Trigueiros, P., Ribeiro, F., & Reis, L. P. (2012). A comparison of machine learning algorithms applied to hand

gesture recognition. In 7th Iberian Conference on Information Systems and Technologies (CISTI 2012) (pp. 1-6).

[18]. Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C. L., & Grundmann, M. (2020). MediaPipe Hands: On-device Real-time Hand Tracking. arXiv preprint arXiv:2006.10214.

[19]. Zivkovic, Z., & Van Der Heijden, F. (2006). Efficient adaptive density estimation per image pixel for the task of background subtraction. Pattern recognition letters, 27(7), 773-780.

[20].https://www.kaggle.com/datasets/belalelwikel/asl-and-some-words?select=ASL.

[21]. https://google.github.io/mediapipe/solutions/hands.html.

[22]. https://www.kaggle.com/code/vaishnaviasonawane/asl-recognition-model-training-revisited.