Journal of Computer Science

Original Research Paper

Efficient Detection of Palm and Hand Landmark for Speech Impaired People Using Mediapipe Model

[1]Sharmila Rathod, [2]Nilesh Rathod, [3]Nilesh Marathe, [4]Aruna Gawade, [5]Jyoti Kundale and [6]Nikita Kulkarni

[1]Department of Computer Engineering, Mcts Rajiv Gandhi Institute of Technology, Mumbai, India
[2]Department of AIML, Dwarkadas J. Sanghvi College of Engineering, Mumbai, India
[3]Department of CSDS, Dwarkadas J. Sanghvi College of Engineering, Mumbai, India
[4]Department of Computer Engineering, Dwarkdas J. Sanghvi College of Engineering, Mumbai, India, India
[5]Department of IT, Ramrao Adik Institute of Technology, D Y Patil Deemed to be University Navi Mumbai, India
[6]Department of Computer Engineering, KJ College of Engineering and Management Research, Pune, India

Article history
Received: 16-02-2024
Revised: 01-04-2024
Accepted: 22-05-2024

Corresponding Author:
Sharmila Rathod
Department of Computer Engineering, Mcts Rajiv Gandhi Institute of Technology, Mumbai, India
Email: sharmilagaikwad1709@gmail.com

Abstract: Human-machine interaction may be a basic figure in this age of touch-screen gadgets. Numerous gadgets are being created that can be worked without touching the system. So, in this consideration, how to function the framework utilizing signals instead of touching it appears. The point of this study is to create different communication procedures between humans and individual computers that would be required for individuals with engine impedances to take part in the data society. The paper elaborates a framework that will incorporate a hand signal acknowledgment approach for ASL dialect-sign dialect could be a strategy utilized by hard-of-hearing individuals for communication. This study may be, to begin with, a step towards building a conceivable sign dialect interpreter, to communicate in sign dialect and decipher it into composed verbal dialect. This is considered accomplished a useful hand signal acknowledgment framework for ASL communication, utilizing neural systems on webcam-captured pictures. This approach offers the potential for real-time ASL interpretation on common gadgets. The effect is significant, tending to communication challenges and cultivating deaf people.

Keywords: American Sign Language (ASL), Human-Computer Interaction (HCI), Neural Network (NN), World Federation of the Deaf (WFD), Support Vector Machine (SVM)

Introduction

Sign language is the most expressive way for people with hearing loss to communicate. It mostly consists of hand and arm gestures. In the development of gesture-based human-computer interaction systems, SLRs are crucial and prominent. Research on interactions and intelligent data processing has garnered increasing attention in statistics in recent years. In the context of today, computers are now an essential part of society for communication. Hand signals can be seen as an easy and practical way for normal and deaf individuals to communicate with each other because they are a potent human connection modality. There are already a number of techniques for recognizing hand gestures; the majority of them rely on neural networks, fuzzy logic, hidden Markov models, etc. Although the majority of the techniques are quite efficient, they come with a hefty computational cost. We created a different technique for American sign language hand motion recognition in order to get around this. A World Federation of the Deaf (WFD) poll indicates that 32 million children and 328 million adults worldwide- roughly 5% of the global population have hearing impairment. For example, the alphabets used in Italian and Indian sign languages differ significantly from those used in American Sign Language (ASL).

There are hence regional variations in sign language. Moreover, both single and double hands can be used to articulate important signals. With sign language's shortened version, words can be expressed with just one motion. These days, sign language also incorporates fingerspelling, which uses various signs to represent each letter of the alphabet to convey each word. Because many vocabulary terms in sign language dictionaries are still not standardized, finger-spelling is often employed to make a word manifest. Approximately 150,000 phrases used in spoken English still do not have an equivalent in ASL.

Problem Statement

People with speech impairments communicate using gestures and hand signs. It is tough for normal people to grasp them. Therefore, a system that can identify various signs and gestures and translate them into meaning for regular people is required. It creates a barrier between those with physical disabilities and the general public.

Objectives

This study's primary goal is to evolve the area of sign language recognition, with a particular emphasis on static gesture recognition. This study concentrated on using deep learning to recognize numerals 0-9 and 24 alphabets. The article describes a convolutional neural network classifier that has a high degree of accuracy in recognizing static sign language motions. The network is trained in a variety of configurations and the results are reported and analyzed. The outcome demonstrates that during training, accuracy increases when data from several subjects are used. The document also includes a basic Java GUI application for testing classifiers.

Literature Review

Supervised and unsupervised classification methods can be utilized to address the challenging issue of hand gesture identification in machine learning, enabling the detection of both static and moving hand motions. To detect movements of the hands with high precision, one requires a sophisticated method, a vast dataset, and intricate mathematical processing. Image processing is an important step in the gesture tracking process that entails getting images ready for analysis. We used the Google Mediapipe framework, which is available as open-source software, to precisely identify human body parts.

In a research paper, the identification of sign language is done by Halder and Tayade (2021) Google's Mediapipe. Additionally, the paper guarantees an average accuracy of 99% and claims that the suggested model is accurate, reliable, and effective. Without the need for wearable sensors, the Support Vector Machine (SVM) method provides a more convenient and accurate real-time detection (Halder and Tayade, 2021).

A study on neural network-based sign language recognition was published by Murakami and Taguchi (1991). People with physical disabilities are guided through the use of computer vision (Murakami and Taguchi, 1991).

The article written by Wang and Popović (2009) explains hand-tracking algorithms with colorful gloves. Wang and Popović (2009) using the K-Nearest Neighbors (KNN) technique, which necessitates a constant feeding of hand streams, the color pattern recognition of gloves is presented. But Kurdyumov *et al.* (2011); Tharwat *et al.*, 2015; Gaikwad *et al.* (2021); Rekha *et al.* (2011) propose the Support Vector Mechanism (SVM) algorithm for their research findings.

Elakkiya *et al.* (2012) describe a framework for subunit recognition of alphabets using a mixture of two algorithms: SVM learning and boosting algorithm. Although 26 alphabets are not predicted by the system, a 97.6% accuracy rate was achieved. Baranwal and Nandi (2017); Ahmed and Aly (2014) combined PCA with local binary patterns to extract characteristics from 23 distinct Arabic sign languages. The system's usage of threshold operators prevents it from recognizing the consistent grey-scale patterns in the signature region, even with a 99.97% accuracy rate in signer-dependent mode. In machine learning, hand gesture detection is a difficult topic. The majority of early attempts to discern hand gestures from image frames use a standard convolutional network (Elakkiya *et al*., 2012).

A consider by Sharma *et al*. (2024) on preparing a machine learning demonstration for recognizing numeric signs utilizing 80,000 person signs and over 500 pictures per sign. The strategy included making a database for a hand-detection framework and a motion acknowledgment framework. Picture pre-processing was performed utilizing highlight extraction to normalize the input data sometime recently while preparing the demonstration. The pictures were changed over to grayscale to move forward question form and a standardized determination was utilized. Also, the pictures were smoothed into a smaller number of one-dimensional components. Highlight extraction made a difference in extricating important highlights around the pixel information from the pictures and nourishing them to a CNN, driving more exact expectations and simpler preparation (Ahmed and Aly, 2014).

Hand tracking has been done in two and three dimensions by Liu *et al*. (2015) With the use of skin saliency, which, for better feature extraction, extracts skin tones within a range, they were able to reach a classification accuracy of about 98% (Sharma *et al*., 2024).

A paper published by Pavlovic *et al*. (1997) compares different approaches to recognizing hand movements in order to communicate with computers. The paper notes the advantages and limitations of using either a 3D model or a human-like hand's appearance. While 3D models offer more detailed modeling of hand gestures, they present computational tasks that make simultaneous processing for human-computer interaction difficult. This article also discusses examples of implemented gestural systems and prospective uses for gesture recognition based on vision (Pavlovic *et al*., 1997).

Vogler and Metaxas (2004) in Parallel Covered up Markov Models (PaHMMs) to recognize ASL. They claimed that in a continuous acknowledgment system, phonemes can be used in place of entire signs. In the paper, one can understand that words can be dissected into their component phonemes in the same way as they are recognized in speech and utilized both of the available channels appropriately and without using their hands. An accuracy rate of 87.88% was observed when a channel of the Well demonstration was used for a limited vocabulary, such as recognizing 22 signs. The structure does not account for a greater vocabulary, hand positioning, introduction, or facial emotions (Thakur *et al*., 2020; Vogler and Metaxas, 2004).

In their research, explored the usage of microcontroller systems for speech detection and translation from sign language. Every time a sign dialect move was detected, the suggested framework would play a recorded voice. Four components make up the planned show: A voice capacity unit, a detecting unit, a preparing unit, and a remote communication unit. The PIC16F877A was used to connect the flux sensor and APR9600. Gloves with built-in flux sensors respond to a signal. Through the use of an appropriate circuit, the sensor's response was provided to the microcontroller, which then used the APR9600 to play back the recorded voice. This framework promised swift response times and towering, unwavering quality.

The article published by Kalidolda and Sandygulova (2018) depicted a study, that pointed to translating the mechanical framework of sign dialect. The paper aimed to encourage fingerspelling acknowledgment in real-time and in the wild and to urge deaf-mute persons to criticize the NAO mechanical deciphering framework. The automated framework consists of a number of hardware and software components, such as Jump Movement SDK for hand position and tracking and Microsoft Kinect SDK for human detection and tracking. A stationary NAO robot in the humanoid form serves as a common crossing point. Furthermore, NAO adopts the part of an automated mentor for kids, as a social interface, a human stationary pepper robot, Applications for coaching are run on an Android tablet and a computer screen for the virtual robot (Kalidolda and Sandygulova, 2018; Thakur *et al*., 2020).

The purpose of this research is to discuss the various effective deaf-mute communication translation frameworks and their deaf-quiet communication mediators. The two main types of communication aids that deaf-mute people employ are wearable technology and online learning frameworks. Underneath the wearable communication approach are the Handicom touch-screen, keypad strategy, and glove-based framework. The three sub-partitioned techniques mentioned above accelerometer, the appropriate microcontroller, the content-to-discourse transition module, the keypad, and the touch screen all make use of different sensors. The moment technique, or online learning framework, can eliminate the need for an external device to interpret messages between hard of hearing silent people and non-deaf-mute people.

A productive system for Indian sign dialect acknowledgment utilizing wavelet change. As a strategy for design acknowledgment, the ISLR framework incorporates the two vital modules of extraction and classification. Discrete wavelet changes highlight extraction and NN classifier are coupled to comprehend sign dialect. The preliminary findings demonstrate that the suggested hand gesture acknowledgment framework, when employing a cosine remove classifier, accomplishes the greatest classification precision of 99.23% (Rajesh *et al*., 2023).

Hand signal recognition with PCA. In addition to compelling layout coordination, the authors of this study presented a scheme for database-driven hand motion recognition based on skin color display and thresholding, which may be applied to mechanical and human technology applications as well as comparative other applications. First, the skin tone representation in the YCbCr color space is used to divide the hand region. Thresholding is utilized within the taking after organizing to partition the frontal area and foundation. At last, for acknowledgment, a template-based coordinating procedure is created utilizing foremost component investigation (PCA) (Ahuja and Singh, 2015; Pandey and Jain, 2015; Rathod and Wankhade, 2022a-b; Thakur *et al*., 2020).

Proposed System

The proposed System has the main component as the Mediapipe model i.e., an open-source framework from Google that handles palm detection and hand landmark detection efficiently. MediaPipe hands employments an ML pipeline created with a few interconnected models, a show for recognizing palms that employments the whole picture and produces a leaning hand bounding box a hand point of interest shows that runs on the palm detector cropped picture locale and yields exceedingly precise 3D hand key focuses. With this approach, we can obtain 21 key points of the hand and at most 100 photos per sign in order to create a database. i.e., [x, y, z] are its coordinates. The width and stature of the picture normalize values for x and y to [0.0, 1.0].

Limitation Existing System or Research Gap

The existing systems for hand gesture recognition face several limitations that can hinder their performance and accuracy. Some of the limitations include.

It is not feasible to manage the video data inside the constrained GPU memory. The majority of CNN techniques only use movies with additional temporal information that are image-based. A simple scaling approach could cause the loss of crucial temporal information if the fine-tuning and classification are done on each frame independently.

The SLR is weak because of ambiguities and a lack of training datasets. As a result, standardized, large-scale datasets with both manual and automatic features are crucial.

Neural computing has some restrictions. The neural network's inability to effectively describe the model it has created is its main drawback. Analysts frequently inquire as to the reason behind the model's behavior. Although neural networks produce better results, they struggle to explain how they arrived at those results. First, deriving rules from neural networks is challenging. When someone must explain their response to someone else or has worked with artificial intelligence, especially rule-based expert systems, this can be significant. Neural computing performance heavily relies on the representativeness of the data used. Poor quality input data will inevitably lead to inadequate output. Furthermore, training a model from a complex data set can be a time-consuming process. The accuracy of a vision-based system can be compromised due to reduced resolution resulting from poor camera quality. Furthermore, it is important for the device to be reliable, cost-effective, and easy to maintain. Maintaining an optimal distance between the signer and the camera is crucial as being too close or too far can have a negative impact on the performance of the system. The computational demand of neural techniques makes them slower on low-end machines without math co-processors. However, it is crucial to keep in mind that the overall time required to achieve results could still be faster compared to other data analysis approaches, despite the longer time taken for training the system. Another obstacle is the incorrect camera setup, which can result in a loss of crucial sign information when a sign is either static or dynamic, depending on the signer’s performance.

The performance of neural networks is not solely determined by their processing speed. Additionally, unlike other analytical approaches, developing neural networks does not require as much time to program, debug, or test assumptions. The rapid and continuous execution of sign language gestures by the signer can pose difficulties in accurately segmenting and extracting relevant features, resulting in a break between the letters or signs.

There are several challenges in sign language recognition, including hand and face occlusion, overlapping gestures, and variations in signs between different signers and contexts. The ability to recognize and classify is influenced by factors such as location and illumination. Choosing appropriate loss functions during training can be costly. Using larger batch sizes may result in local convergence rather than global convergence, whereas smaller batch sizes lead to longer training iterations and increased training costs.

One area of active research is the creation of AI-based models for realistic sign language translation and the production of avatar models, incorporating both manual and non-manual gestures. Another trend in this field is the development of web-based or smartphone applications for sign language learning and translation using AI.

Analysis

Most usage encompassing this assignment has endeavored it through exchange learning, The common design comprises numerous dropout layers and dense layers. The design included an input layer followed by a dropout layer a dense layer and a dropout layer followed by two last dense layers (Gaikwad *et al*., 2019).

Algorithm

SVM is a well-known machine learning calculation utilized for classification issues. In hand gesture recognition, SVM can be used to classify the extracted features of a hand gesture. SVM finds a hyperplane that isolates the classes and maximizes the edge between them. KNN could be a non-parametric calculation that can be utilized for both classification and relapse. KNN can be connected to hand signal acknowledgment by comparing the extricated highlights of a signal with the k-nearest neighbors within the preparing information and classifying the signal based on the larger part course of those neighbors.

CNN is a deep learning algorithm that can be used for the classification of an image. In hand gesture recognition, CNN can be trained on the raw image data of hand gestures and it can learn to recognize highlights and designs within the pictures that are important to classifying the gestures. Principal component investigation (PCA) may be a procedure utilized for dimensionality lessening. In hand motion acknowledgment, use of PCA for reducing the dimensionality of the extracted features of a gesture, while retaining the utmost vital information. This can help to improve the accuracy and efficiency of the classification algorithm. Hidden Markov Models (HMM) and Partially Observed Hidden Markov Models (PAHMM) are probabilistic models that can be used for gesture recognition. HMM can be used to model the temporal sequence of features in a gesture and classify it based on the likelihood of the sequence. PAHMM is a more complex version of HMM that can handle incomplete or missing data, which is often the case in real-world applications. Multilayer Perceptron (MLP) is a type of artificial neural network that can be used for classification problems. In hand gesture recognition, MLP can be trained on the extracted features of a gesture and it can learn to classify the gesture based on the patterns in the features. MLP is particularly useful for non-linearly separable classes, where a simple linear classifier like SVM may not be effective (Ma *et al*., 2000).

Design Details

A palm detection model and a land landmark model that depend on one another form the backbone of the ML pipeline for the hand detection solution. The landmark model is then given the precisely cropped palm image that was provided by the palm detection model. With this method, deep learning models use less data augmentation (such as rotations, flips, and scaling) and focus more of their processing resources on landmark localization. The conventional method is to locate landmarks over the current frame after detecting the hand from the frame. However, this palm detector uses a different approach to overcome ML pipeline issues. Following hand landmark models enter the scene when the palm detection scans the entire image frame. In the observed hand regions, this model accurately localized 21 3D hand-knuckle coordinates (i.e., x, y, and z-axis). The model is so adept at detecting hands and reliable in doing so that it even assigns coordinates to hands that are only half visible. Now that the palm and hand identification model is operational, it is applied to a dataset of diverse languages. The alphabet from A when using the dataset for American sign language. Therefore, the hand detection on each alphabet folder containing photographs using this detection model is run. The acquired landmark points are then saved in a CSV file. Extraction of the landmark points is carried out simultaneously with an elimination task. For training the ML model in this case, just the x and y coordinates discovered by the hand landmark model are taken into account. Figure 1 shows a hand landmark using Mediapipe.

The duration of landmark extraction varies based on the dataset size, ranging from 10-15 min. To combine all the data points into a single file, each image in the dataset is put through stage 1, which only considers the detector's x and y coordinates. The panda library function is then used to scrape this file in order to look for any null entries. Sometimes a fuzzy image prevents the detector from identifying the hand, which causes a null entry to be added to the dataset. Therefore, in order to prevent bias when developing the prediction model, these issues need to be resolved. The table's rows containing these null items are located and removed using their indexes. We removed unnecessary areas from the x and y coordinates before normalizing them to fit within the system. The data file should be ready to be divided into training and validation sets later. Eighty percent of the data is used to train the model, with twenty percent reserved for model validation. using various optimization and loss functions. Machine learning techniques were used to undertake predictive analysis of several sign languages, with Deep Neural Networks (DNN) outperforming other systems. Layers of neural networks are stacked along the depth and width of lesser structures to create deep neural networks, a potent class of machine learning algorithms. A computer's DNN sorts data according to its components, such as a sound's pitch, as it receives input from the user.

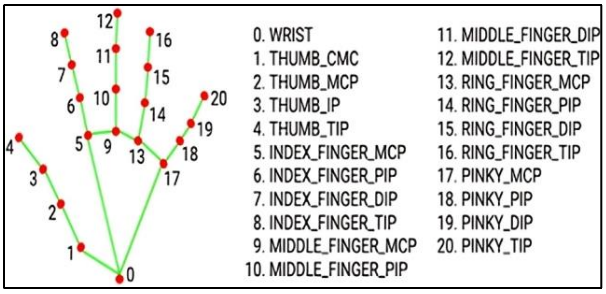
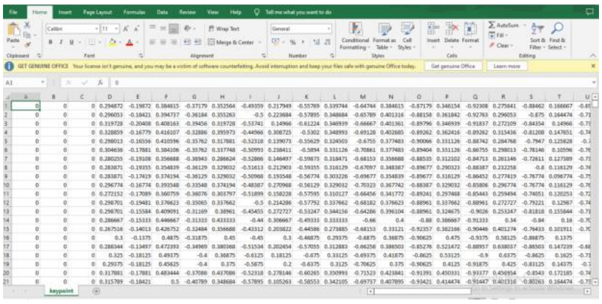


Fig. 1: Hand landmark using Mediapipe



GET GENUINE OFFICE Your license isn't genuine, and you may be a victim of software counterfeiting. Avoid interruption and keep your files safe with genuine Office today.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	0	0	0	0.294872	-0.19872	0.384615	-0.37179	0.352564	-0.49359	0.217949	-0.55769	0.339744	-0.64744	0.384615	-0.87179	0.346154	-0.92308	0.275641	-0.88462	0.166667	-0.65
2	0	0	0	0.296053	-0.18421	0.394737	-0.36184	0.355263	-0.5	0.223684	-0.57895	0.348684	-0.65789	0.401316	-0.88158	0.361842	-0.92763	0.296053	-0.875	0.164474	-0.71
3	0	0	0	0.319728	-0.20408	0.408163	-0.39456	0.319728	-0.53741	0.14966	-0.61224	0.346939	-0.66667	0.401361	-0.89796	0.346939	-0.91837	0.272109	-0.84354	0.14966	-0.73
4	0	0	0	0.328859	-0.16779	0.416107	-0.32886	0.395973	-0.44966	0.308725	-0.5302	0.348993	-0.69128	0.402685	-0.89262	0.362416	-0.89262	0.315436	-0.81208	0.147651	-0.74
5	0	0	0	0.298013	-0.16556	0.410596	-0.35762	0.317881	-0.52318	0.139073	-0.55629	0.324503	-0.6755	0.377483	-0.90066	0.331126	-0.88742	0.284768	-0.7947	0.125828	-0.7
6	0	0	0	0.304636	-0.17881	0.384106	-0.35762	0.337748	-0.50993	0.238411	-0.5894	0.331126	-0.70861	0.377483	-0.89404	0.331126	-0.86755	0.298013	-0.78146	0.10596	-0.76
7	0	0	0	0.280255	-0.19108	0.356688	-0.38943	0.286624	-0.52866	0.146497	-0.59873	0.318471	-0.68153	0.356688	-0.88535	0.312102	-0.84713	0.261146	-0.72611	0.127389	-0.71
8	0	0	0	0.283871	-0.19355	0.354839	-0.36129	0.329032	-0.51613	0.212903	-0.59355	0.316129	-0.67097	0.348387	-0.89677	0.290323	-0.88387	0.232258	-0.8	0.116129	-0.74
9	0	0	0	0.283871	-0.17419	0.374194	-0.36129	0.329032	-0.50968	0.193548	-0.56774	0.303226	-0.69677	0.354839	-0.89677	0.316129	-0.86452	0.277419	-0.76774	0.096774	-0.75
10	0	0	0	0.296774	-0.16774	0.393548	-0.33548	0.374194	-0.48387	0.270968	-0.56129	0.329032	-0.70323	0.367742	-0.88387	0.329032	-0.85806	0.296774	-0.76774	0.116129	-0.76
11	0	0	0	0.272152	-0.17089	0.360759	-0.36076	0.303797	-0.51899	0.158228	-0.57595	0.310127	-0.66456	0.341772	-0.89241	0.297468	-0.85443	0.259494	-0.74051	0.120253	-0.72
12	0	0	0	0.298701	-0.19481	0.376623	-0.35065	0.337662	-0.5	0.214286	-0.57792	0.337662	-0.68182	0.376623	-0.88961	0.337662	-0.88961	0.272727	-0.79221	0.12987	-0.74
13	0	0	0	0.298701	-0.15584	0.409091	-0.31169	0.38961	-0.45455	0.272727	-0.53247	0.344156	-0.64286	0.396104	-0.88961	0.324675	-0.9026	0.253247	-0.81818	0.155844	-0.71
14	0	0	0	0.286667	-0.15333	0.446667	-0.31333	0.433333	-0.44	0.306667	-0.49333	0.333333	-0.66	0.4	-0.88	0.386667	-0.91333	0.34	-0.84	0.16	-0.70
15	0	0	0	0.267516	-0.14013	0.426752	-0.32484	0.356688	-0.43312	0.203822	-0.44586	0.273885	-0.68153	0.33121	-0.92357	0.382166	-0.90446	0.401274	-0.76433	0.101911	-0.70
16	0	0	0	0.3	-0.1375	0.4875	-0.31875	0.45	-0.45	0.3	-0.46875	0.29375	-0.6875	0.36875	-0.90625	0.475	-0.9375	0.58125	-0.88875	0.1375	
17	0	0	0	0.288344	-0.13497	0.472393	-0.34969	0.380368	-0.51534	0.202454	-0.57055	0.312883	-0.66258	0.386503	-0.85276	0.521472	-0.88957	0.638037	-0.86503	0.147239	-0.68
18	0	0	0	0.325	-0.18125	0.49375	-0.4	0.36875	-0.6125	0.18125	-0.675	0.33125	-0.69375	0.41875	-0.8625	0.53125	-0.9	0.6375	-0.8625	0.1625	-0.71
19	0	0	0	0.29375	-0.18125	0.45625	-0.4	0.375	-0.5875	0.2	-0.6375	0.3125	-0.70625	0.375	-0.90625	0.4125	-0.91875	0.425	-0.83125	0.14375	-0.7
20	0	0	0	0.317881	-0.17881	0.483444	-0.37086	0.437086	-0.52318	0.278146	-0.60265	0.350993	-0.71523	0.423841	-0.91391	0.450331	-0.93377	0.456954	-0.8543	0.172185	-0.74
21	0	0	0	0.315789	-0.18421	0.5	-0.40789	0.348684	-0.57895	0.105263	-0.58553	0.342105	-0.69737	0.407895	-0.93421	0.414474	-0.91447	0.401316	-0.80263	0.164474	-0.75

Fig. 2: Training the model

Experimental Setup

In this study, the data refers to the captured coordinates of hand landmarks during sign language gestures. These coordinates are obtained using the MediaPipe library and are stored in a CSV file format. The data is then used to train the machine learning model to recognize and classify the sign language gestures. Additionally, the video and audio data captured during the video chat feature of the website are also considered as data. Figure 2 represents the Python code for training the model. Images of 1000 distinct object categories, including mice, pens, keyboards, and other animals, can be classified by the pre-trained network. Consequently, a vast array of image-rich feature representations have been trained into the network. The network is capable of processing pictures with an input size of 224×224. Figure 2 shows a training of the model.

Required Python Libraries

- Keras: Google API for easy implementation of neural network
- Tensorflow: An open-source library to create machine learning models easily
- OpenCV: It is an opensource library of computer vision
- Flask: It is a microweb framework, doesn't need any library also no need of database
- MediaPipe: It is a suite of libraries for quick enabling of machine learning techniques

Materials and Methods

Following is the methodology used in the paper, which provides an efficient way to solve the problem.

Stage 1: With the MediaPipe framework, programmers may create multi-modal (video, audio, and any time series data) cross-platform applied machine learning pipelines. All of the human body identification and tracking models available on MediaPipe have been trained on the biggest and most varied dataset available on Google. They serve as the framework for nodes, edges, or markers, tracking important places on various body parts. In three dimensions, each coordinate point has been normalized. TensorFlow lite models developed by Google developers simplify the process of adjusting and changing information flow. Nodes on a graph, which make up MediaPipe, are frequently provided in a pb.txt file. C++ files are connected to these nodes. An extension of these files is the MediaPipe basic calculator class. This class receives contracts for media streams from other nodes in the graph and guarantees that it is connected, just as a video stream. Once the other nodes in the pipeline are connected, the class produces its own output of the processed data. Each stream of information is sent to each calculator using packet objects that can store a wide variety of information. Side pockets, where a calculator node can be added with auxiliary information like constants or static characteristics, can also be forced onto a graph. This streamlined pipeline structure makes it simple to add or modify components and it also makes it easier to accurately regulate the direction of data flow. Two interdependent models compose the backend machine learning pipeline of the hand detection system. Model for palm detection, the first model of a physical landmark. The landmark model is then given the precisely cropped palm image that was provided by the palm detection model. With this method, deep learning models use less data augmentation (such as rotations, flips, and scaling) and focus more of their processing resources on landmark localization. The conventional method is to locate landmarks over the current frame after detecting the hand from the frame. However, the proposed system's key component is the Mediapipe model, an open-source framework from Google that effectively tackles palm detection and hand landmark detection, in this palm detector using ML pipeline problems with a new approach. An ML pipeline used by MediaPipe hands is composed of numerous connected models: A palm detection model that creates an aligned hand bounding box using the entire image a hand landmark model that generates very precise 3D hand key points while operating on the portion of the image that was clipped by the palm detector. With this method, we can collect up to 100 images per sign and 21 critical spots on the hand in order to build a database. Specifically, its coordinates are [x, y, z]. Because you have to perform image processing, thresholding, and work with a range of hand sizes, detecting hands is a time-consuming process. First, a trained palm detector estimates bounding boxes around rigid objects like palms. Second, a larger scene context is extracted using an encoder-decoder. Subsequent hand landmark models enter the scene once the palm detection scans the entire image frame. The model is so adept at detecting hands and reliable in doing so that it even assigns coordinates to hands that are only half visible. The hand landmark model's 21 identified landmark locations. Now that the palm and hand identification model is operational, it is applied to datasets of different languages. This article discusses that the alphabet from a-z when using the dataset for American sign language. Therefore, we run hand detection on each alphabet folder containing photographs using this detection model, which gives us 21 landmark points. The acquired landmark points are then saved in a CSV file. Extraction of the landmark points is carried out simultaneously with an elimination task. For training the ML model in this case, just the x and y coordinates discovered by the hand landmark model are taken into account. Landmark extraction takes between 10 and 15 min, depending on the size of the dataset.

Stage 2: Similar to step 1, the detector's x and y coordinates are gathered and saved in a file. After that, the data is examined for null entries brought about by hazy photos, and the panda's library is used to eliminate these rows. After normalizing the residual data to match the model, the file is divided into a validation set (25%) and a training set (75%). After that, the model is trained using a variety of optimization and loss functions, and the validation set is used to assess its performance. The data-cleaning process is necessary to avoid biases in the model caused by the presence of null entries, which can be caused by blurry images (Halder and Tayade, 2021).

Stage 3: The model architecture consists of three dense layers, as shown in Figure 3 with the first two layers having a dropout layer for regularization. The input shape of the model is (21 * 2,) which means it expects a one-dimensional input vector of length 42. ReLU is the activation function used in the first two layers, while SoftMax activation is utilized in the last layer because the problem is a multi-class classification (NUM_CLASSES is the number of classes). Consequently, this model can be applied to classification problems as a Multi-Layer Perceptron (MLP). MLPs are a particular kind of neural network that can be used for supervised learning, which is the process of using labeled data to train a model so that it can predict new, unseen data. MLPs are particularly powerful because they can learn complex, non-linear mappings between inputs and outputs, making them useful for a wide range of tasks.

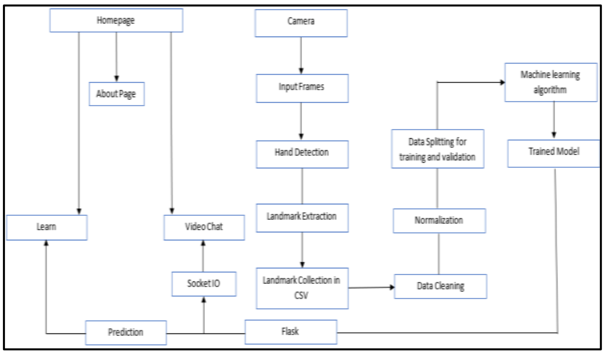


Fig. 3: System architecture

Stage 4: Flask is a lightweight web application framework for WSGI, which can be scaled up to complex applications. It is quick and simple to start. Here, flask helps as the framework, and flask itself is a Python class data type. Stated differently, Flask serves as the prototype for building instances of web apps-or, to put it more simply, web applications. The home function will run when the user navigates to localhost: 5000 and it will return its result on the webpage. The function output would be displayed when the visitor visited localhost: 5000/about/if the route method had received a different input, such as "/about/."

Performance Evaluation

Classification Report

Figure 4 shows a classification report used to measure the degree of quality predictions.

Precision is the ability of a classifier to accurately classify or to avoid labeling an occurrence as positive when it is actually negative. Its definition is the proportion of true positives to all false positives in each class.

Precision-accuracy of positive predictions can be calculated as follows:

$$Precision = TP\left(TP / FP\right) \qquad (1)$$

FP = False Positives
TP = True Positives

The parameter known as recall measures how often a machine learning model correctly chooses positive examples, or true positives, from among all of the dataset's actual positive samples:

$$Recall = TP / \left(TP + FN\right) \qquad (2)$$

Recall = Fraction of positives that were correctly identified
FN = False Negatives

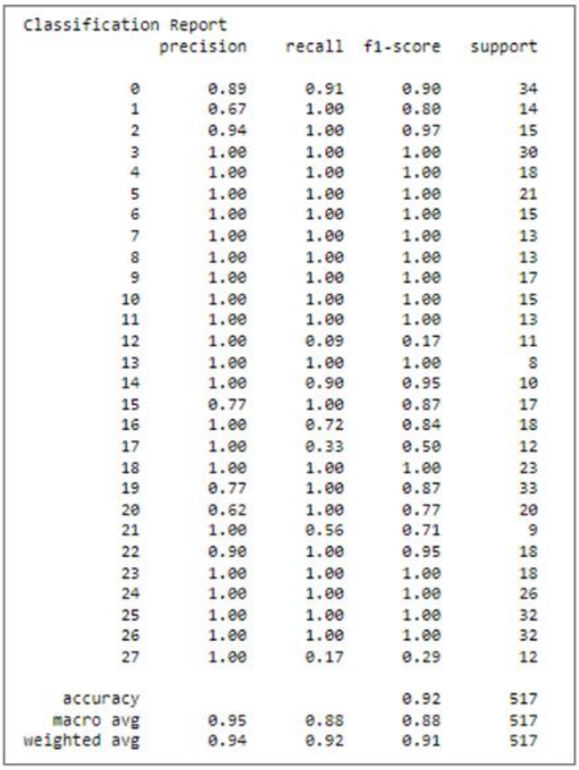
The *F1 score* is a weighted average of recall and precision, with a maximum score of 1.0 and a minimum value of 0.0. Since *F1 scores* account for memory and precision in their computations, they are less than direct measures. As a general rule, the weighted mean of *F1* should be used for comparing classification models, rather than for absolute accuracy:

$$F1\text{-}score = 2 * \left(Recall * Precision\right) / \left(Recall + Precision\right) \quad (3)$$

Confusion Matrix

An overview of the expected outcomes for a classification task is called a confusion matrix.

The confusion matrix in Fig. 5 for efficient detection of palm and hand landmarks for speech-impaired people using the Mediapipe model.

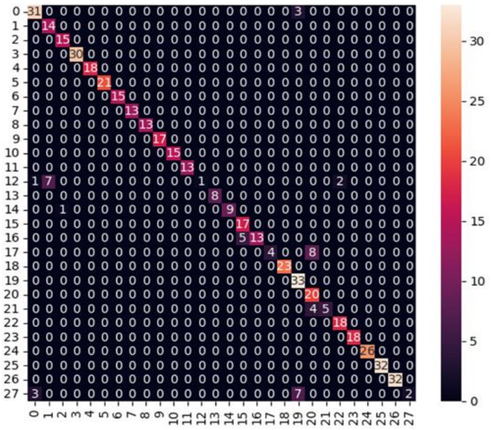


```
Classification Report
              precision    recall  f1-score   support

           0       0.89      0.91      0.90        34
           1       0.67      1.00      0.80        14
           2       0.94      1.00      0.97        15
           3       1.00      1.00      1.00        30
           4       1.00      1.00      1.00        18
           5       1.00      1.00      1.00        21
           6       1.00      1.00      1.00        15
           7       1.00      1.00      1.00        13
           8       1.00      1.00      1.00        13
           9       1.00      1.00      1.00        17
          10       1.00      1.00      1.00        15
          11       1.00      1.00      1.00        13
          12       1.00      0.09      0.17        11
          13       1.00      1.00      1.00         8
          14       1.00      0.90      0.95        10
          15       0.77      1.00      0.87        17
          16       1.00      0.72      0.84        18
          17       1.00      0.33      0.50        12
          18       1.00      1.00      1.00        23
          19       0.77      1.00      0.87        33
          20       0.62      1.00      0.77        20
          21       1.00      0.56      0.71         9
          22       0.90      1.00      0.95        18
          23       1.00      1.00      1.00        18
          24       1.00      1.00      1.00        26
          25       1.00      1.00      1.00        32
          26       1.00      1.00      1.00        32
          27       1.00      0.17      0.29        12

    accuracy                           0.92       517
   macro avg       0.95      0.88      0.88       517
weighted avg       0.94      0.92      0.91       517
```

Fig. 4: Classification report



	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27
0	31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0
1	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	1	7	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	1	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	13	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	8	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	23	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	20	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	5	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	18	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	26	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	32	0
27	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	2

Fig. 5: Confusion matrix

Model Accuracy

Model accuracy is the model correctly predicts the predictions made. For assessing the performance of a model, we can use a graph as shown in Fig. 6.

Accuracy = Number of correct predictions/total number of predictions.

This graph shows the relation between accuracy and the epoch. As the number of iterations for training the model is increasing the accuracy is also increasing. so, we can say that the accuracy and the epoch are directly proportional to each other.

Model Loss

Model loss is a metric that expresses how poorly the model predicted a specific case. If the model works efficiently then the loss is zero, Fig. 7 illustrates the Model loss for the model:

$$Loss = 1 - Accuracy \quad (4)$$

This graph shows the relationship between the epoch and loss; the y-axis represents the loss and the x-axis represents the epoch. As the total number of iterations for training the model is increasing the loss is decreasing. So, we can say that epoch and loss are inversely proportional with respect to each other.

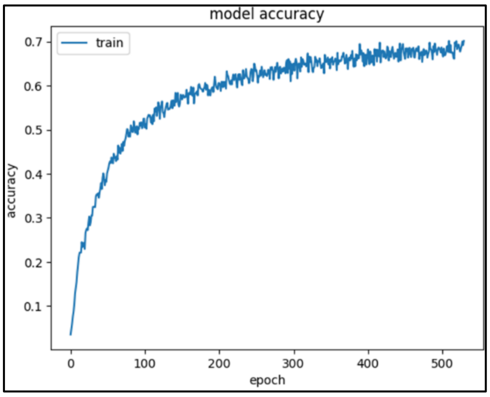


Fig. 6: Model accuracy

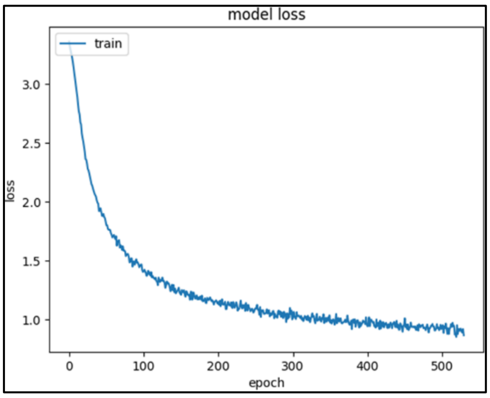


Fig. 7: Model loss

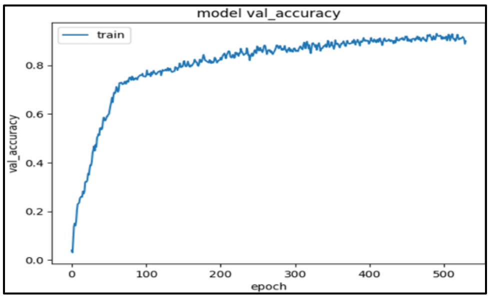


Fig. 8: Model validation accuracy

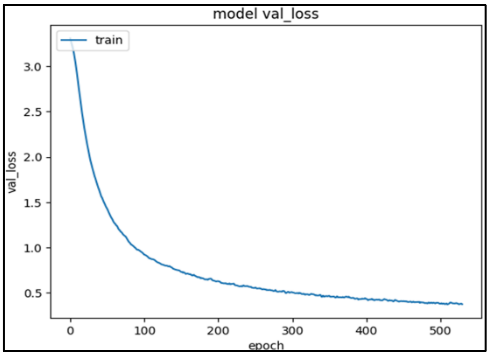


Fig. 9: Model validation loss

Model Validation Accuracy

Accuracy is one of the most critical parameters in method validation. Accuracy confirms the suitability of the method to the greatest extent and hence method developers must design suitable extraction procedures to ensure accurate quantification of the analyte in the presence of a sample matrix.

This graph i.e., Fig. 8 is between the epoch and the validation accuracy. The y-axis represents validation accuracy whereas the x-axis represents the epoch.

Model Validation Loss

A deep learning model's performance on the validation set is assessed using a metric known as validation loss.

This is the graph i.e., Fig. 9. between validation loss and epoch for the validation set of data, where epoch means the total number of iterations for validating the ML model with all the validating data in one cycle. In this graph, as the epoch is increasing which means the iterations for validation are increasing then there is a validation loss.

Result Analysis

To train the model we use layers like dense layer, dropout layer, etc. Dropout is connected between the two hidden layers and between the final hidden layer and the output layer. Once more, a dropout rate of 40% is utilized as may be a weight imperative on those layers.

Development of a machine learning model: To train our machine learning model, we employed the MLP method. An input layer, one or more hidden layers, and an output layer are the three layers of nodes that make up the MLP, a particular kind of feedforward neural network. The normalized x, y, and z coordinates of the hand landmarks are fed into the input layer. For the input gesture, the output layer generates the expected class label. Using the TensorFlow and Keras frameworks in Jupiter Notebook, we created our MLP model. The model architecture was composed of two dense layers of 10 nodes each with the activation functions of SoftMax and Rectified Linear Unit (REL) and another dropout layer with a rate of 0.4. The input layer was followed by a dropout layer with a rate of 0.2 and two dense layers of 20 nodes each. By randomly eliminating some nodes during training, the dropout layers assisted in lowering overfitting. To find the output of a dense layer, use the formula:

$$output = activation\left(dot\left(input, weights\right) + bias\right) \tag{5}$$

where input is the input to the layer, weights are the weights of the layer, bias is the bias vector of the layer and activation is the activation function applied element-wise to the output. The formula for calculating the output of a dropout layer is:

$$output = input * mask \tag{6}$$

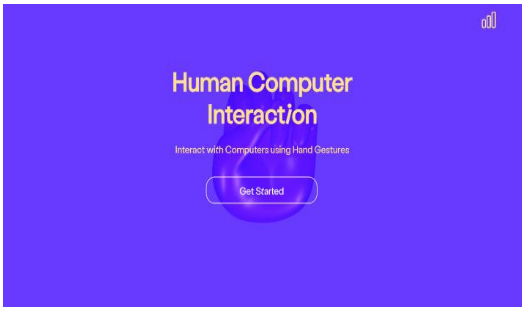
where input is the input to the layer and mask is a binary mask generated randomly for each training batch. The mask zeroes out a fraction of the input values to encourage the network to learn more robust features.

The Adam optimizer, a stochastic gradient descent optimization algorithm that calculates adaptive learning rates for each parameter, was utilized to compile the model. The "sparse-categorical-crossentropy" loss function was chosen because it works well for multi-class classification issues. The model's performance was gauged during training and assessment using the accuracy metric.

The website's introduction page is displayed in Fig. 10. This page provides an overview of the website's content. For additional information about the webpage, the viewer can visit the "About us" page. CSS is used to add the gradient that is utilized in the backdrop.

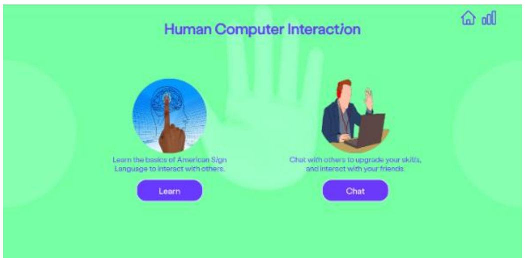
Figure 11 shows a page where users can learn the language as well as further interact with others. The user can also chat with others to upgrade their skills and interact with friends.

Figure 12 shows the selection of the dominant hand which they are going to use to perform the gestures for American sign language. The programming elements used to design this webpage are flexbox from CSS on which the Hoover effect was applied.


Human Computer Interaction

Interact with Computers using Hand Gestures

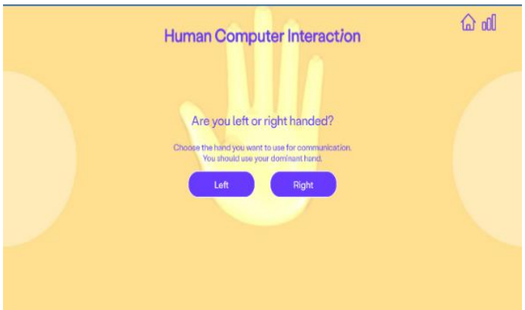
Fig. 10: Home screen


Human Computer Interaction

Learn the basics of American Sign Language to interact with others.

Chat with others to upgrade your skills, and interact with your friends.

Fig. 11: Learn and chat


Human Computer Interaction

Are you left or right handed?

Choose the hand you want to use for communication. You should use your dominant hand.

Fig. 12: Dominant hand selection page

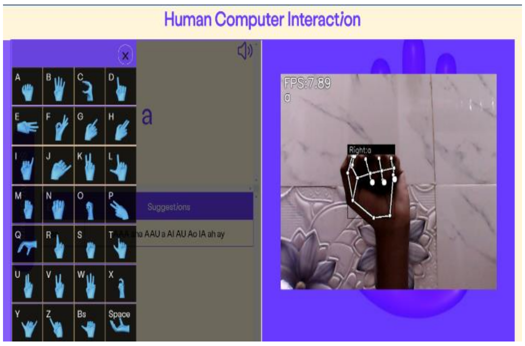

Human Computer Interaction

Fig. 13: Learning section

The following Fig. 13 shows the page to learn the language. The user can learn the alphabet of American sign language using the present images provided to help them. The predicted word will be displayed on the side of the face cam window.

Figure 14 shows a page to create a room or join a room. This generates a link that can be used by other users to join the meet socket IO is being used.

Flask applications access low-latency bi-directional communications between the clients and the server.

Figure 15 represents the chat room where the user can chat using American sign language. A room is an arbitrary channel that sockets can join and leave. It can be used to broadcast events to a subset of clients.

Discussion-Comparative Study

The dataset underwent K-fold cross-validation with ten folds and the average accuracy of different methods was calculated and tabulated. The results indicated that SVM outperformed other techniques such as ANN and MLP, as well as machine learning algorithms like KNN and random forest in terms of accuracy. Table1 shows the average accuracy for different algorithms.

In order to ensure comprehensive testing, every dataset undergoes pre-processing to extract features through the MediaPipe framework and is then trained using multi-layer perceptron to accurately classify gestures which shows in Tables 2-3.

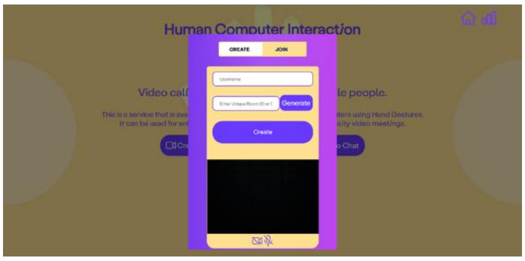
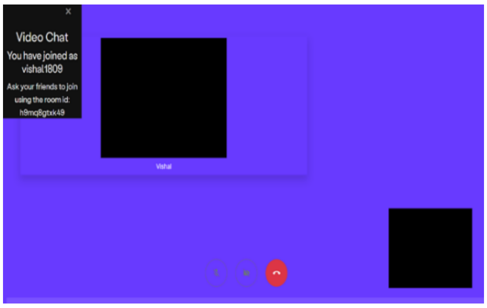

Human Computer Interaction
CREATE
JOIN
Username
Enter Unique Room ID or C
Generate
Create
Video cal
le people.
This is a service that is ava
It can be used for ent
ters using Hand Gestures.
ality video meetings.
Cr
o Chat

Fig. 14: Chat section



You have joined as vishal1809

Ask your friends to join using the room id: h9mq8gtxk49

Fig. 15: Video chat

Table 1: Summary of the architecture used for the MLP model

Layer type	Number of layers	Input shape	Output shape	Activation function
Input	1	(42,)	-42	None
Dropout	1	-42	-42	None
Dense	1	-42	-20	ReLU
Dropout	1	-20	(20,)	None
Dense	2	(20,)	(10,)	ReLU
Dense	1	(10,)	(28,)	Softmax

Table 2: Machine learning model average accuracy

Algorithm	Dataset 1 (ASL)	Dataset 2 (ASL)
SVM	88.64	83.97
KNN	70.41	20.63
Random forest	55.74	25.74
ANN	91.67	79.87
MLP	95.65	92.07

Table 3: Performance analysis

Dataset	Training accuracy	Testing accuracy	Pre	Re	F1
1	97.83	95.65	95.65	95.65	95.65
2	95.54	92.07	92.07	92.07	92.07

Conclusion

This study focuses on the potential of hand gesture-based interactions as a viable alternative to touch-screen devices. By focusing on American Sign Language (ASL) recognition, the study explains a hand gesture recognition system that leverages standard webcams for data collection and neural networks for classification. This breakthrough holds promise for real-time ASL translation on widely available personal devices, paving the way for greater inclusivity and communication accessibility for the deaf and mute community. This study marks a significant step toward breaking down communication barriers and fostering a more interconnected society. Ultimately, this research highlights the transformative power of technology in enhancing human-machine interaction for individuals with diverse communication needs.

Acknowledgment

We acknowledge Dr. Sanjay Bokade, principal of Rajiv Gandhi Institute of Technology, Mumbai, and Dr. Hari Vasudevan, principal, of D J Sanghavi College of engineering for their valuable support which motivates us to research in this field.

Funding Information

We have not received any funds from any organization.

Author’s Contributions

Sharmila Rathod: Conceived and designed the analysis; collected the data; contributed data or analysis tools; performed the analysis**,** exploration, and selection of the primary hand gesture dataset, the model training, ensuring efficient detection of palm and hand landmarks for speech-impaired people.

Nilesh Rathod: Played a pivotal role in dataset research, ensuring the selection of a diverse and representative hand gesture sign dataset. Led the comprehensive testing phase, validating the model's accuracy, precision, recall and F1-score.

Nilesh Marathe: Contributed insights into dataset requirements, influencing the decision-making process for effective model training. Fine-tuned trained parameters and optimized the model.

Aruna Gawade: Contributed to the technical aspects, she has made a substantial contribution to the concept or design of the article; and the acquisition, analysis, or interpretation of data for the article.

Jyoti Kundale: Contributed to the technical aspects and drafted the article or revised it critically for important intellectual content.

Nikita Kulkarni: Revised and approved the version to be published.

All authors reviewed the results and approved the final version of the manuscript.

Ethics

We have followed research ethics like scientific integrity, human rights and dignity, and collaboration between science and society. These principles ensure that participation in studies.

References

Ahuja, M. K., & Singh, A. (2015). Hand Gesture Recognition Using PCA. *International Journal of Computer Science Engineering and Technology (IJCSET)*, *5*(7), 267–271. http://www.ijcset.net/docs/Volumes/volume5issue7/ijcset2015050714.pdf

Ahmed, A. A., & Aly, S. (2014). Appearance-based Arabic Sign Language recognition using Hidden Markov Models. *2014 International Conference on Engineering and Technology (ICET)*, 1–6. https://doi.org/10.1109/icengtechnol.2014.7016804

Baranwal, N., & Nandi, G. C. (2017). An efficient gesture based humanoid learning using wavelet descriptor and MFCC techniques. *International Journal of Machine Learning and Cybernetics*, *8*(4), 1369–1388. https://doi.org/10.1007/s13042-016-0512-4

Elakkiya, R., Selvamani, K., Rao, R. V., & Kannan, A. (2012). Fuzzy Hand Gesture Recognition Based Human Computer Interface Intelligent System. *UACEE International Journal of Artificial Intelligence and Neural Netowrks*, *2*(1), 29–33. https://doi.org/10.3850/978-981-07-1403-1_741

Gaikwad, S., Patel, S., & Shetty, A. (2021). Brain tumor detection: An application based on machine learning. *2021 2nd International Conference for Emerging Technology (INCET)*, 1–4. https://doi.org/10.1109/INCET51464.2021.9456347

Gaikwad, S., Shetty, A., Satam, A., Rathod, M., & Shah, P. (2019). Recognition of American Sign Language using Image Processing and Machine Learning. *International Journal of Computer Science and Mobile Computing*, *8*(3), 352–357.

Halder, A., & Tayade, A. (2021). Real-time Vernacular Sign Language Recognition using MediaPipe and Machine Learning. *International Journal of Research Publication and Reviews*, *2*(5), 9–17. https://doi.org/10.13140/RG.2.2.32364.03203

Kalidolda, N., & Sandygulova, A. (2018). Towards Interpreting Robotic System for Fingerspelling Recognition in Real Time. *Companion of the 2018 ACM/IEEE International Conference on Human-Robot Interaction*, 141–142. https://doi.org/10.1145/3173386.3177085

Kurdyumov, R., Ho, P., & Ng, J. (2011). Sign language classification using webcam images. *Computer. Therm. Sci*, *10*, 9029.

Liu, W., Fan, Y., Li, Z., & Zhang, Z. (2015). RGBD Video Based Human Hand Trajectory Tracking and Gesture Recognition System. *Mathematical Problems in Engineering*, *2015*(1), 863732. https://doi.org/10.1155/2015/863732

Ma, J., Gao, W., Wu, J., & Wang, C. (2000). A continuous Chinese sign language recognition system. *Proceedings Fourth IEEE International Conference on Automatic Face and Gesture Recognition*, 428–433. https://doi.org/10.1109/afgr.2000.840670

Murakami, K., & Taguchi, H. (1991). Gesture recognition using recurrent neural networks. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 237–242. https://doi.org/10.1145/108844.108900

Pavlovic, V. I., Sharma, R., & Huang, T. S. (1997). Visual interpretation of hand gestures for human-computer interaction: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(7), 677–695. https://doi.org/10.1109/34.598226

Pandey, P., & Jain, V. (2015). Hand Gesture Recognition for Sign Language Recognition: A Review. *International Journal of Science Engineering and Technology Research*, *4*(3), 464–470.

Rajesh, K., Rathod, S., Kundale, J., Rathod, N., Anand, M. C. J., Saikia, U., Tiwari, M., & Martin, N. (2023). *A Study on Interval Valued Temporal Neutrosophic Fuzzy Sets*. *23*(1), 341–349. https://doi.org/10.54216/IJNS.230129

Rathod, N., & Wankhade, S. (2022a). Optimizing neural network based on cuckoo search and invasive weed optimization using extreme learning machine approach. *Neuroscience Informatics*, *2*(3), 100075. https://doi.org/10.1016/j.neuri.2022.100075

Rathod, N., & Wankhade, S. (2022b). Investigation of optimized ELM using Invasive Weed-optimization and Cuckoo-Search optimization. *Nonlinear Engineering, 11*(1), 568–581. https://doi.org/10.1515/nleng-2022-0257

Rekha, J., Bhattacharya, J., & Majumder, S. (2011). Hand gesture recognition for sign language: A new hybrid approach. *Proceedings of the International Conference on Image Processing, Computer Vision and Pattern Recognition (IPCV)*, 1–7. http://worldcomp-proceedings.com/proc/p2011/IPC4065.pdf

Sharma, U., Tiwari, O., Sankhe, R., & Yadav, S. M. (2024). Customizable Sign Language Gesture Prediction for Assistive Devices Using Machine Learning. *International Journal for Research in Applied Science and Engineering Technology*, *12*(4), 4041–4047. https://doi.org/10.22214/ijraset.2024.60791

Thakur, A., Budhathoki, P., Upreti, S., Shrestha, S., & Shakya, S. (2020). Real Time Sign Language Recognition and Speech Generation. *Journal of Innovative Image Processing*, *2*(2), 65–76. https://doi.org/10.36548/jiip.2020.2.001

Tharwat, A., Gaber, T., Hassanien, A. E., Shahin, M. K., & Refaat, B. (2015). SIFT-Based Arabic Sign Language Recognition System. *Afro-European Conference for Industrial Advancement: Proceedings of the First International Afro-European Conference for Industrial Advancement AECIA 2014*, 359–370. https://doi.org/10.1007/978-3-319-13572-4_30

Vogler, C., & Metaxas, D. (2004). Handshapes and Movements: Multiple-Channel American Sign Language Recognition. *Gesture-Based Communication in Human-Computer Interaction: 5th International Gesture Workshop, GW 2003*, 247–258. https://doi.org/10.1007/978-3-540-24598-8_23

Wang, R. Y., & Popović, J. (2009). Real-time hand-tracking with a color glove. *ACM Transactions on Graphics*, *28*(3), 1–8. https://doi.org/10.1145/1531326.1531369