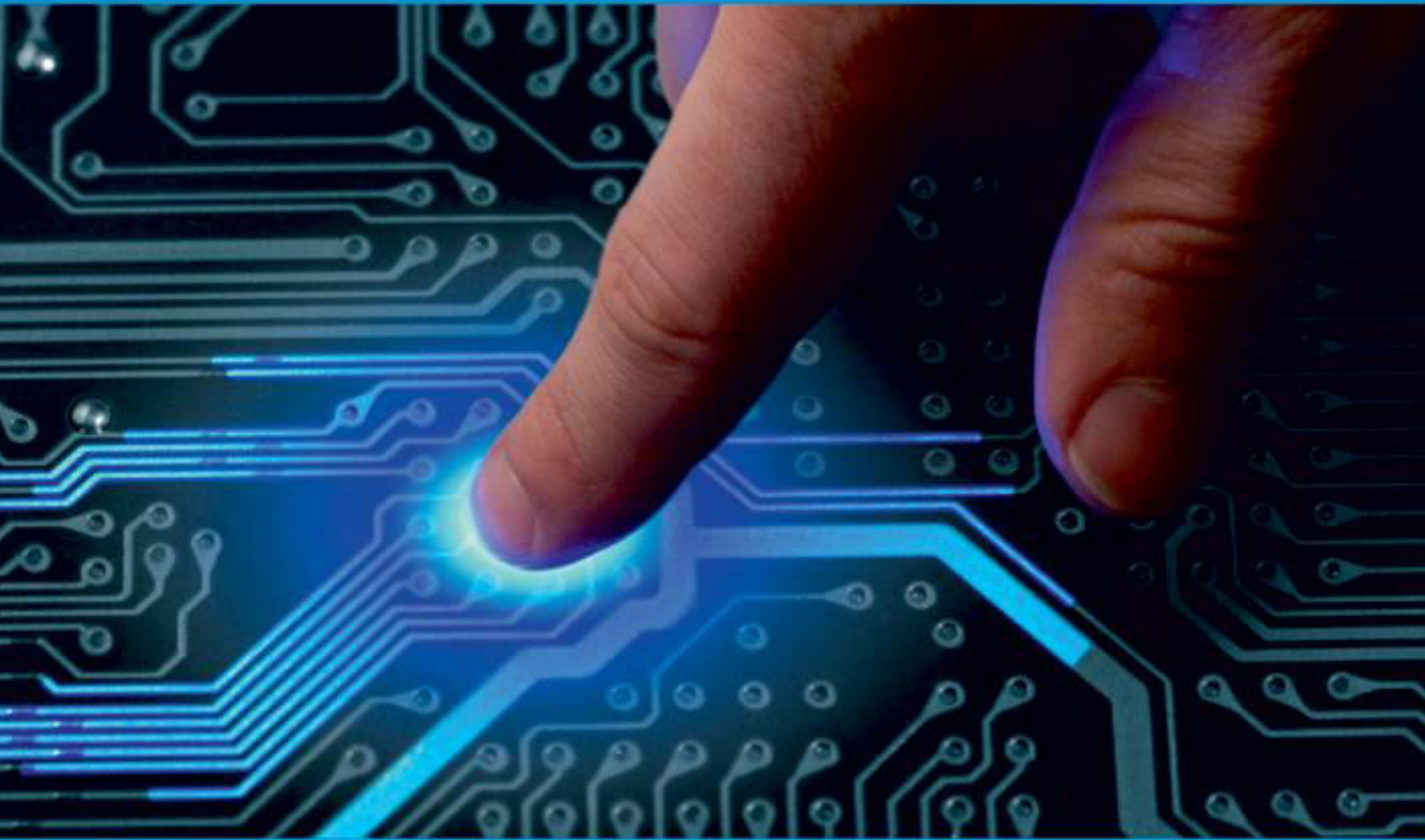
IJIRCCE

e-ISSN: 2320-9801 | p-ISSN: 2320-9798



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

Volume 10, Issue 6, June 2022

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.165

International Journal of Innovative Research in Computer and Communication Engineering

| e-ISSN: 2320-9801, p-ISSN: 2320-9798| www.ijircce.com | |Impact Factor: 8.165 |

|| Volume 10, Issue 6, June 2022 ||

| DOI: 10.15680/IJIRCCE.2022.1006225|

Sign Language Translator for Speech- and Hearing- Impaired

Aradhana M K, Inchara T, Ishita Soni, Indu K S

UG Student, Dept. of I.S.E., The Oxford College of Engineering, Bangalore, Karnataka, India

UG Student, Dept. of I.S.E., The Oxford College of Engineering, Bangalore, Karnataka, India

UG Student, Dept. of I.S.E., The Oxford College of Engineering, Bangalore, Karnataka, India

Assistant Professor, Dept. of I.S.E., The Oxford College of Engineering, Bangalore, Karnataka, India

ABSTRACT: Speech and hearing impairment is a disability which affects an individual's ability to communicate like a normal person. People who are affected by this use other mediums of communication such as sign languages. Although sign language is ubiquitous in recent times, there remains a challenge for non-sign language speakers or non-signers to communicate with sign language speakers or signers. To reduce the communication gap between signers and non-signers, the following methodology is adopted. A software-based application is implemented to allow sign language to text/speech conversion and vice-versa. For the sign language to text/speech conversion, video sequence is captured in real time using built-in webcam, which is then pre-processed and fed into Convolutional Neural Network (CNN) model, that is trained using dataset containing signs of American Sign Language (ASL). The model then outputs the predicted sign in textual and audio formats. For text/speech to sign language conversion, textual or audio input is given using keyboard or built-in microphone. Then the equivalent sign images from the dataset is acquired and displayed in series as output. The accuracy of the system is improved with the help Deep Learning technique, CNN and system can be adaptable for real-time use and to be integrated to build mobile/web-based applications. The similar model was also developed using ML algorithms such as MLP, SVM, Naïve Bayes and k-NN. But the highest accuracy was achieved using CNN compared to others.

KEYWORDS: American Sign Language; Convolutional Neural Network; Deep Learning; Machine Learning; Sign Recognition and Translation; Speech input and output

I. INTRODUCTION

Sign Language (or signed language) is the common form of communication used by people with hearing and/or speech impairment. Sign languages are languages that use visual-manual modality, generally referred to as gestures, to convey messages. A sign includes movement of one or both hands and change of hand shape, accompanied with facial expressions that correspond to a specific meaning. Although the deaf and dumb or signers can communicate using sign languages without any problem amongst themselves, there are still serious challenges faced by the deaf and dumb community in their day-to-day lives, especially trying to integrate into educational, social and work environments, when compared to non-signers.

Generally, for the communication between signers and non-signers, trained sign language interpreters are needed, whose demand have been increasing rapidly over the past five years. Other means such as video-based remote human interpreters using high-speed Internet connections have been introduced. Though these provide an easy-to-use sign language interpreting service, they also include many major limitations such as interpreters are expensive to afford, not very reliable, inefficient, may not be present at all times in case of sudden requirements or emergencies and so on. Thus, our system is designed to entirely overcome these limitations.

The overall goal of this project is to develop a new vision-based technology for recognizing and translating continuous signs to text/speech, and vice-versa. Hence, the project is named as Sign Language Translator for Speech- and Hearing- Impaired, and is developed using Deep Learning approach. Specifically a dataset containing numerous signs, each with numerous images for training and testing the models is created. Based on validating k-Nearest Neighbor, Support Vector Machine, Multi-layer perceptron, Naïve Bayes and Convolutional Neural Network algorithm- based models, CNN was chosen as an ideal and best fit for implementation because of its highest accuracy compared to others. These trained models were also tested in real-time for which graphical interface is integrated to provide better experience and easy understanding. Furthermore, the trained model can also be easily integrated to build mobile and web- based applications for real world use. Thus, the objective to meet the day-to-day communications requirements of disabled people with normal people and vice-versa can be achieved with a system such as this.

II. Related work

Literature review of our proposed system shows that there have been many explorations done to tackle the sign language recognition in videos and images using several methods and algorithms. Also, the past work is mostly categorized into hardware-based and visual cues or software-based approaches.

Helene Brashear, etal. [1] proposed a hardware-based approach to build a constrained, lab-based sign language recognition system with the goal of making it a mobile assistive technology[9]. Multiple sensors for disambiguation of noisy data were used to improve recognition accuracy. Though user with this system could have the benefit to monitor the camera's view via the head-mounted display, it inherited the disadvantages in terms of cost, usability and maintenance. Also, while statistically enough accuracy was achieved, dataset used to train model was very small making it not much effective.

The work of Geetha M, etal. [2] aimed at recognizing 3D dynamic signs corresponding to ISL(Indian Sign Language) words using Microsoft Kinect Camera and has proposed a novel method for feature extraction of dynamic gestures. This method integrated both local and global information of dynamic signs. A new trajectory based feature extraction method using the concept of Axis of Least Inertia (ALI) and Eigen distance based method using seven 3D key points were introduced for global and local feature extraction respectively. Integrating both features improved the performance of the system and its accuracy, but the methods are considerably less user-friendly and more expensive.

Yellapu Madhuri, etal. [4] presented a mobile vision- based sign language translation device for automatic translation of ISL into speech in English to assist hearing- and/or speech- impaired people to be able to communicate with hearing people. The proposed system could recognize finger spelling in real-time using a single camera to track the user's hands. The approach was broken down into mainly three stages, beginning with the image acquisition followed by image processing to extract the features for recognition and finally identifying the signs and providing audio output, which was played on some audio device. The system worked with fairly high precision and accuracy, but the proposed method focused did not much focus on facial expressions and did not support 2-way communication model.

A L C Barczak, etal. [3] proposed a method to produce exemplars that could be fed into Machine Learning algorithms to establish some recognizable patterns for all possible classes. This was the major contribution, where as minor contribution was given to use specific form of feature extraction method called moment invariants, for which the computation methods and values were furnished with the dataset. Since major focus was on producing exemplars, method could not be used as recognition system. Also, the images were taken at certain angel, perpendicular to the system, which limited the number of samples and the trained model's efficiency.

To overcome use of expensive hardware-based approaches, Matheesha Fernando etal. [5] proposed a low cost approach for real-time sign language recognition. The paper suggested possible ways to deal with sign language postures to identify the signs and convert them into text and speech using appearance-based approach with a low-cost web camera. From a series of image processing techniques used, the Hu-Moment Classification [11] was identified as the best approach since it provided enough accuracy without a controlled background and with less light adjustments. But the approach only focused on hand postures, making it ineffective for hand gestures.

There are many different types of sign languages from around the world. There is no single sign language that can be called as universal. Also, they have no or very little relevance to spoken languages. Interestingly, most countries that share the same spoken language do not necessarily have the same sign language as each other. There are somewhere between 138 and 300 different types of sign languages used around the globe today. Some of them include ASL(American Sign Language), ISL (Indian Sign Language), BSL (British Sign Language), Auslan (Australian Sign Language), DGS (German Sign Language), LSF (French Sign Language) and so on. Similarly, the research also focused on not one single sign language but also many other.

Sandrine Tornay, etal. [6] proposed a sign language recognition system which was modeled by hand shape information, obtained from pooling resources from multiple sign languages. They developed a multilingual sign language approach, where hand movement modeling was done with target sign language independently by deriving hand movement subunits. The approach was then validated by investigating it on Swiss German Sign Language, DGS and Turkish Sign Language. Though the approach demonstrated that sign language recognition systems can be effectively developed by using multilingual sign language resources, it still does not address resource constraint issues such as developing a system with reduced number of examples and signers, and has space limitation problems.

Roberto Nurena-Jara, etal. [7] proposed a system for PSL(Peruvian Sign Language) recognition. The method proposed constructed a dataset consisting of 3D spatial positions of static gestures from PSL alphabets, using HTC

Vive device and a well-known technique to extract 21 key points from the hand(s) to obtain a feature vector. Then to validate the appropriateness of this dataset, a comparison of four baseline classifiers were used, which were able to provide highest accuracy in terms of F1 score. But the system still possess some drawbacks such as it does not support dynamic gestures and also have some misclassifications due to some alphabets in PSL having same sign shapes.

At present, most of the research on sign language recognition systems are still based on traditional machine learning non-end-to-end systems, which also requires a lot of manual design work and has poor generalization abilities. Mengyi Xie, etal. [8] proposed a sign language recognition system using the Residual Neural Network[13], to implement end-to-end recognition of ASL. The proposed approach provided high accuracy on static gestures, though it wasn't tested on dynamic signs.

From this detailed literature survey, it is observed that Sign Language Translator that can provide two-way communication between disabled and normal people is still not explored. This is thus, the important inclusion in out implementation, along with use of Deep Learning concept that can provide better results and performance compared to others and use of dataset containing ASL, which is considered to be one of the oldest and most common sign languages used by signers.

III. Proposed Methodology

Sign language is the daily language or natural way of communication for people with speech and hearing disability, and is also the main tool for special education schools to teach and convey ideas. It is a visual language that uses hand gestures, change of hand shape etc to track the information in order to express meaning. But non-signers find it extremely difficult to understand this way of communication. Hence, trained sign language interpreters are needed mainly during medical and legal appointments, educational and training sessions, etc. In order to provide a solution to the above problem, a Sign Language Translator is proposed as it meets the requirement to bridge the gap of communication between signers and non-signers. The proposed model has two modes, namely, sign language to text and speech conversion, and text and speech to sign language conversion. Following figure 1 represents these two modes and the procedure involved.

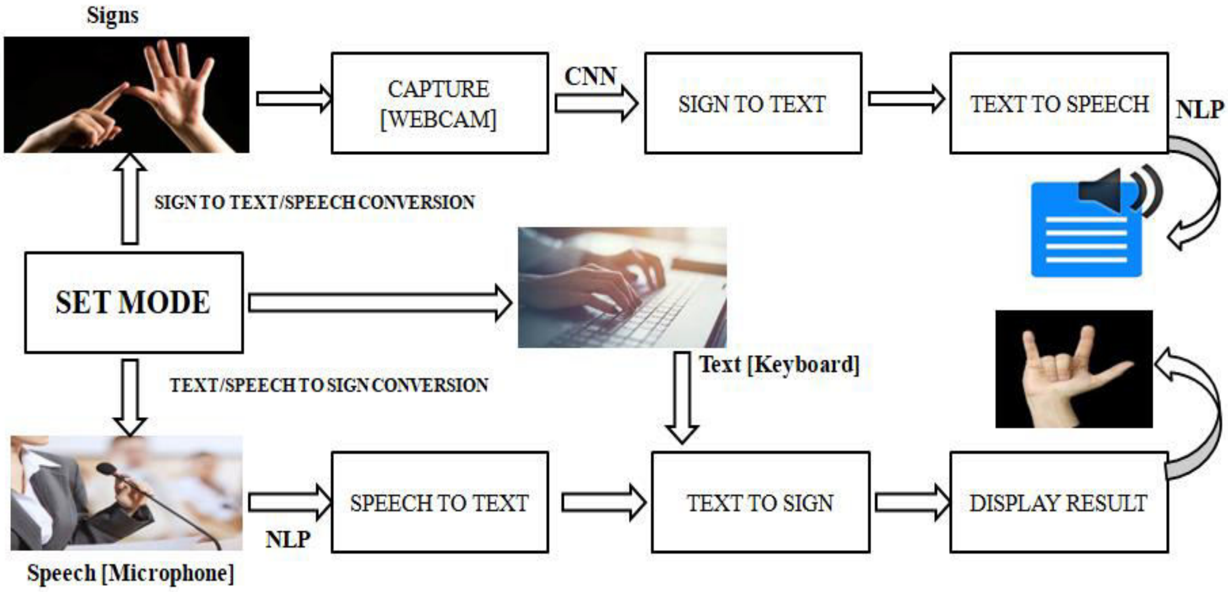


Fig.1. Represents sign to text/speech and text/speech to sign language conversion modes and its working

A. *Sign language to text and speech conversion:*

This is the first mode of the model, which allows hand gestures or signs given as input to be converted into its equivalent text and audio outputs. The hand gestures are captured in the video form using built-in webcam. After processing the captured video, number of frames as images are obtained, which are then given as input to the ML/DL algorithms. The model was initially developed using Deep Learning algorithm[15], specifically Convolutional Neural

Network algorithm, which is very popular for image extraction and classification. The process includes the following steps –

Step 1: Image pre-processing:

A live video of a signer making gestures for the camera serves as the system's input for translating between sign languages. Each frame of each video is analysed, with each image/frame serving as the input to the Convolutional Neural Network at the current time step. The pre-processing step involves converting RGB image frames into grayscale images and then normalizing them. Finally, the preprocessed and rescaled images are given as input to trained model for extraction and classification. Figure 2 represents the use of image preprocessing in our system's implementation.

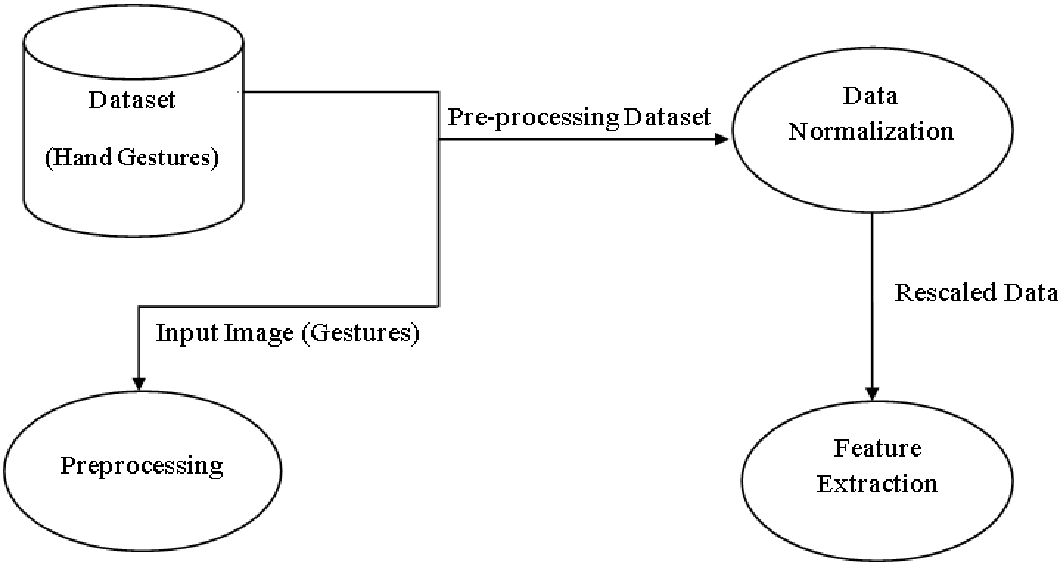


Fig.2. Image preprocessing representation

Step 2: Feature extraction:

The selection of good features is crucial for gesture recognition because hand gestures are rich in shape variation, motion and textures. Although hand postures can be recognized by extracting some geometric features such as fingertips, finger directions, and hand contours. Moreover, a number of other non-geometric features are available, such as colour, silhouette, and textures, which are inadequate for recognition and hence, can be ignored. Various operations are performed within CNN algorithm for better temporal feature extraction [10]. The feature matrix obtained will consider information about only the hand signs in the image after discarding background and other noisy data.

Step 3: Model validation:

The ability of model to process dataset can help in identifying anomalies and test correlations while searching for patterns across the data feed. Here we split our feature extracted dataset into training and testing dataset. Where one trains a model, the other confirms it works (or doesn't work) correctly with previously unseen data. Once the model is trained, it is validated using the test dataset. Best model is selected and used further.

Along with CNN model for classification, other ML-based models were designed and tested to finally choose the best model. For this purpose, ML algorithms[13] chosen are k-Nearest Neighbor (KNN)[17], Multilayer Perceptron[18], Support Vector Machine (SVM)[19] and Naïve Bayes[20]. One of the most basic Machine Learning algorithms, kNN is mostly used for categorization. It classifies data points based on the similarity of previously stored data points. The 'k' in the kNN denotes how many nearest neighbors were employed to categorise the fresh data points. SVMs can be used for both regression and classification tasks. However, it is often used in classification. Support Vector Machines are highly preferred because they produce considerable accuracy with less computational power. The goal of the support vector machine algorithm is to find a hyperplane in N-dimensional space (N-number of features) that uniquely classifies data points. MLP or Multi Layer Perceptron is a fully connected class of feed-forward artificial neural network that generates a collection of outputs from a collections of inputs. It has number of variations which can be used for image classification but it has a lot of deficiency than CNN. Finally, Naïve Bayes is a kind of classifier that works based on Bayes' theorem, where the theorem works with conditional probabilities. Conditional probabilities indicate the probability of an event using prior knowledge. The class with the highest probability is considered the best class and based on this classification results are obtained. From implementing all these algorithms, finally CNN[14] [16] was chosen as the best because it resulted in better accuracy and performance compared to others.

Step 4: Classification using CNN:

A Convolutional Neural Network (CNN) is a form of artificial neural network that is specifically made to process pixel input and is used in image recognition and processing. A neural network is a system of hardware and/or software patterned after the operation of neurons in the human brain. Traditional neural networks are not ideal for image processing and requires images to be fed in reduced-resolution pieces. CNN have their “neurons” arranged more like those of the frontal lobe, the area responsible for processing visual stimuli in humans and other animals. The layers of neurons are arranged in such a way as to cover the entire visual field avoiding the piecemeal image processing problem of traditional neural networks. Also, CNN uses a system much like a multilayer perceptron that has been designed for reduced processing requirements. The layers of a CNN consist of an input layer, an output layer and a hidden layer that includes multiple convolutional layers, pooling layers, fully connected layers and normalization layers. The removal of limitations and increase in efficiency for image processing results in a system that is far more effective, simpler to trains limited for image processing and natural language processing.

CNN model consists of number of layers, each performing some specific task ranging from feature extraction to actual prediction. An image Input Layer is where we initialize the size of input image. This size represents height, width, and the number of channels. In this case, input data is a grayscale image, hence the number of channels is 1.
The convolutional layer is the second layer in the CNN network and used to perform convolution operation on input images. In a CNN, the input is a tensor with a shape - (number of inputs) x (input height) x (input width) x (input channels). After passing through a convolutional layer, the image becomes abstracted to a feature map, also called an activation map, with shape - (number of inputs) x (feature map height) x (feature map width) x (feature map channels). The convolutional layer is the core building block of a CNN. Also, convolution layer is sometimes called feature extractor layer because features of the image are get extracted within this layer. A convolution operation involves simple application of a filter/kernel to an input image resulting in an activation. Repeated application of the same filter to an input will result in a map of activations or simply, feature map, indicating the locations and strength of a detected feature in an input image. Thus, the objective of this operation is to extract the high-level features such as edges, from the input image.

The convolution layer is always present in combination with another layer that performs an activation. Here, ReLU layer is present as ReLU activation is chosen. ReLU is the abbreviation of rectified linear unit, which applies the non-saturating activation function . It effectively removes negative values from an activation map by setting them to zero. Convolution layer contains ReLU activation to make all negative value to zero, thus performed together. The main reason why ReLU is used is because it is simple, fast, and empirically it seems to work well, and also because early papers observed that training a deep network with ReLU tended to converge much more quickly and reliably than training a deep network with sigmoid or tanh activation functions.

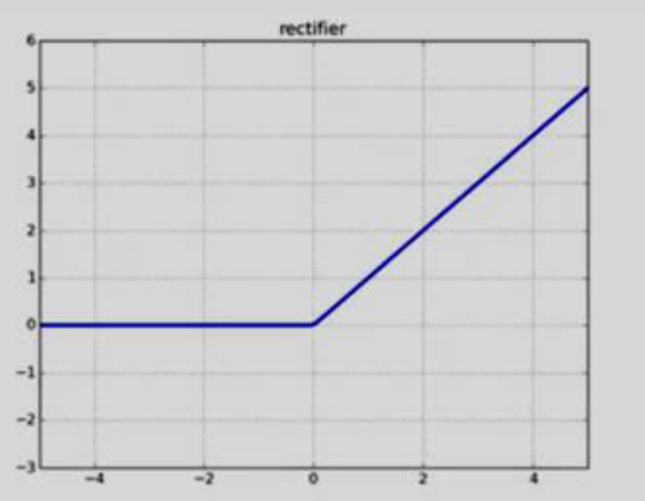


Fig.3. Representation of ReLU activation function.

The pooling layer partitions the input image into a set of rectangles and, for each such sub-region, outputs the maximum or an average. The pooling layer serves to progressively reduce the spatial size of the representation, to reduce the number of parameters, memory footprint and amount of computation in the network. It is common to periodically insert a pooling layer between successive convolutional layers (each one typically followed by an activation function, i.e., a ReLU layer) in a CNN architecture. Furthermore, it is useful for extracting dominant features. From the available two types of pooling layers (average and max pooling), max pooling is chosen, since it returns the maximum value from the portion of the image under the filter/kernel and also because of increased efficiency and performance compared to average pooling

There can be several convolutional and max pooling layers in the model depending on the requirement and chosen architecture. In total, these layers were used for three timed to develop CNN model for this project. After these convolutional and max pooling layers, the final classification is done via fully connected layers, which is simply as feed-forward neural netowrk. Neurons in a fully connected layer have connections to all activations in the previous layer. The input to the fully connected layer is the output from the final max pooling layer, which is flattened and then fed into the fully connected layer. Softmax layer is the last layer of CNN model within this fully-connected network of layers, residing in the end. Softmax is for multi-classification and uses Softmax activation function, which is used to get probabilities of the input being in a particular class for classification. The Output layer is the layer in a neural network model that directly outputs a prediction, which is the indication of the class the given hand gesture belongs to. This class will be the text output of the Sign Language Translator and internally converted to provide audio output using Google API, gTTS.

B. *Text and speech to sign language conversion:*

This is the second part of the process. It includes text or audio data given as input via entering text through keyboard or inbuilt microphone. In case of audio input, it is internally converted into its equivalent textual data and then final output is displayed. The textual data represents the class to which sign or hand gestures belong to. Once the class is identified, final output consisting of best image or series of images are displayed.

IV. Simulation Results

The proposed model is evaluated based on ASL dataset. We have created a dataset comprising 1000 samples (sign language words) each to train the ML/DL models. The dataset is split in the ratio of 75:25 for training and testing respectively. Once the neural network was trained using train dataset, it was tested on the test dataset. After this, performance measures like accuracy, precision, recall, and f1 score was measured via confusion matrix, which is used to measure performance of classification algorithms. The Machine Learning models, namely, Naïve Bayes, SVM, kNN and MLP displayed same accuracy values approximately, which is 33.3% on provided dataset. This showed that the performance of these models were the almost same to one another. Figure 4 below shows the result of recall metric of each of these algorithms. As seen, they all have the same measure without much variation from one another.

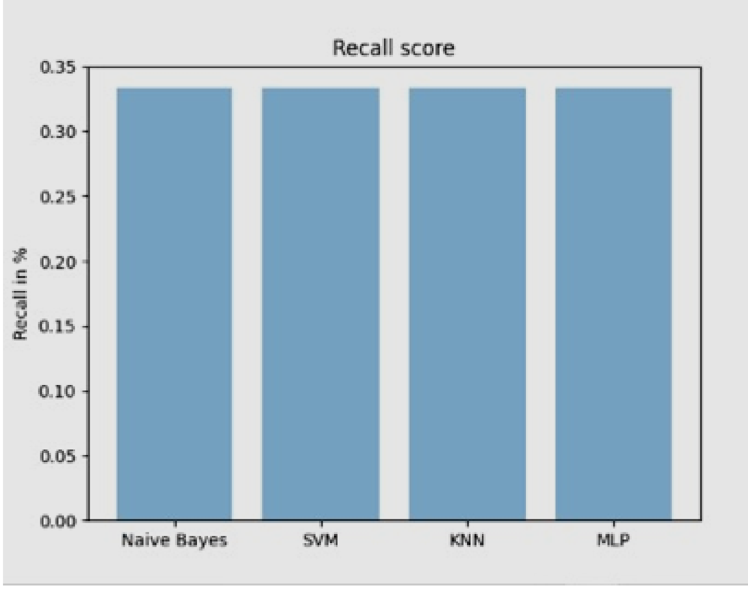


Fig.4. Comparing Recall metric values of Naïve Bayes, SVM, kNN and MLP algorithms.

Later similar procedure of training and testing using same dataset using CNN was performed. It was trained on 1000 epoch value. The trained CNN model, on testing displayed highest accuracy of approximately 98%. This shows that CNN performed lot better compared to previously tested ML algorithms. Thus, CNN was chosen for the final implementation of the Sign Language Translator system. Following table shows the approximate accuracy achieved using each model –

Table.1. Approximate accuracy achieved by each algorithms after testing phase.

Method	Accuracy
CNN	98%
K-Nearest Neighbor	33.3%
MLP	33.3%
Naïve Bayes	33.3%
SVM	33.3%

Further, implemented model performed really well on real-time testing as well. Graphical interface added to the implementation made it easier for the usage purpose. It visibly provides the choice of both modes, out of which one can be chosen using simple clicks. Once a specific mode is chosen, the new interface to perform chosen operation appears, providing multiple window-like configurations. Following figures 5 and 6 demonstrates the outputs of sign language to text conversion mode and text to sign language conversion mode.

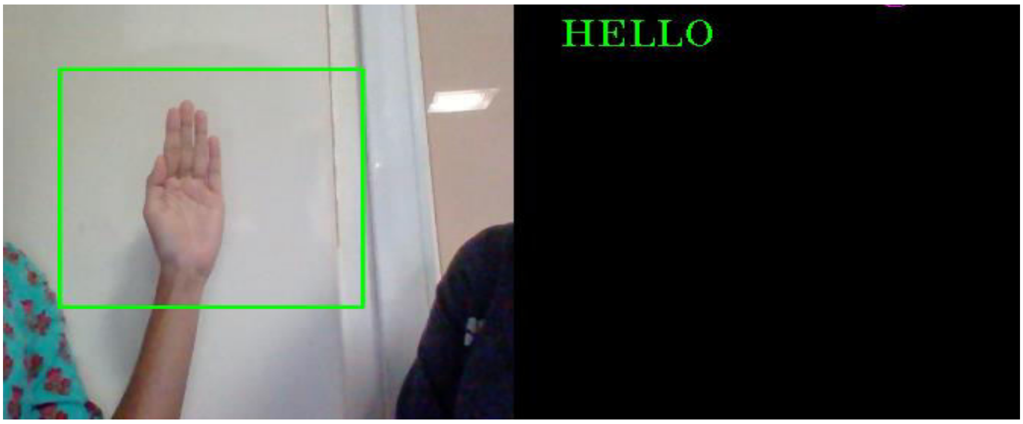
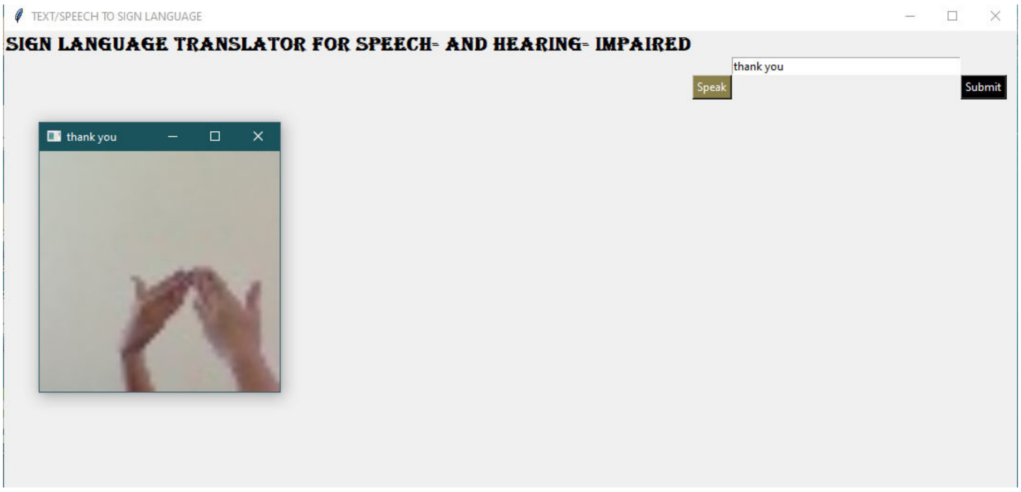


Fig.5. Demonstration of sign language to text conversion mode – hand gestures are made at the left side and textual output indicating the description of the sign is provided at the right side.


TEXT/SPEECH TO SIGN LANGUAGE

SIGN LANGUAGE TRANSLATOR FOR SPEECH- AND HEARING- IMPAIRED

thank you

Fig.6. Demonstration of text to sign language conversion mode – given input text is shown at the left side and output of sign or hand gesture is displayed at the right side.

V. Conclusion and Future Work

In this paper, we have proposed a system to translate American Sign Language into English Text and speech and vice versa, based solely on Deep Learning and Machine Learning concept. The dataset is created using numerous classes, each class containing numerous images of signs or hand gestures for training the models and for better performance. The main objective of this implementation is to reduce the gap of communication between signers and non-signers and helping hearing and speech impaired people in real time to freely communicate with normal people and vice-versa. This in turn also allows disabled people to feel and be equal to normal people in the society.

Although several other ML methods such as MLP, kNN, SVM and Naïve Bayes are tested and validated, the Deep Learning based Convolutional Neural Network is concluded best for developing Sign Language Translator. It is also because the layers of a CNN model have multiple convolutional filters working and scanning the complete feature matrix and carry out the dimensionality reduction continuously. This enables CNN to be a very appropriate choice for image classification and processing. Furthermore, the implemented models can be incorporated to build mobile applications and web-based application for real-time use and easy access.

Although this paper has achieved high accuracy, the data set is not large enough and does not include all the sign language words. Also, the data belongs to signs of American Sign Language only. Hence for the future work, it is suggested to find ways to make an inclusion of different types of Sign Languages used across the world and also to provide textual and audio inputs/outputs in regional languages as well.

References

1. Helene Brashear, etal. “Using Multiple Sensors For Mobile Sign Language Recognition”, Seventh IEEE International Symposium on Wearable Computers, 2003.
2. Geetha M, etal . “A Vision Based Dynamic Gesture Recognition Of Indian Sign Language On Kinect Based Depth Images” , International Conference on Emerging Trends in Communication, Control, Signal Processing and Computing Applications (C2SPCA), 2013.
3. A L C Barczak, etal. “A New 2D Static Hand Gesture Color Image Dataset for ASL Gestures”[massey.ac].
4. Yellapu Madhuri, etal. “Vision-Based Sign Language Translation Device”, International Conference on Information Communication and Embedded Systems (ICICES), 2013.
5. Matheesha Fernando etal. “Low cost approach for Real Time Sign Language Recognition”, IEEE 8th International Conference on Industrial and Information Systems, 2013.
6. Sandrine Tornay, etal. “Towards Multilingual Sign Language Recognition”, IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2020.
7. Roberto Nurena-Jara, etal. “Data Collection Of 3D Spatial Features Of Gestures From Static Peruvian Sign Language Alphabet For Sign Language Recognition”, IEEE Engineering International Research Conference (EIRCON), 2020.
8. Mengyi Xie, etal. “End-to-End Residual Neural Network With Data Augmentation For Sign Language Recognition”, IEEE 4th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), 2019.
9. Lilit Hakobyan, etal. “Mobile Assistive Technologies For The Visually Impaired”, Survey Of Ophthalmology, 2013.
10. Isabelle Guyon, etal. “An Introduction To Feature Extraction”, Studies in Fuzzies and Soft Computing book series (STUDFUZZ, volume 207), 2006.
11. Mohamed Rhouma, etal. “Improving The Performance Of Hu Moments For Shape Recognition”, International Journal of Applied Environmental Sciences, 2014.
12. Kaiming He, etal. “Deep Residual Learning For Image Recognition”, Microsoft Research, 2019.
13. Batta Mahesh, “Machine Learning Algorithms – A Review”, International Journal of Science And Research (IJSR), 2019.
14. Laith Alzubaidi, etal. “Review Of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions”, Journal of Big Data, 2021.
15. Ajay Shrestha, etal. “Review Of Deep Learning Algorithms And Architectures”, IEEE Access (Volume: 7), 2019.
16. Hagyeong Lee, etal. “Introduction To Convolutional Neural Network Using Keras: An Understanding From A Statistician”, Communications for Statistical Applications and Methods, 2019.
17. Siddharth Nandakumar Chikalkar, “K-Nearest Neighbors Machine Learning Algorithm”, International Journal of Creative Research Thoughts (IJCRT), 2020.
18. Marius-Constantin Popescu, etal. “Multilayer Perceptron And Neural NEtworks”, WSEAS Transactions on Circuits and Systems, 2009.
19. Mariette Awad, etal. “Support Vector Machines For Classification”, Efficient Learning Machines, 2015.
20. Anshul Goyal, etal. “Performance Comparison Of Naïve Bayes And J48 Classification Algorithms”, IJAER, 2012.

INNO SPACE
SJIF Scientific Journal Impact Factor
Impact Factor: 8.165

doi cross ref

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

निस्केयर
NISCAIR

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN COMPUTER & COMMUNICATION ENGINEERING

9940 572 462 6381 907 438 ijircce@gmail.com

Scan to save the contact details

www.ijircce.com