Hand Gesture Recognition and Text Voice Conversion for Deaf and Dumb

Prajwal M Dixit[1], Prashanth Patel G P[2], Priya G E[3], Shubhashree T S[4], Narendra Kumar S[5]

[1,2,3,4]Student, Dept. of CS&E, JNNCE, Shivamogga, India
[5]Assistant Prof., Dept. of CS&E, JNNCE, Shivamogga, India

Corresponding author: Prajwal M Dixit (e-mail: prajwalmdixit@gmail.com).

"This work is supported in part by the Karnataka State Council for Science and Technology under the 46th series of Student Project Programme."

ABSTRACT People basically communicate with each other in daily life using various languages. Similarly, sign language is being used by blind and deaf people for communication. Sign language was created with an intention to help the deaf communicate with others. Hearing Impairment(HI) is another issue to consider. According to WHO (2018) data, 63,000,000 people are suffering with significant auditory loss. Adult-onset deafness is predicted to affect 7.6% of Indians, whereas childhood-onset deafness affects 2%.

INDEX TERMS Blind, Hearing impairment, Sign language.

I. INTRODUCTION

Sign gesture recognition is one of the most sophisticated areas where computer vision and artificial intelligence have helped improve communication with physically challenged individuals in emergency circumstances, such as accidents, calling for help, and so on. Sign language provides a definite way of communication while preserving their unique grammatical patterns. Hand gesture detection is used for deaf communication as well as to allow gesture-based signalling systems.

Hand gesture recognition is a very complicated field that has been utilised to improve deaf communication as well as to allow gesture-based signalling systems. Various handcrafted feature methods and deep learning-based algorithms have been proposed to handle the gesture recognition challenge. In the subject of sign recognition, deep learning is recently gaining the general implementation of sign language recognition. Images are required for static movements, while a sequence of images or videos is necessary for dynamic motions to extract the spatio-temporal characters. Pre-processing includes choosing several frames from the video, adding various filters if necessary, and resizing the frames according to the model's input.

The further chapter in the paper include, the brief discussion of contents in the different papers referred for the work. The various methods available for image pre-processing, training the model and also the various models that can be used along with their effectiveness, advantages, disadvantages are observed. The methodology followed in the current work is, pre-processing of the input images with different steps being followed, transferring it to the model, applying a classifier and finally obtaining the text output which is further converted to voice output also. The work is implemented by collecting the image data in a dataset, pre-processing the images and training the model. The model used in the work is MobileNetV2, which is a 53 layered CNN model. The model is tested and found that most of the predicted label is same as the actual label. Real time images are given as input to the model, which gives the final result with the text and voice output.

II. RELATED WORK

In the study [1], data acquisition is made using appropriate sensors, cameras, or microphones. The mean filter method is used for noise reduction, image segmentation is done using Threshold method and the image is identified and classified using the CNN and ANN algorithms. The main advantage of this implementation is, it does not require any pricy technology and CNNS can capture very complex details in the hand gestures, as they are expert at extracting hierarchical features from images. But it is difficult to find threshold value and thresholding cannot be applied to a multiple channel image.

The study [2] can be observed with the pretrained VGG-16 and a RNN with LSTM making up the model. Here, the first step is dataset acquisition and next is the pre-processing step. 5 frames are taken at equal interval per video and after extracting the frames, images are resized and evaluated from scale of 0-7. Further, these frames are sent to VGG-16 then to LSTM then to SoftMax classifier. It uses both spatial and

temporal characteristics on a positive note. But, VGG-16 is a very huge network and takes more time to train its parameter. The models can be updated and retrained to recognize new gestures.

The Binarized Neural Network (BNN) boosts the performance by improving the speed of training and reduces the memory size in case of [3]. First the image is converted from RGB to HSV colour space then gaussian blurring is applied to remove noise and for segmentation, Otsu thresholding is used, morphological operations are also applied. Unlike the other studies, here BNN is used instead of CNN, which is faster, more efficient and it reduces computational complexity. But it misclassifies some signs because of similar kind of shapes and limited to small number of classes.

Hand detection is performed by using the gloves in the work [4]. Here also, first the image is converted from RGB to HSV then, thresholding is done to extract hand portion and the histogram of the hand with glove is obtained. The model used is CNN with 3 convolution layers and pooling layer. ReLU activation function is used in convolution layers and SoftMax is used at the output layer. This wok reached constant accuracy at 5th epoch itself. The issue is that, the background must be always light with good lighting conditions.

In the work [5], the designed CNN architecture performs using four different types of image pre-processing steps. This work concluded through few experiments that the solid background gives the best results. Here, CNN model's performance is compared to various transfer learning architectures to get the best classifier performance.

A comprehensive dataset of Sinhala Sign Language gestures is collected in the work [6], using a high-quality video camera or depth sensor device to capture a wide range of gestures. The model hyperparameters are optimized and also, data augmentation is used to improve generalization. Here, machine learning models with low computational complexity and memory requirements are used. So, it is suitable for mobile devices. These mobile devices enable real-time communication between deaf and hearing people, which enables deaf individuals to communicate effectively with hearing population.

In the work [7], it is found that a CNN-based hand gesture recognition allows for instant communication and also CNNs provide high accuracy for hand gesture recognition. Here, the images are acquired during runtime through an integrated webcam. These images are stored in a directory and compared with images stored for specific letters in the database using the SIFT algorithm. The interface of the application provides a button to start, stop, and capture a frame. This SIFT algorithm detects hand movement in any orientation and SIFT image features provide a set of features that are not affected by scaling, rotation, or translation, and are extracted using a 4-stage filtering approach.

Considering these works, it is found that there are various image pre-processing methods and various models available for classification. Observing all these methods, grayscale of images, gaussian blur for noise removal, Otsu thresholding for segmentation and morphological operations are found to be useful for the work.

III. METHODOLOGY

The methodology of the proposed hand gesture recognition system is shown in Fig. 1. The proposed technique of hand gesture recognition scheme follows different phases, namely pre-processing of image, transfer of Pre-processed image to CNN model and text to voice conversion.

It is important to design an architecture which is not only good at learning features but also is scalable to massive datasets. The convolutional layers take advantage of inherent properties of images. They use convolution of image and filters to generate invariant features which are passed on to the next layer.

At first the training of the model is done by sending segmented image and the model gets trained, then image that has to be classified is taken from web camera which is further connected to Raspberry Pi.

Pre-Processing of Images: The image taken from the camera is subject to pre-processing

This pre – processing step involves Grayscale, Gaussian blur, Adaptive Threshold with Otsu threshold followed by Morphological operations.

CNN model: The segmented image is sent to the CNN model for classification. Here the image gets classified based on the trained CNN model, and it outputs a text based on classification of Hand Gesture image. The CNN model that is considered here is MobileNetV2 which is light weight and suitable for low end devices.

Text to Voice Conversion: The classified text is converted to voice using an API

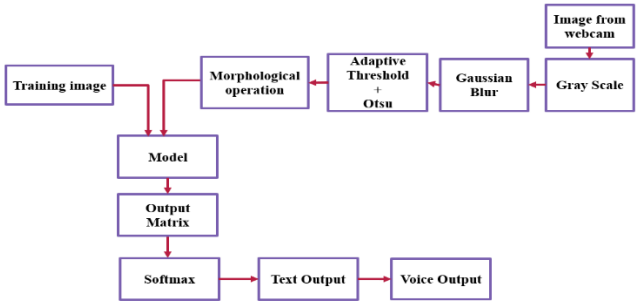


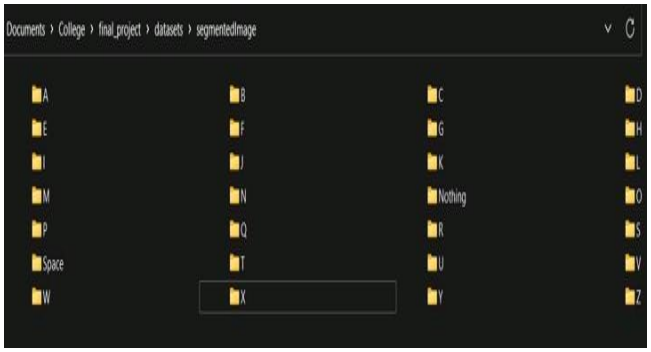
Fig.1 System Architecture

IV. IMPLEMENTATION

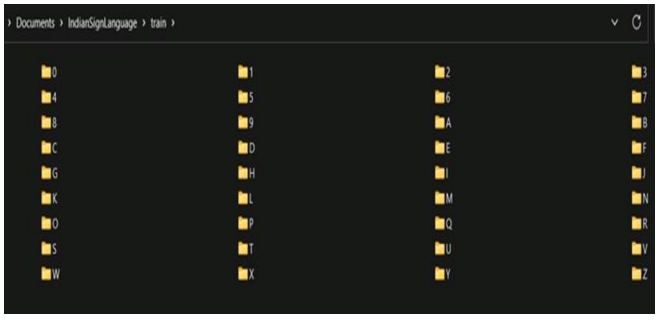
The images from the dataset are considered and these images undergo certain pre-processing steps after which they are provided to the classification model and thus the model gets trained. At the final stage, the model is tested by providing the image input in the real-time and based on the output obtained, the accuracy is calculated.

A. Data collection

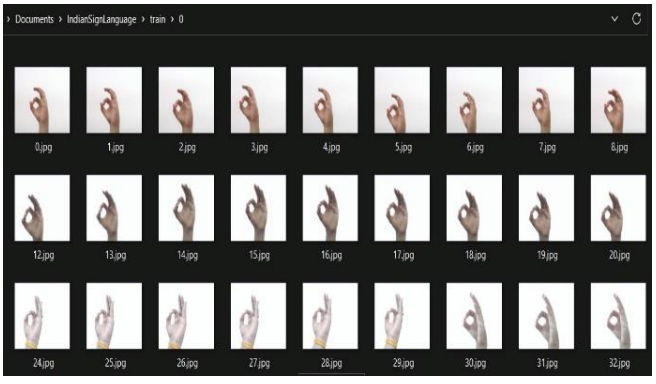
Initially, the image data is collected in a dataset. In the proposed model two datasets from Kaggle namely American dataset by Kapil Londhe, Indian dataset by Satvik pasumarthi is used. Indian Dataset have total 36 classes, in which 26 are alphabets, and 10 are digits. American Dataset have total 28 classes, in which 26 are alphabets, 1 is space and another one is Nothing. Each class of American dataset have 1500 images, and each of Indian dataset have 70 images. They are in various angles and the images in the dataset are converted to the size of 128x128. Fig. 2 shows the datasets considered and images inside Indian dataset class 'O'.



a) American sign language dataset



b) Indian sign language dataset



c) Image inside the class 'O' of Indian dataset

Fig. 2 Dataset snapshots a) American b) Indian c) Images inside a class 'O'

B. Pre-processing

The images are pre-processed at different levels before training the model. Figure 3 depicts the images after each stage of pre – processing. As mentioned before. pre-processing involves Grayscale, Gaussian Blur, Adaptive Threshold with Otsu and Morphological operations.

Grayscale: The image is converted to Grayscale where the image is converted from 3 channelled colour image to single channelled Grayscale image.

Gaussian Blur: The noise in this Gray scaled image is further removed by applying Gaussian blur to it which smoothens the image. The extent of smooth depends upon the radius chosen for blurring. Each pixel will select a new value set to a weighted average of its neighbouring pixels, with more weight given to the near ones than to those further away.

Adaptive Threshold with Otsu method: The image after blurring is further segmented by using Adaptive thresholding, particularly Otsu's method is being used in the project. Otsu's method is an adaptive thresholding way for binarization, a part of image processing. It can detect the input image's optimal threshold value by going through all possible threshold values (from 0 to 255).

Morphological Operations: On the segmented image morphological operations are performed which includes dilation, erosion. If there is any imperfection in the structure of the image, these operations will rectify them. Dilation increases the object size whereas erosion causes the object to lose its size. Erosion basically improves the zero valued pixels in number and significantly decreases the number of pixels with value one.

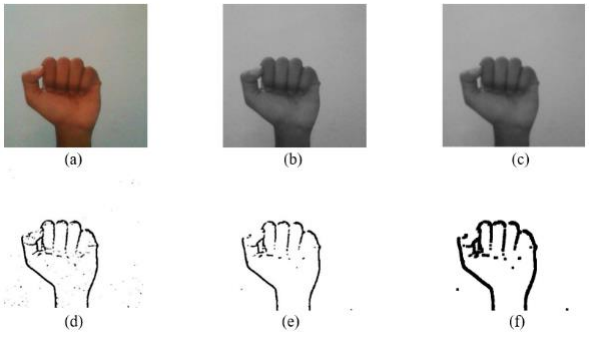


Fig. 3 Pre-processing steps (a) Original image (b) Grayscale (c) Gaussian blur (d) Otsu thresholding (e) Morphological-DILATE (f) Morphological-ERODE

C. Training and classification

The pre-processed images of the dataset are further used to train the model. The pre-processed image is transferred to the MobileNetV2 model for classification purpose. The detailed explanation of MobileNetV2 is given in section V.

V. MobileNetV2

Mobilenetv2 is a convolutional neural network model specially designed for mobile devices which contains 17 bottleneck layers each bottle neck layer includes 3 convolution layers and one convolution layer at first another at last, totally consist of 53 convolutional layers. Bottleneck layer consist of expansion layer where a small number of features are expanded to large extent using large number of 1x1 filter, then convolution with padding is done to preserve the same dimension. Then pointwise convolution where 1x1 filter with small number of filters is done to down the features that pass to next layer, it is also called projection as we go from large feature projecting to small number of features. and also, it contains residual block where features from previous layer fed as input for next layer. Using this bottleneck layers this model delegates computational complexity to training stage. Fig 4 depicts the bottleneck layer of MobileNetV2.

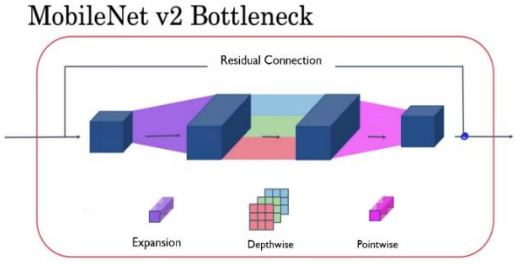


Fig. 4 MobileNetV2 bottleneck layer

VI. RESULTS AND ANALYSIS

The images from the dataset before performing the pre-processing processes will be like as shown in figure 5, from this figure it is evident that the images in the dataset will be initially consisting of colored images with RGB channels present in them.

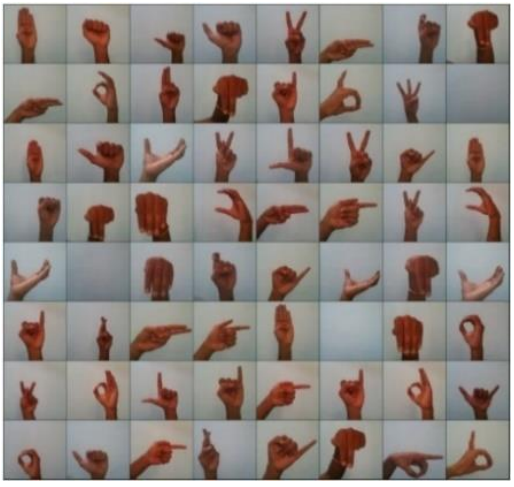


Fig. 5 Before pre-processing

The images will look like as shown in figure 6 after performing all the pre-processing steps, this figure is evident that after performing all the pre-processing steps the image is devoid from noise, segmented and can be used to train the model for effective classification.

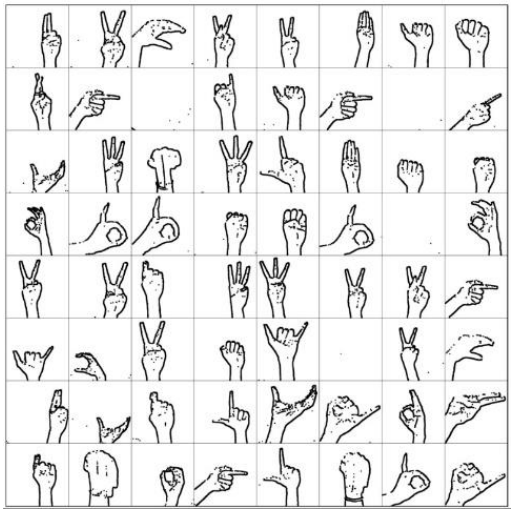


Fig.6 After pre-processing

Figure 7 shows comparison of actual label with predicted label. As it can be observed that the model has same actual and predicted label.

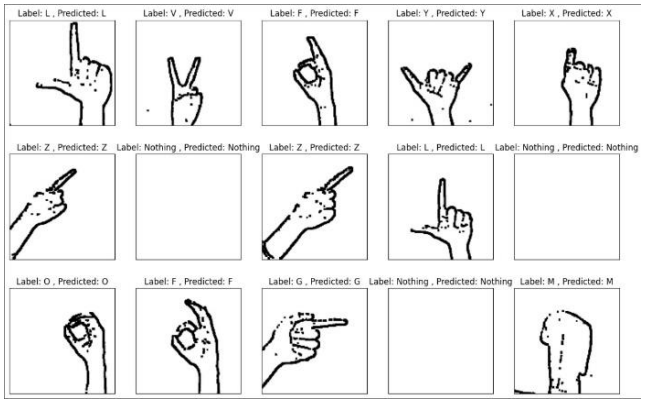
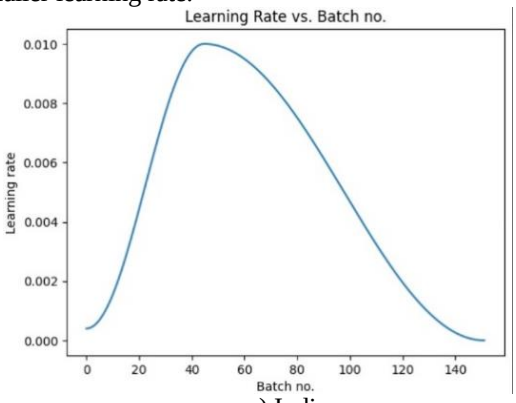


Fig. 7 Result comparing actual label with predicted label

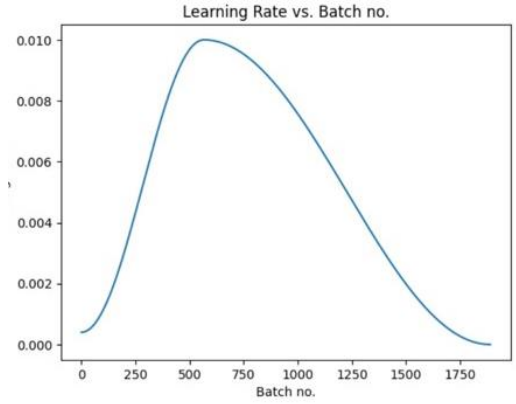
The result analysis is further explained with respect to the hyperparameters which controls the learning process and therefore their values directly impact other parameters of the model which consequently impacts how well the model performs. Hyperparameter tuning consists of finding a set of optimal hyperparameter values for a learning algorithm. Here hyperparameters such as learning rate, accuracy and loss are considered.

The learning rate scheduler which helps to quickly converge model by changing the learning rate dynamically is used here. Fig 8 depicts the graph of the different learning rate that has

been used by model to converge quicky vs batch number. As it can be observed the maximum learning rate is 0.01, the learning rate has gone up to 0.01 then got decreased in both Indian and American model which depicts the proposed model works well with smaller learning rate.



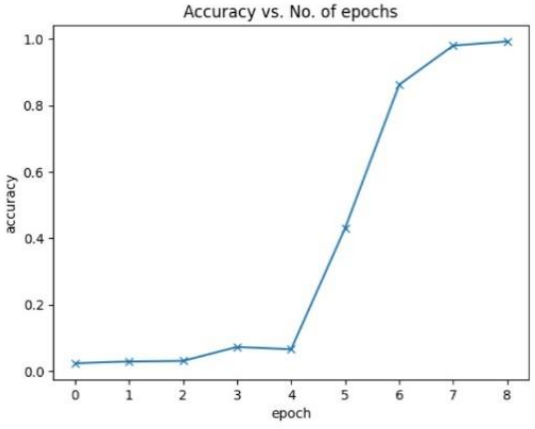
a) Indian



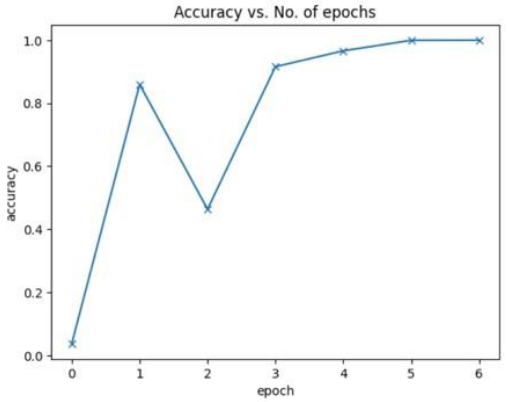
b) American

Fig. 8 Learning curve of learning rate versus batch number for Indian and American dataset

The Figure 9 is the graph of accuracy against number of epochs. As it can be observed, the accuracy of both Indian and American reaches saturation very early with 8 and 6 epochs respectively depicting that model is learning great.



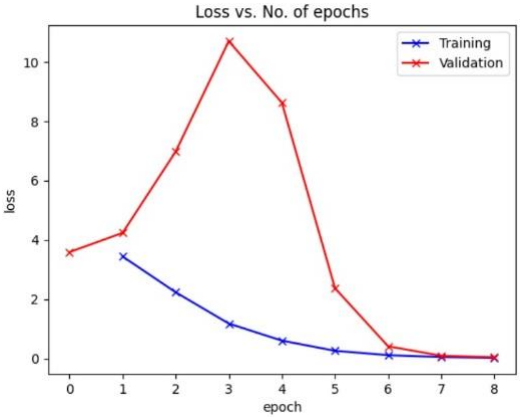
a) Indian



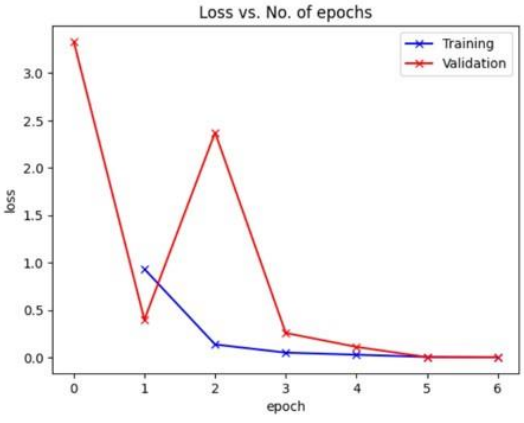
b) American

Fig. 9 Learning curve of Accuracy versus number of epochs for Indian and American dataset

Figure 10 shows graph of loss against number of epochs. The error for both training and validation data is calculated. Here the training curve is smooth because the average loss of batch is considered and the error is getting decreased with number of epochs depicting the same that proposed model is learning great.



a) Indian



b) American

Fig. 10 Learning curve of loss versus epoch for Indian and American dataset

A Standalone application that can recognize hand gestures in real-time and translate them into text and speech is developed to make proposed hand gesture recognition system more accessible. Figure 11 shows a snapshot of the application user interface. This application takes users sign action as an input and detects gesture in real-time. Here image is taken from webcam or laptop camera. This application can recognize both Indian and American sign language. 'Clear' button is given to clear the text inside 'sentence' box. 'Speak' button is used to convert text inside 'sentence' box to speech. '<-' is used to delete the detected letter one by one inside 'sentence' box.

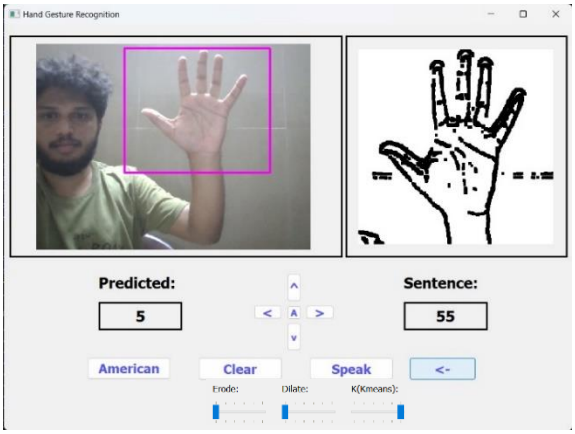


Fig. 11 Standalone Desktop Hand Gesture Recognition Application

The detected images label is displayed in 'Predicted' box. If the last 30 frames that is considered have same label in 28 frames that letter is appended to the Sentence text box. When specific letter is written in 'Sentence' box, the next sign action needs to be performed. The user's camera result is displayed in the left section, whereas the image after preprocessing which is considered for model is displayed on the right. The special feature of the application includes detection of both American and Indian sign language, freedom for user to choose automatic detection and manual detection of hand, control for user to vary the level of application of Erode, Dilate, k value (K-means).

VII. CONCLUSION

Hand gesture recognition and voice-to-text conversion technologies are powerful tools that have the potential to bridge the communication gap between the deaf and mute community and the hearing world. The sole purpose of this project is to bridge the communication gap between deaf, dumb and hearing world. The proposed system improves access to education and engages specially abled people in social interactions. As mentioned in section VI both American and Indian sign language can be translated. As MobileNetV2 is a light weight model specifically designed for mobile devices, it can be deployed in low end devices.

VIII. FUTURE SCOPE

The proposed system can be further improved to efficiently work with any kind of background. Further advancement can be done by including emergency, greeting words or sentences in dataset.

REFERENCES

[1] Surekha P, Niharika Vitta, Pranavi Duggirala, Teja Sree Desani, Venkata Surya Saranya, "Hand Gesture Recognition and Voice, Text Conversion Using CNN and ANN", Second International Conference on Artificial Intelligence and Smart Energy,2022.

[2] Qazi Mohammad Areeb, Mohammad Nadeem, "Deep Learning Based Hand Gesture Recognition for Emergency Situation: A Study on Indian Sign Language", International Conference on Data Analytics for Business Industry (ICDABI),2021.

[3] Mohita Jaiswal, Vaidehi Sharma, Abhishek Sharma, Sandeep Saini, Raghuvir Tomar, "An Efficient Binarized Neural Network for Recognizing Two Hands Indian Sign Language Gestures in Real-time Environment", 17th India Council Conference (INDICON), 2020.

[4] Diksha Hatibaruah, Anjan Kumar Talukdar, Kandrapa Kumar Sarma, "A Static Hand Gesture Based Sign Language Recognition System Using Convolutional Neural Networks", IEEE India Council International Conference (INDICON),2020.

[5] Atharva Dumbre, Shrenik Jangada, Shreysas Gosavi, Jaya Gupta, "Classification of Indian Sign Characters Utilizing Convolution Neural Networks and Transfer Learning Models with Different Image Processing Techniques", IEEE World Conference on Applied Intelligence and Computing (AIC),2022.

[6] V. J. Dhanawansa, T. P. Rajakaruna, "Sinhala Sign Language Interpreter Optimized for Real-Time Implementation on a Mobile Device", 10th International Conference on Information and Automation for Sustainability ICIAFS, 2021.

[7] S. Vanaja, R.Preetha, S.Sudhan, "Hand Gesture Recognition for Deaf and Dumb using CNN Technique", 6th International Conference on Communication and Electronics Systems(ICCES), 2021.

[8] https://youtu.be/a99p_fAr6e4 ,Custom Hand Gesture Recognition with Hand Landmarks Using Google's Mediapipe + OpenCV in Python.

[9] https://www.analyticsvidhya.com/blog/2021/06/building-a-convolutional-neural-network-using-tensorflow-keras/, Building a Convolutional Neural Network Using TensorFlow – Keras.

[10] https://youtu.be/KuXjwB4LzSA, But what is a convolution? , 3 Blue1Brown.