Real-Time Two Way Communication System for Speech and Hearing Impaired Using Computer Vision and Deep Learning

Tanuj Bohra
Information Technology
Dwarkadas J. Sanghvi college of engineering
Mumbai, India
tanuj.bohra97@gmail.com

Shaunak Sompura
Information Technology
Dwarkadas J. Sanghvi college of engineering
Mumbai, India
shaunaksompura@gmail.com

Krish Parekh
Information Technology
Dwarkadas J. Sanghvi college of engineering
Mumbai, India
krishparekh9@gmail.com

Purva Raut
Assistant Professor
Information Technology
Dwarkadas J. Sanghvi college of engineering
Mumbai, India
purvaraut@gmail.com

***Abstract*—Sign Language is the most expressive form of communication for speech and hearing impaired people to communicate with normal person but a normal person cannot understand sign language. So in order to break this barrier of communication there needs to be a system that can enable conversion of sign language to voice or text and voice or text to sign language and do it in real time. The systems that currently exist are not real time, do not facilitate two-way communication, require static surrounding conditions or have low recognition accuracy. There exist systems that have good accuracy but require external hardware like gloves [3] which increases the cost. Our contribution to solving this problem consists of a Sign Language Communication System. It is a real-time communications system built using the advancements in Image Processing, Deep Learning and Computer Vision that provides real-time sign language to text and text to sign language conversion. The project is software-based which can be installed on any computer with good specifications. It is also a two-way communication system allowing not just speech and hearing impaired to communicate with normal people but also other way around. The primary goal of our system is to enable hearing and speech impaired people to communicate with people that are not disabled in real time by interpreting alphabets, numbers and words in the Indian sign language. The system is able to predict 17600 test images in 14 seconds with an average prediction time of 0.000805 seconds with an accuracy of 99%.**

Keywords—convolutional neural network, gesture recognition, text to sign, sign to text

I. INTRODUCTION

In India close to 3 million people are categorized as speech and hearing impaired according to an Indian Government report on disabled person [1]. One of the biggest problems faced by the speech and hearing impaired community is the barrier in communication. To overcome this barrier, they are taught sign language since a very young age. Sign language allows them to communicate with other members of the community as well as the members of the society. The reality of the society is such that other than the people interacting with the deaf and dumb, very few people actually know sign language. The problem that exists here is that a person who is not deaf is highly unlikely to understand Sign Language and finding an Interpreter is not easy as there are only about 250 certified sign language interpreters in India [2]. No efforts are taken by the educational institutes to teach the able populace even a few basic signs. This brings us back to square one with regards to the barrier in communication. Tackling this problem would mean educating the society about the sign language. It involves years of efforts on a massive scale. Another solution would be to use the available existing technology to bridge this communication gap. This would allow people without any prior knowledge of sign language to better understand the disabled community.

A software based approach was needed as the current approaches of using hardware like hand gloves [3] and Kinect camera[4] added extra cost. Out of the existing systems, using CNN architecture provided with the most robust sign recognition system. It was important that we have a large enough database to train our system. For this purpose we created our own databases which consisted of various signs and its variations. These variations enabled the system to be trained on a variety of different forms for the same sign, thus making the system more accurate. This paper utilizes Image Processing techniques and Deep Learning based Convolutional Neural Networks to build an ISL recognition system which has reduced cost and increased efficiency.

978-1-7281-2119-2/19/$31.00 ©2019 IEEE

II. Data Research

A. Data Set Selection:

We have prepared our own dataset for this study. The creation of the dataset was done using OpenCV. For each gesture, we have captured 1200 images which were 50X50 pixels. For the initial purpose of testing, we have added 26 alphabets from a to z, 10 numbers from 0 to 9 and 4 words specifically ‘All the best’, ‘remember,’ ‘like’ and ‘you’. The dataset had 40 gestures total at the end. This same data-set will be used to convert text to sign as well as a sign to text.

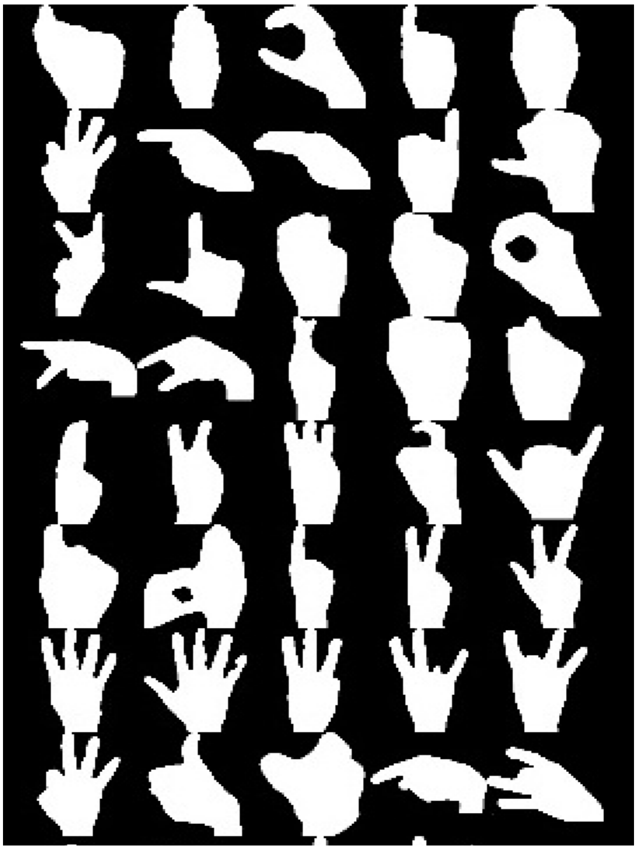
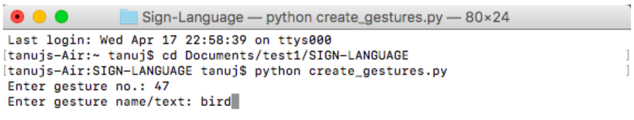


Figure 1: A Snapshot of signs that were inserted in the database

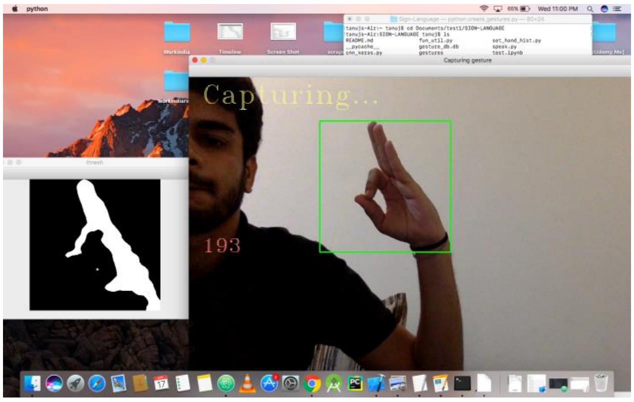
B. Data Pre-processing:

The captured images in 50X50 pixels were converted to grayscale and all the images were flipped. This led to the generation of 2400 images for each gesture. The total numbers of images were 96000. Each of these images was linked to a word and ‘gesture_db.db’ file was created mapping each of the gesture to a string. This database file was used to convert sign to text. In Text to sign conversion mapping of each string gesture to a string was done. New signs and gestures can be easily added using a script that we have written that takes 2400 grey-scale images of the gestures and then allows you to enter the text associated with that gesture.



```
Last login: Wed Apr 17 22:58:39 on ttys000
tanujs-Air:~ tanuj$ cd Documents/test1/SIGN-LANGUAGE
tanujs-Air:SIGN-LANGUAGE tanuj$ python create_gestures.py
Enter gesture no.: 47
Enter gesture name/text: bird
```

Figure 2: Adding a gesture in the sign database



```
tanujs-Air:~ tanuj$ cd Documents/test1/SIGN-LANGUAGE
tanujs-Air:SIGN-LANGUAGE tanuj$ ls
README.md           fun_util.py         set_hand_hist.py
__pycache__         gesture_db.db       speak.py
cnn_keras.py        gestures            test.ipynb
```

Figure 3: Adding a gesture in the sign database (capturing)

III. Literature Survey

A. Existing Work

1. Literature Related to Existing Systems

Existing Systems focuses on translating American Sign Language (ASL) finger spelled alphabet (26 letters). They utilise transfer learning to extract features, followed by a custom classification block to classify letters. This model is then implemented in a real-time system with OpenCV - reading frames from a web camera and classifying them frame-by-frame. This repository contains the code & weights for classifying the American Sign Language (ASL) alphabet in real-time.

1.1 Hand Glove-based System:

Existing methodologies based on hardware most commonly made use of hand glove based method. This technique employs sensors attached to the glove. Look-up table software is usually provided with the glove to be used for hand gesture recognition. The sensors are placed on every finger of the hand. Five sensors are thus used to detect the signing gesture. The drawback of this method is that it has higher hardware costs due to sensors on the hand gloves.[3]

1.2 Microsoft Kinect based system:

One of the existing methodologies that give significant result is implemented with the help of a framework named OpenNI and Kinect Camera. Depth maps representing the discrete range measurements of the environment are generated by using the structured light technique in the Kinect camera [4]. OpenNI framework is used for further processing of this measurement to identify the gesture. The gestures that are performed by user are treated as a sequence of frames which is processed by the system and most accurate representation of sign is given as output

B. Literature related to algorithms used in existing work

Many approaches have been taken previously to solve this complex problem. The approaches range from using specialized hardware fitted with sensors to track hand movements to using fully software based implementation using Image Processing and Convolutional Neural Networks. Every approach has its own shortcomings and it is best fitted to select an approach that best fits the given use case. This process of detection of sign can broadly be accomplished using two methods which is either using just image processing and computer vision or by using machine learning algorithms to classify the image of the sign.

1. Sign Detection using Image Processing

Two of the most widely used methods for sign detection using image processing are discussed in the Table(1):

Parameters	**Skin Colour Detection Algorithm**	**Deep Learning for Hand Detection**
Background conditions	Need fixed background, preferably single coloured.	Can be used in varied background.
Lighting conditions	Needs fixed and good lighting.	Can be used in varied lighting.
Movement	Static camera and hand	Movement allowed
Accuracy	83.3%[5]	94.75%[6]

Table 1: Comparison between Skin Colour Detection Algorithm and Deep Learning for Hand Detection

2. Sign Detection using Machine Learning

Three of the most widely used Machine Learning algorithms for sign detection using image processing are discussed in the Table(2):

SVM	**CNN**	**RNN**
Accuracy obtained 97.5% in case of very small dataset [7]	82.5% accuracy on the alphabet gestures, and 97% validation set accuracy on digits[8]	Accuracy obtained 66% with stacked RBM(Restricted Boltzmann Machine)[9]
Accuracy reduces in case of large dataset in case of SVM	Accuracy increases with increase in dataset	Accuracy increases with increase in dataset
SVM is widely used in case of binary classification	CNN is widely used and considered the best in case of Image classification	RNN is widely in sequential data and where context is needed

Table 2: Comparison between SVM, CNN and RNN algorithm

IV. IMPLEMENTATION

A. Sign to text:

1. Image processing part

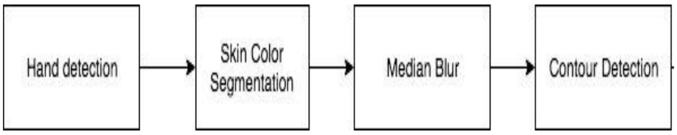


Figure 4: Flow chart of image processing algorithms applied

Hand detection is done using defining a region of interest on the screen in which the user must place its hand. We need to define the range color of the skin in 'hsv' format so that skin color segmentation can be performed on the same. We have written a script that extracts that sets the hand histogram for us.

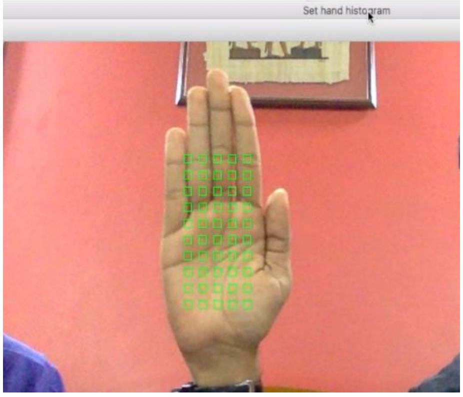


Figure 5: Deciding the range of skin colour of the user

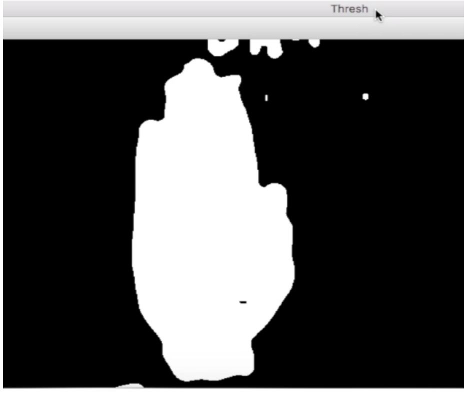


Figure 6: Checking result after applying median blur and contour detection (fingers close to each other)

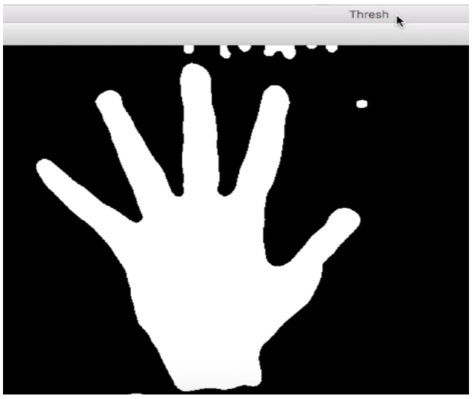


Figure 7: Checking result after applying median blur and contour detection (fingers far from each other)

In this process of getting the hand histogram, we extrapolate the hand to fill dark spots within and blur the image using median blur. This reduces the noise in the image up to a great extent. The next step in this process is contour detection. We find the contours in the image using the 'findContours' function of OpenCV. Then find the contour with the maximum area which is the hand.

2. Machine Learning Algorithm

we have used Convolutional Neural Network to classify gestures as it is arguably the best for image classification and for the size of our database.

2. Text to sign:

The text to sign conversion was done using python. The sentence or word to be converted to sign was taken as input. This was then tokenized and the unnecessary conjunctions were removed. The words that were left were then looked for in the database and if they are present an output image of the same in respective order was shown.

V. ARCHITECTURE:

Our general architecture was a fairly common CNN architecture, consisting of multiple convolutional and dense layers. We have used keras to implement our deep learning based CNN model. There are total of 9 layers in our model The activation function used is 'relu'. In order to avoid over-fitting, we have set the dropout to 0.2. We have divided the images into 70:30 for training and testing. All the layers and their respective parameters are specified in figure 10

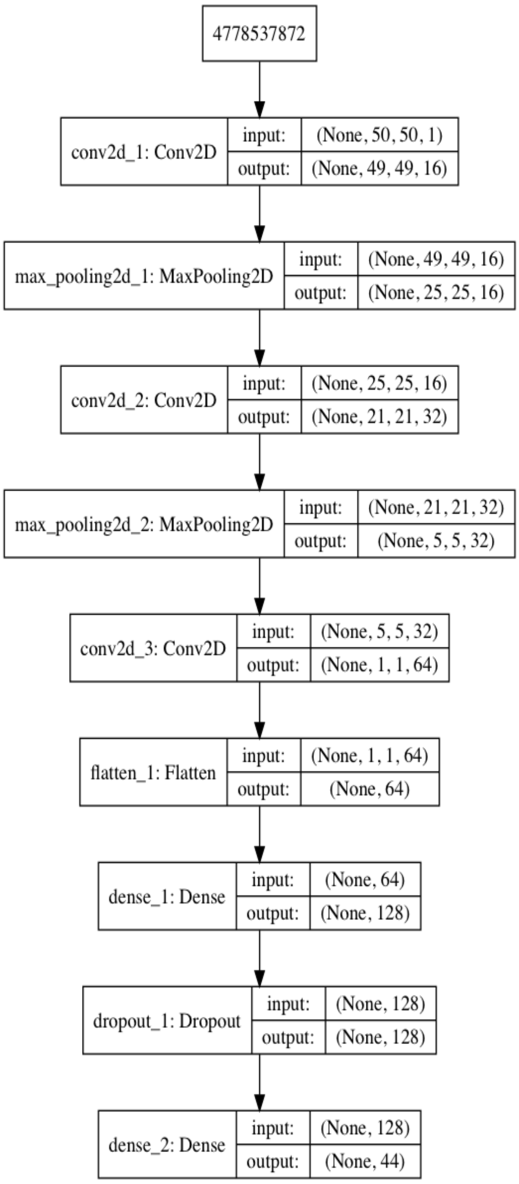


Figure 8: Layers of CNN model

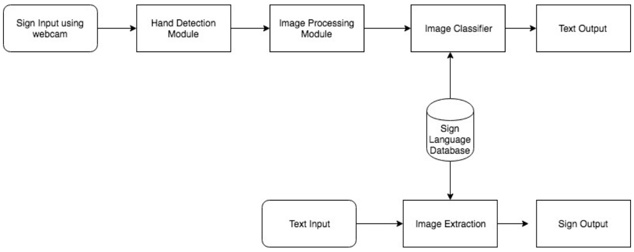
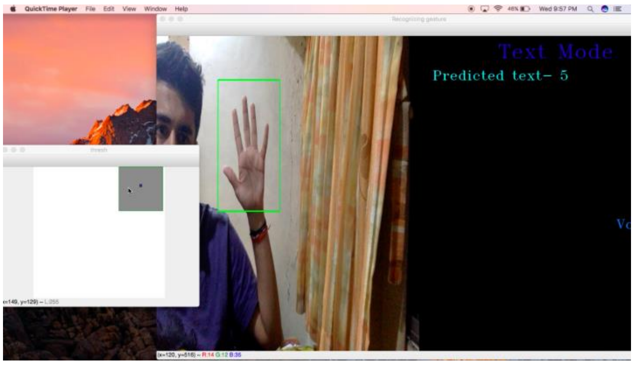


Figure 9: Architecture of our system

VI. RESULTS

The combination of Computer Vision, Image Processing Algorithms and Machine Learning Algorithms enabled us to achieve significant results and improvement over existing systems. Using 2 dimensional images of 50X50 pixel enabled us reduce processing time to a great extent. Our CNN model was able predict 17600 test images in 14 seconds with an average prediction time of 0.000805 seconds with an accuracy of 99%

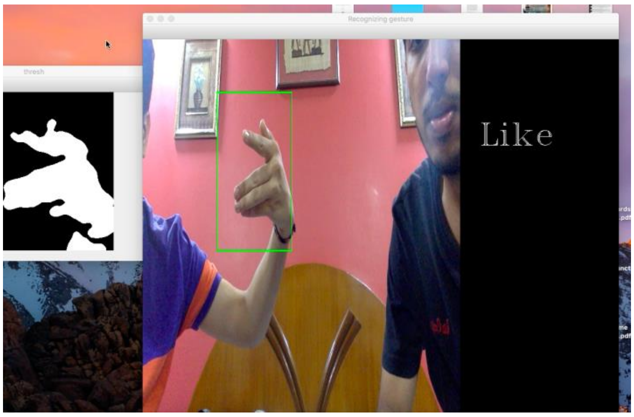
On real-time testing, the results were fairly quick and the output for classification after every 15 frames was shown without any delay. We were able to classify all the alphabets from a to z without significant error or delay. Although, there were some delay and misclassification that occurred in classifying the letter n due to its similarity to letter m. We were also able to detect words accurately without any delay.



Text Mode

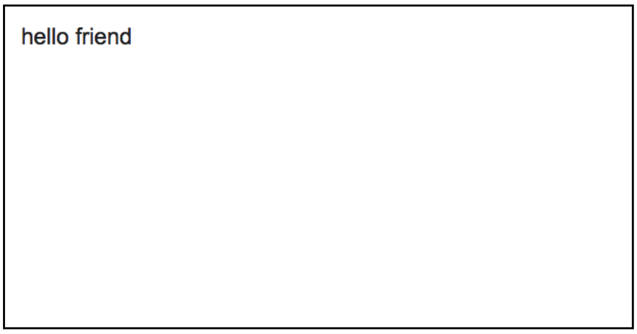
Predicted text- 5

Figure 10: Sign converted to text(number)



Like

Figure 11: Sign converted to text(word)



hello friend

Figure 12: User entering text to convert to sign language

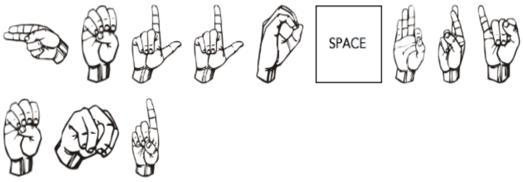


Figure 13: User Entered text converted to sign language

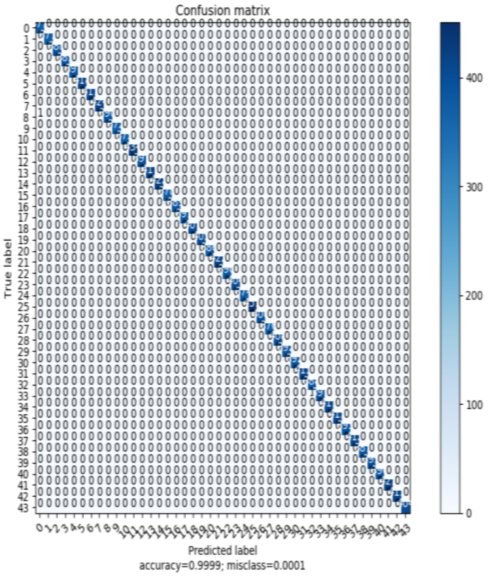
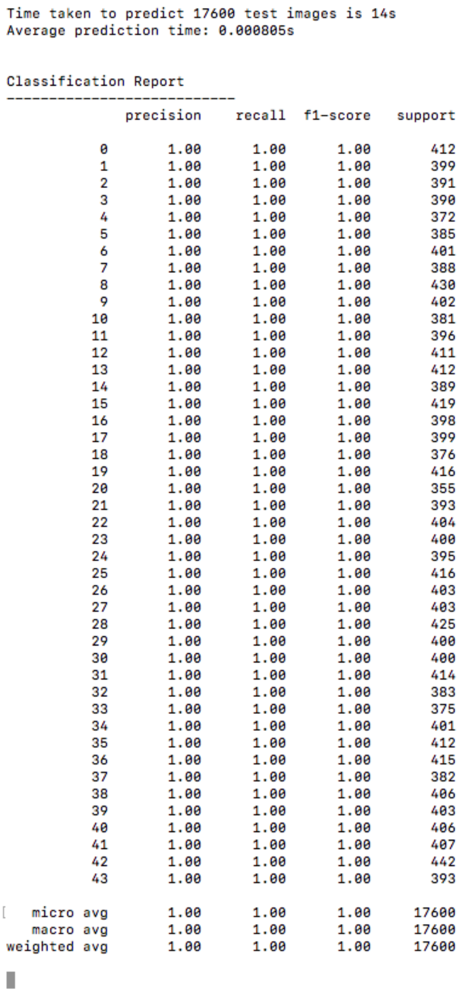


Figure 14: Confusion Matrix showing accuracy and misclassification



```
Time taken to predict 17600 test images is 14s
Average prediction time: 0.000805s


Classification Report
---------------------------
              precision    recall  f1-score   support

           0       1.00      1.00      1.00       412
           1       1.00      1.00      1.00       399
           2       1.00      1.00      1.00       391
           3       1.00      1.00      1.00       390
           4       1.00      1.00      1.00       372
           5       1.00      1.00      1.00       385
           6       1.00      1.00      1.00       401
           7       1.00      1.00      1.00       388
           8       1.00      1.00      1.00       430
           9       1.00      1.00      1.00       402
          10       1.00      1.00      1.00       381
          11       1.00      1.00      1.00       396
          12       1.00      1.00      1.00       411
          13       1.00      1.00      1.00       412
          14       1.00      1.00      1.00       389
          15       1.00      1.00      1.00       419
          16       1.00      1.00      1.00       398
          17       1.00      1.00      1.00       399
          18       1.00      1.00      1.00       376
          19       1.00      1.00      1.00       416
          20       1.00      1.00      1.00       355
          21       1.00      1.00      1.00       393
          22       1.00      1.00      1.00       404
          23       1.00      1.00      1.00       400
          24       1.00      1.00      1.00       395
          25       1.00      1.00      1.00       416
          26       1.00      1.00      1.00       403
          27       1.00      1.00      1.00       403
          28       1.00      1.00      1.00       425
          29       1.00      1.00      1.00       400
          30       1.00      1.00      1.00       400
          31       1.00      1.00      1.00       414
          32       1.00      1.00      1.00       383
          33       1.00      1.00      1.00       375
          34       1.00      1.00      1.00       401
          35       1.00      1.00      1.00       412
          36       1.00      1.00      1.00       415
          37       1.00      1.00      1.00       382
          38       1.00      1.00      1.00       406
          39       1.00      1.00      1.00       403
          40       1.00      1.00      1.00       406
          41       1.00      1.00      1.00       407
          42       1.00      1.00      1.00       442
          43       1.00      1.00      1.00       393

[  micro avg       1.00      1.00      1.00     17600
   macro avg       1.00      1.00      1.00     17600
weighted avg       1.00      1.00      1.00     17600
```

Figure 15: Classification Report showing weighted averages and time taken to predict

VII. Conclusion And Future Scope

Using image processing techniques and CNN model, a reliable recognition of sign language gestures can be carried out. The model proves it is robust enough in different lighting conditions and backgrounds as effective image processing techniques are applied before it is fed into the model. The multiple image processing stages ensures that the system functions well irrespective of the environmental factors. The ISL system is able to recognize all the alphabets and basic words such as – 'remember', 'like', 'best of luck'.

This system can be used in the following ways with certain advancements in the future:

1. Expanded database which includes new signs and use cases commonly encountered in daily life
2. A mobile application which includes the same features and speed with which the desktop application operates.
3. Integration with smart home devices like Alexa and google home.
4. An API can be formed which can be easily integrated with new softwares.
5. The software can be made open source allowing a community to be formed and new advances can be made by various developers throughout the world

VIII. References

[1] http://censusindia.gov.in/Census_And_You/disabled_population.aspx

[2] https://www.pri.org/stories/2017-01-04/deaf-community-millions-hearing-india-only-just-beginning-sign.

[3] N. Praveen, N. Karanth and M. S. Megha, "Sign language interpreter using a smart glove," *2014 International Conference on Advances in Electronics Computers and Communications*, Bangalore, 2014, pp. 1-5.

[4] H. V. Verma, E. Aggarwal and S. Chandra, "Gesture recognition using kinect for sign language translation," *2013 IEEE Second International Conference on Image Information Processing (ICIIP-2013)*, Shimla, 2013, pp. 96-100

[5] Aznaveh M.M., Mirzaei H., Roshan E., Saraee M.H. (2009) A New and Improved Skin Detection Method Using Mixed Color Space. In: Hippe Z.S., Kulikowski J.L. (eds) Human-Computer Systems Interaction. Advances in Intelligent and Soft Computing, vol 60. Springer, Berlin, Heidelberg

[6] Dadashzadeh, Amirhossein & Targhi, Alireza & Tahmasbi, Maryam. (2018). HGR-Net: A Fusion Network for Hand Gesture Segmentation and Recognition. 10.1049/iet-cvi.2018.5796.

[7] Raheja, J.L., Mishra, A. & Chaudhary, A. Pattern Recognit. Image Anal. (2016) 26: 434. https://doi.org/10.1134/S1054661816020164

[8] Bheda, Vivek & Radpour, Dianna. (2017). Using Deep Convolutional Networks for Gesture Recognition in American Sign Language.

[9] Panwar, Prateek. (2016). Talking Hands: RNN-based Sign Language Recognition.

[10] Nikolskaia, Kseniia & Ezhova, Nadezhda & Sinkov, Anton & Medvedev, Maksim. (2018). Skin Detection Technique Based on HSV Color Model and SLIC Segmentation Method.

[11] International Journal of Advance Engineering and Research Development (IJAERD) Volume 5, Issue 02, February-2018, e-ISSN: 2348 - 4470, print-ISSN: 2348-6406

[12] Mishra, Shubham Kr, Sheona Sinha, Sourabh Sinha, and Saurabh Bilgaiyan. &quot;Recognition of Hand Gestures and Conversion of Voice for Betterment of Deaf and Mute People.&quot; In International Conference on Advances in Computing and Data Sciences, pp. 46-57. Springer, Singapore, 2019.

[13] Khan, Saleh Ahmad, Amit Debnath Joy, S. M. Asaduzzaman, and Morsalin Hossain. &quot;An Efficient Sign