# Real-Time Indian Sign Language to Speech Interpreter for deaf and mute people

Faraz Suhail[1]
*School of Computer Science (SCOPE)*
*VIT Chennai*

Zara Iqbal[1]
*School of Computer Science (SCOPE)*
*VIT Chennai*

Hardik Govil[1]
*School of Computer Science (SCOPE)*
*VIT Chennai*

***Abstract*— Communication for the majority of people is not difficult. It should be the same way for the deaf and mute people. Inability to speak is indeed a true disability and such people use different modes to communicate with others. There are a number of methods available for their communication and one such method of communication is sign language. Developing an Indian sign language application for deaf and mute people can prove to be vital, as they'll be able to communicate easily with even those who don't understand sign language. Our project offers a platform for bridging the communication gap between speaking people and deaf and dumb people with the help of sign language. The main focus of this work is to create a vision-based system to identify Indian sign language gestures, keeping resource intensity of the application bare minimum. The project deals with designing and implementing a user-friendly and more intuitive way of communication. The paper primarily focuses on increasing the overall efficiency of the algorithm for Indian Sign Language recognition by leveraging Image Processing to accurately predict.**

***Keywords*— *Image processing, deep learning, computer vision, hand gesture recognition, Indian sign language, convolutional neural networks.***

## I. Introduction

14,000,000 people in India have speech/hearing impairment. People with speech and hearing impairments often rely on sign language to communicate with others but most of the general population cannot understand sign language and sign language itself is difficult to learn. Eventually, people who are hearing impaired are often left behind, creating barriers to inclusion. We want to make it easy for thousands of deaf people across the Indian subcontinent to be independent for their daily communication needs and we think enabling a communication channel would alleviate their problems.

Our project aims to develop an application and train the model which, when shown a real time video of hand gestures of the Indian Sign Language, displays the output for that particular sign in text format on the screen. Given a hand gesture, the application detects predefined Indian sign language (in real time through video feed) and provides the facility for the user to be able to form words and sentences thereby facilitating a smooth conversation. Our project classifies the Indian sign languages with a significant accuracy in order to offer a spontaneous communication between the deaf and hearing.

We have implemented 36 symbols(A-Z, 0-9) of the ISL in our project. The principle aim of our project is to design and develop a digital autonomous platform for the hearing impaired and the English speakers. The project scope pertains to the work of converting the sequence of gestures into text i.e., word and sentences and then converting it into the speech which can be heard. In our method, the hand is first passed through a set of image filters and after the filters are applied, the hand is passed through a classifier which predicts the class of the hand gestures.

## II. Literature Survey

In 2016, D. Naglot & M. Kulkarni [7] developed a Real time sign language recognition system using the leap motion controller. The system proposed captures signs of American Sign Language using new digital sensor called "Leap Motion Controller" which is 3D non-contact motion sensor that tracks and detects hands, fingers, bones and finger-like objects. The proposed system used Multi-Layer Perceptron (MLP) neural network with Back Propagation (BP) algorithm to build a classification model which takes feature set as input. Multi-Layer Perceptron (MLP) neural network is used to recognize different signs. Recognition rate of the proposed system is 96%.

In 2020, S.C. Bodda, P. Gupta, G. Joshi & A. Chaturvedi [8] developed a new architecture for hand-worn Sign language to Speech translator. The Designing and implementation of Smart glove is described in this project, a hand-worn hardware device capable of translating American Sign Language gestures into English speech by tracking the finger's orientation, gestures and hand motion. It uses hardware sensors like the Flex, the Accelerometer and gyroscope and intelligent software to capture and translate the gestures into speech.

In 2020, T. Raghuveera, R. Deepthi, R. Mangalashri & R. Akshaya [9] developed a depth-based Indian sign language recognition system using microsoft kinect. The system captures hand gestures through Microsoft Kinect, the dataset includes single-handed signs, double-handed signs and fingerspelling, totaling to about 4600 images. The hand region was accurately segmented and hand features are extracted

using Speeded Up Robust Features, Histogram of Oriented Gradients and Local Binary Patterns to recognize the hand posture. The system ensembles the three feature classifiers trained using Support Vector Machine to improve the average recognition accuracy up to 71.85%. The system then translates the sequence of hand gestures recognized into the best approximate meaningful English sentences.

In 2020, R. Rastgoo, K. Kiani & S. Escalera [6] developed Video-based isolated hand sign language recognition system using a deep cascaded model. The purpose of this project was to classify signed American Signed Language letters using simple images of hands taken with a personal device such as a laptop webcam. Make a future implementation of a real time ASL-to-oral/written language translator. Project was structured into 3 distinct functional blocks: Data Processing, Training, Classify Gesture. Compiled dataset of American Sign Language (ASL) called the ASL Alphabet from Kaggle was the primary source of data for this project. Data preprocessing was done using the PILLOW library and sklearn.decomposition library, which is useful for its matrix optimization and decomposition functionality. Adam optimizer and Cross Entropy Loss were used to train the modules.

## III. PROPOSED WORK

The architectural description involves the modelling and representation of various phases of the system:

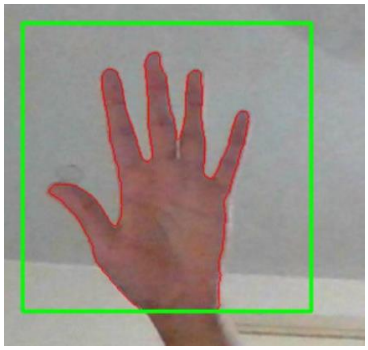
i. Dataset

MNIST data set consisting of 26 alphabets of the Indian Sign Language and 10 numbers is used. The data set consists of 1200 images for each character.

ii. Data Pre-processing

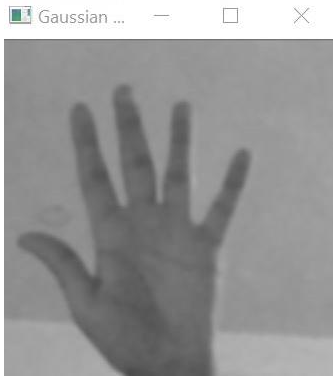
Image pre-processing is done to improve the image data suppressing unwilling distortions and enhances some image features important for further processing.

Images are pre-processed both during training stage as well as recognition stage. The following filters are applied to the images/ frames to enhance the features and consequently help the system recognize better:-



RGB to grayscale


Gaussian Blur: reducing the size of an image (downsampling) by reducing the image's high-frequency components




Thresholding: If the pixel value is smaller than the threshold(25), it is set to 0, otherwise it is set to a maximum value(255). Applying binary thresholding makes the objects completely black and white. The objects of interest, and their border will be completely white having the same color intensity. binary threshold converts it to simple binary image (black and white)


Contour Detection: When we join all the points on the boundary of an object, we get a contour. Finding contours is like finding white object from black background i.e. foreground extraction. Hand will be white and bg will be black. Using contour detection, we can detect the borders of fingers/ hand. contours is a Python list of all the contours in the image. Each individual contour is a Numpy array of (x,y) coordinates of boundary points of the object. returns countour(object) with max area

Median Blur: Run through each element of the image and replace each pixel with the median of its neighboring pixels. It processes the edges while removing the salt and pepper noise (only a few pixels are noisy, but they are very noisy). Sharpen the edges by blending the images.


Morphological transformation: used for removing noise. The 5x5 kernel with 1s slides through the image. morphologyEx is erosion followed by dilation. Erosion erodes away the boundaries of foreground object. Dilation increases the object area. Erosion removes white noises, but it also shrinks our object. So we dilate it.


Apply automatic Canny edge detection using the computed median.

## IV. Experimental Setup

### IV.I CNN Model:

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 100, 100, 32)      320
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 98, 98, 32)        9248
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 49, 49, 32)        0
_________________________________________________________________
dropout (Dropout)            (None, 49, 49, 32)        0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 49, 49, 64)        18496
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 47, 47, 64)        36928
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 23, 23, 64)        0
_________________________________________________________________
dropout_1 (Dropout)          (None, 23, 23, 64)        0
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 23, 23, 64)        36928
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 21, 21, 64)        36928
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 10, 10, 64)        0
_________________________________________________________________
dropout_2 (Dropout)          (None, 10, 10, 64)        0
_________________________________________________________________
flatten (Flatten)            (None, 6400)              0
_________________________________________________________________
dense (Dense)                (None, 512)               3277312
_________________________________________________________________
dropout_3 (Dropout)          (None, 512)               0
_________________________________________________________________
dense_1 (Dense)              (None, 36)                18468
=================================================================
Total params: 3,434,628
Trainable params: 3,434,628
Non-trainable params: 0
```

1st Convolution Layer : The input picture has resolution of 100x100 pixels. It is first processed in the first convolutional layer using 32 filter weights (3x3 pixels each). This will result in a 100X100 pixel image since padding parameter is set to ‘same’.

2nd Convolution Layer : Second convolutional layer uses 32 filter weights (3x3 pixels each). This will result in a 98X98 pixel image, one for each Filter-weights.

1st Pooling Layer : The images are down sampled using max pooling of 2x2 i.e., we keep the highest value in the 2x2 square of array. Therefore, our picture is down sampled to 49x49 pixels. We are using a dropout layer of value 0.25 to avoid overfitting.

3rd Convolution Layer : The input picture has resolution of 49x49 pixels. This will result in a 49X49 pixel image since padding is the same.

4th Convolution Layer : Fourth convolutional layer uses 32 filter weights (3x3 pixels each). This will result in a 47X47 pixel image, one for each Filter-weights.

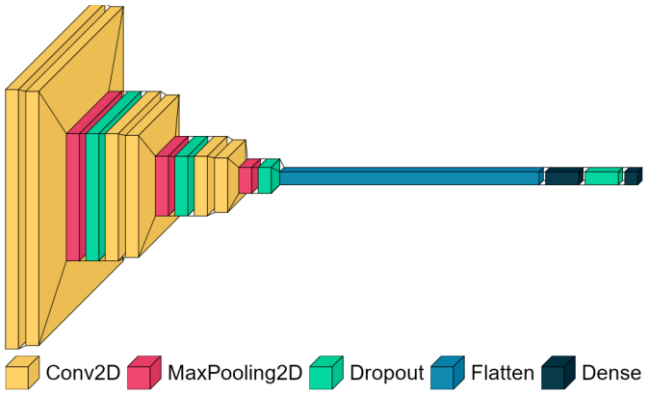
2nd Pooling Layer : The resulting images are down sampled again using max pool of 2x2 and is reduced to 23X23 resolution of images. We are using a dropout layer of value 0.25 to avoid overfitting.

5th and 6th Convolution Layer : The input picture has resolution of 23x23 pixels. This will result in a 21X21 pixel image.

3rd Pooling Layer : The resulting images are down sampled again using max pool of 2x2 and is reduced to 10X10 resolution of images. We are using a dropout layer of value 0.25 to avoid overfitting.

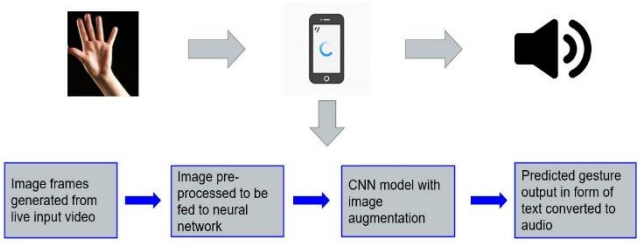
1st Densely Connected Layer : Now these images are used as an input to a fully connected layer with 512 neurons and the output is reshaped to an array of 10x10x32 =3200 values. The input to this layer is an array of 3200 values. The output of these layer is fed to the 2nd Densely Connected Layer. We are using a dropout layer of value 0.5 to avoid overfitting.

2nd Densely Connected Layer : Now the output from the 1st Densely Connected Layer is used as an input to a fully connected layer with 36 neurons i.e., number of classes we are classifying (alphabets and numbers).




After the training is completed, the system detects only the relevant parts from the camera window. The camera window filters out the hand from the background. Background subtraction removes the irrelevant background pixels from the window. By applying hand detection, the system disregards the region which is not required and thus reduces the amount of calculation needed during training and testing of the system. The system computes the location of the hand by adjusting the range on the trackbars. Then the camera starts capturing the gestures and displays the corresponding alphabet on the screen.

Gaussian blur filter and threshold is applied to the frame taken with OpenCV to get the processed image after feature extraction. A set of image processing filters are applied to this image.




This processed image is passed to the CNN model for prediction and if a letter is detected for more than 90 frames then the letter is printed and taken into consideration for forming the word. Whenever the count of a letter detected exceeds a specific value and no other letter is close to it by a threshold, we print the letter and add it to the current string(In our code we kept the value as 90 and difference threshold as 25). Otherwise, we clear the current dictionary which has the

count of detections of present symbol to avoid the probability of a wrong letter getting predicted.

These letters can also be converted into sentences by giving the spaces between the alphabets. The words and sentences formed are converted to English speech for facilitating a spoken communication with the deaf and dumb. The stream of characters is appended to form a word until the user stops showing gestures. The user is provided a delimiter and until that delimiter is encountered every scanned word will be appended with the previous results forming a stream of meaning-full sentences.

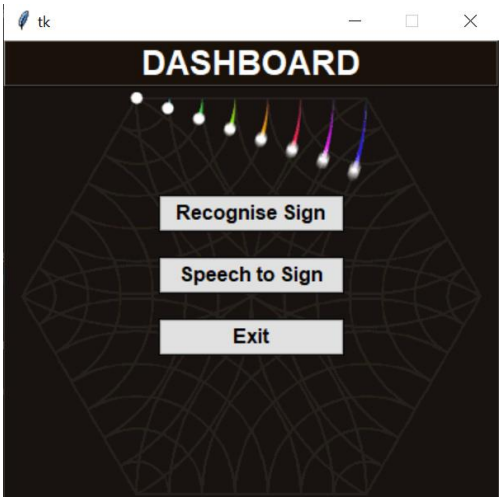
A python library SpellChecker is used to suggest correct alternatives for each (incorrect) input word and the system automatically matches the current word to append it to the current sentence. This helps in reducing mistakes committed in spellings and assists in predicting complex words.

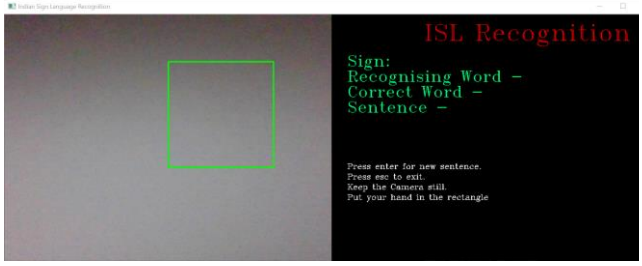
### IV.II GUI Design

The frontend is built on Tkinter and Tensorflow backend is used. Our application has two main modules:-

Speech to Sign: detects spoken words/sentences and displays the appropriate representation in Indian Sign Language. This module can be used for learning Sign Language.

Sign Recognition: simply detects the gesture which is stored into a buffer so that a string of character could be generated forming a meaningful word. Additionally, the word is checked for grammatical errors and corrected with the closest possible word. These set of words are further appended to form a meaningful sentence.



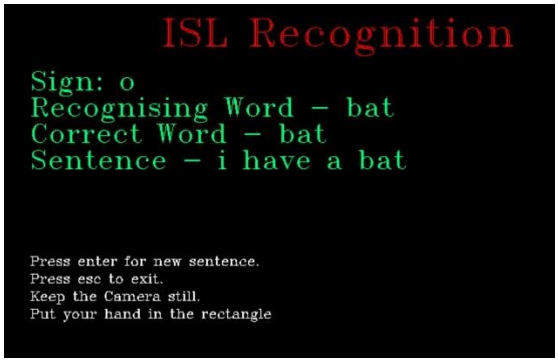



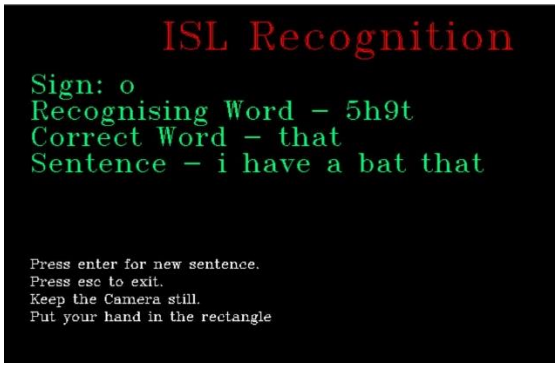

The project GUI has three windows to show:

One main window displays the video feed.

The second screen focuses on the hand region, eliminating the other parts of the background i.e., cropping the useless part of the view.

When the trained model identifies the gesture, the character and words are printed on the third screen.



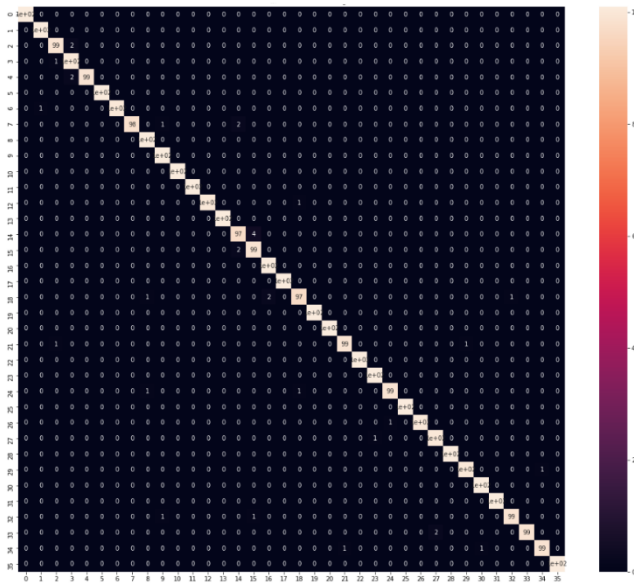




## V. RESULTS

We tried working with the OpenCV Histogram method Saha, 2018 [13], but the histograms were an unnecessary overhead. It required to set a new histogram for the model, every time there was a change in lighting conditions. This was the biggest turndown when we compared it with a simple OpenCV functions-based system. So, we bent towards the process of filtering with a mask and then blurring. It was a fast and smooth method and gave even better results.

In the 2021 publication of Sign Language Interpreter[14], gesture made by the hand inside the recognition area in each frame is matched with the pre-classified data available in the dataset and when a match is declared, the corresponding alphabet/number is given as a text output, easily understandable by a normal person who doesn't know the sign language. The biggest limitation in the paper is that despite using 2 CNN models with various layers, the model still falls short in terms of accuracy. The main reason behind this is the absence of background subtraction in their model. Another flaw in their model is that they have not focused a lot on preprocessing the images which resulted in 98% accuracy.

The following is our project's confusion matrix for the CNN model:-



We are able to improve our prediction after implementing the various layers of image pre-processing on the hand in which we filter the images before training and testing to improve the overall efficiency of the model. We have achieved an accuracy of 99.09% in our model using our algorithm which is a better accuracy than most of the current research papers on Indian sign language.

## VI. CONCLUSION

Hand gestures are a powerful way for human communication, with lots of potential applications in the area of human computer interaction. In this report, a fully functional real time vision based Indian sign language recognition for the deaf and mute people has been developed for ISL alphabets and numbers. Prediction has been improved after implementing the autocorrect feature for words in which we verify the spellings and predict the closest possible word which is more relevant in the sentence.

This paper elaborates on an efficient way to recognize Indian sign gestures, which is faster and robust than previous works. Main contributions of the study to the Indian Sign Language Recognition field:

A fast and more accurate method to recognize Hand Gestures using native OpenCV functions.

Improving the pre-processing to predict gestures in low light conditions with a higher accuracy.

A Deep Learning model capable of predicting ISL Gestures with 99.09% accuracy.

Reduce the time-consuming overhead of the histogram dependence for changing lighting conditions.

All the Indian sign language gestures are represented with bare hands, so it eliminates the problem of using any artificial devices for interaction. One thing should be noted that our model doesn't uses any Kinect devices as our primary goal was to create an application which can be used with readily available resources. Our prediction model uses a normal cam of the laptop hence it is great plus point as it will be compatible with almost all laptops available in the market today.

Our project demonstrates the use of machine learning to develop an Indian Sign Language Recognition System. Convolutional neural networks can be used to accurately recognize different gestures of the Indian sign language, with background subtraction. Our method provides 99.09% accuracy for the 36 classes of the Indian Sign Language.

## VII. REFERENCES


[1] Dutta, K. K., & GS, A. K. (2015, December). Double handed Indian Sign Language to speech and text. In *2015 Third International Conference on Image Information Processing (ICIIP)* (pp. 374-377). IEEE.

[2] Ghotkar, A. S., & Kharate, G. K. (2017). Study of vision based hand gesture recognition using Indian sign language. *International journal on smart sensing and intelligent systems*, *7*(1).

[3] Rajam, P. S., & Balakrishnan, G. (2011, September). Real time Indian sign language recognition system to aid deaf-dumb people. In *2011 IEEE 13th international conference on communication technology* (pp. 737-742). IEEE.

[4] Sawant, Shreyashi Narayan. "Sign language recognition system to aid deaf-dumb people using PCA." *Int. J. Comput. Sci. Eng. Technol.(IJCSET)* 5, no. 05 (2014).

[5] Badhe, P. C., & Kulkarni, V. (2015, November). Indian sign language translator using gesture recognition algorithm. In *2015 IEEE International Conference on Computer Graphics, Vision and Information Security (CGVIS)* (pp. 195-200). IEEE.

[6] Rastgoo, R., Kiani, K., & Escalera, S. (2020). Video-based isolated hand sign language recognition using a deep cascaded model. *Multimedia Tools and Applications*, *79*, 22965-22987.

[7] Naglot, D., & Kulkarni, M. (2016, August). Real time sign language recognition using the leap motion controller. In *2016 International Conference on Inventive Computation Technologies (ICICT)* (Vol. 3, pp. 1-5). IEEE.

[8] Bodda, S. C., Gupta, P., Joshi, G., & Chaturvedi, A. (2020). A new architecture for hand-worn Sign language to Speech translator. *arXiv preprint arXiv:2009.03988*.

[9] Raghuveera, T., Deepthi, R., Mangalashri, R., & Akshaya, R. (2020). A depth-based Indian sign language recognition using microsoft kinect. *Sādhanā*, *45*(1), 1-13.

[10] Ekbote, J., & Joshi, M. (2017, March). Indian sign language recognition using ANN and SVM classifiers. In *2017 International conference on innovations in information, embedded and communication systems (ICIIECS)* (pp. 1-5). IEEE.

[11] Reshna, S., & Jayaraju, M. (2014). Indian Sign Language Recognition System–A Review. In *International Conference on Signal and Speech Processing, ICSSP* (Vol. 14).

[12] Nath, G. G., & Arun, C. S. (2017, April). Real time sign language interpreter. In *2017 IEEE International Conference on Electrical, Instrumentation and Communication Engineering (ICEICE)* (pp. 1-5). IEEE.

[13] Saha, D.. (2018, May 9). Sign-Language (Version 1). figshare. https://doi.org/10.6084/m9.figshare.6241901.v1A very simple CNN project.

[14] Ghali Upendra1, Kokkiligadda Bhuvanendra 2, D Gayathri3(2021, Apr 4). Sign Language Interpreter using Convolution Neural Network.