INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN SCIENCE AND ENGINEERING, VOL.3, NO.05, MAY 2022.

Sign Language to Text Convertor Tool

Pranay Pathrabe[1], Umang Ghatbandhe[1], Sangita Mondal[1], Sejal Parmar[1], Pallavi Chaudhari[2]

[1]Student, Dept. of Information Technology, Priyadarshini College of Engineering, Nagpur, Maharashtra, India.
[2]Assistant Professor, Dept. of Information Technology, Priyadarshini College of Engineering, Nagpur, Maharashtra, India.

Umang Ghatbandhe: umangghatbandhe0@gmail.com

Abstract: - signal language is one of the oldest and most natural shape of language for communique, however considering most people do no longer realize sign language and interpreters are very difficult to come back by we've give you an actual time method using neural networks for fingerspelling based American sign language. In our approach, the hand is first surpassed thru a clear out and after the clear out is carried out the hand is exceeded via a classifier which predicts the class of the hand gestures. Our technique gives 95.7 % accuracy for the 26 letters of the alphabet.

Key Words: — ***Feature Extraction, Artificial Neural Networks (ANN), Convolution Neural Network (CNN), Pooling, Dataset Generation, ROI, RGB, Text to Sign Recognition, Hand Gesture Detection, ReLU, Machine-Learning, python.***

I. Introduction

American Sign Language is a prominent Sign Language, it is used for communication by speech impaired people, so they cannot use the spoken languages which is why it the only way remain for them to communicate is through Sign language. Communication is a process of exchanging ideas as well messages in a variety of ways such as speech, gestures, behaviors and visuals. Deaf and dumb people (D&M) use their hands to express themselves, using different hand gestures to express their thoughts and feelings to other people. These are gestures messages are dynamic and these symbols are understood by it visually. This silent communication between the deaf and the dumb is called Sign language.

In our project we focus on producing a visual sign to text convertor interface which is used to recognize the sign language and form the corresponding text to the sign combining each sign. Gestures intended to train in our project you as given below.

Manuscript revised May 12, 2022; accepted May 13, 2022.
Date of publication May 15, 2022.
This paper available online at www.ijprse.com
ISSN (Online): 2582-7898; SJIF: 5.59

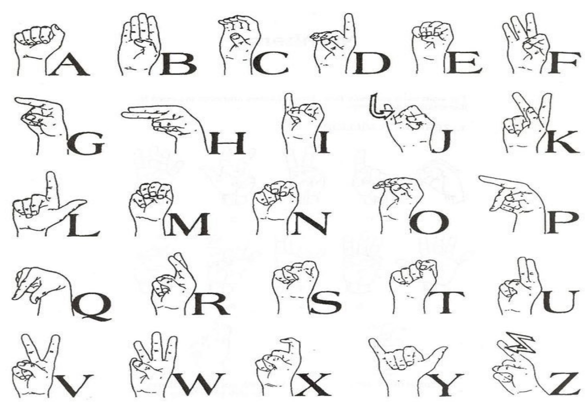

Fig.1. ASL Sign

II. Literature Survey

2.1 Sign Acquiring Methods:

2.1.1 Leap Motion

Leap motion controller is a form of sensor which recognize the hand motion and converts the sign to pc commands. It consists of two IR cameras and 3 infrareds LED's. LED generates IR light signal and digicam generates three hundred frames in step with 2d of pondered records. these indicators are sending to the pc via USB cable for in addition processing. P. Karthick et al. [5] used model that rework Indian signal language into text the use of soar version. The bounce tool detects the records like point, wave, reach, grasp that is generated by using a bounce movement controller.

mixture of DTW and IS set of rules is used for conversion of hand gesture into text. Neuron community turned into used for training the data. [6] used bounce motion controller for recognition ofAustralian sign language. jump movement c ontroller used to feel the hand motion and convert that hand movement into computer instructions. synthetic neuro network is used for schooling symbols. The downside of that gadget was low accuracy and fidelity.

2.1.2 Kinect Sensor

Kinect is Microsoft movement sensor with Xbox 360 gaming console proven in figure.2. it includes RGB camera, depth sensor and multi-array microphone. It acknowledges facial motion and speech. Cao doing et al. [6] used Microsoft Kinect to understand American sign language. depth digital camera is Kinect sensor used to locate ASL alphabet. Distance adaptive scheme turned into used for feature extraction. help vector gadget and RF classifier set of rules used for type purpose.

schooling of statistics was completed using ANN net work. The accuracy of the system changed into 90%. uan yao et al. [7] used kinect sensor for popularity of hand gesture. first of all, it detects hand movement after which matched with counter model. second venture became to locate multi color gl ove and detect exceptional coloration regions. Gausian shade model used for traing facts and in keeping with pixel classifier used for category. This system has one drawback this is restrained accuracy.

III. METHODOLOGY

We have implemented a vision-based approach to our system. Every sign is displayed in front of the camera with an empty hand that completely removes the use of electrical or sensor equipment for use.

3.1 Data Set Generation:

In the project we tried to use a predefined data set from various open sources, but we found that the images of these data set were not meeting our requirements. All the data images we could find were in RGB format. Therefore, we decided to generate our own set of data. The procedure we carried out to create our data set are as given below:

With the help of OpenCV library we successfully generated the data. The first step is to take approximately 800 pictures of each symbol in ASL for training purposes and about 250 pictures for each symbol for testing purposes. first, we captured each frame displayed on our machine's webcam. In each frame we present a region of interest (ROI) represented by a square with a blue border as shown in the picture below.

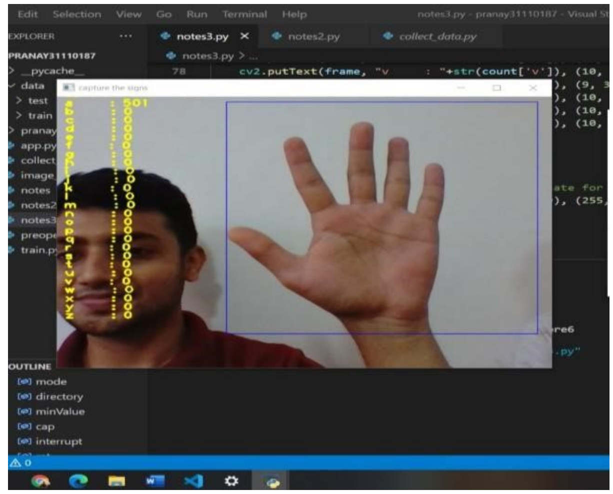

Fig.2: ROI (Region of Interest)

From the above window photograph, we extracted our ROI (place of hobby) that is in RGB (red inexperienced blue channel) and similarly it will be converted to gray scale photo as shown under.

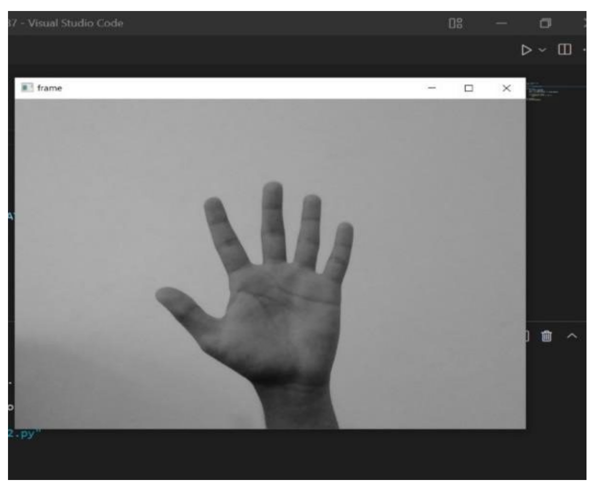

Fig.3: Gray scale image

ultimately, we carried out a gaussian blur filter out to the picture which aided us in extracting distinct functions of the image. Resulted photo after making use of gaussian blur filter is proven beneath.

Fig.4. output image after image pre-processing.

3.2 Gesture Classification:

The method which we are using on this venture is as follows: The approach includes two layers of set of rules to expect the hand gesture of the consumer.

- *Layer1 algorithm:*

  We apply gaussian blur filter and threshold to the image body to the output frame from OpenCV, in order to get the last processed picture after the application of function extraction.
- Result from previous step is forwarded to the CNN for the purpose of prediction. If a character is predicted in more then 50 frames, then the individual is expressed and considered to construct a phrase three. the distance among word is indicated by way of a clean image and taken into consideration.

Layer 1:

3.3 Convolution Model:

- CNN First Layer: All inserted images are 128x128 pixels. It is first processed in the first layer of CNN (Convolution neural network) using 32 filter weights (3x3 pixels each). Which will give the image effect of 126x126 pixels, Weights of one filter each.
- The first layer of merging: Photo subtraction is achieved with 2x2 high-resolution integration. In a 2x2 square of the same members holds a very large number holds a very large amount of it. As a result, our image will be taken down to 63x63 pixels.
- CNN Layer 2: After assembling 63 x 63 pixels the image is placed on the second layer of CNN. The inserted image is processed in the second layer of conversion using 32 filter weights (3x3 pixels each). which will give a 60x60-pixel image to the effect.
- Second Layer of Blending: Using a large 2x2 pool we re-sample the resulting images and as a result we get the image reduced to 30 x 30 adjustment.
- 1st Highly connected layer: Outputs now form the previous layer of integration that we provide as an embedded (embedded) layer in a fully integrated layer consisting of 128 neurons at a instance, and output from CNN layer 2 is set at 30x30x32 = 28800 the same values. The entries for this layer have a list of 28800 values. The layer effect is given by Dense Layer 2. To avoid overheating, we use a drop layer that covers a value of 0.5.
- Highly Connected Layer 2: The highly connected layer 1 output is provided as input to a fully integrated layer consisting of 96 neurons.
- Final Layout: The final layer takes the input into the Second Dense Layer, which includes a number of neurons equal to the class we are identifying (such as alphabetical and empty markings).

3.4 Rectified Linear Unit Activation Function:

It is used (Adjusted Line Unit) throughout the layer (convolutional neurons and fully connected neurons). All input pixels (x, 0) are calculated by ReLu. It helps a very weak feature due to not having additional linearity in the formula. In addition to removing the gradient problem, training is accelerated by reducing the calculation (calculation) time.

3.4.1 Backup layer:

With the use of Max integration in an embedded image that includes (2, 2) the size of the pool and the function of ReLU activation. This reduces the number of parameters, thereby reducing the cost of calculation and overlay.

3.4.2 Leaving layers:

The overuse barrier occurs immediately after training, when the weighted network is converted to training models, after providing a new model as input network performance is poor. The activation set is randomly selected from this layer by keeping the value zero. Due to the precise output of a particular model the network must provide the correct output Even if some of the openings are removed, the correct result should be given to the network [5].

3.4.3 Optimizer:

To update the model in response to output loss function we used Adam optimizer. Combining the features of the two extensions of the descending algorithms of the stochastic gradient.

Layer 2 Algorithm:

This layer works to separate the seemingly same sign.

- If we encounter multiple sets of symbols, it will show the same character in the find. Then use the next step.
- We use a separator made for those groups of symbols, then we distinguish between these same symbols.

Layer 2:

Two layers of algorithms are used to verify and obtain identical symbols to obtain additional information in order to obtain the displayed signal. Through experimentation we saw that the following signs were not as accurate as they were intended to be, but were indicative of other signs that should not be shown in a given sign:

- For A: E and S
- For P: N and Q
- For I: T, D, K

In order to catch the above errors, we see, we have created three different dividers to separate these sets:

- {A, E, S}
- {T, K, D, I}
- {P, N, Q}

3.5 Sentence Formation

- whilst the remember of a letter is recognized, that it is going past the particular value and no different letter is near to it by way of a threshold, after that we print the letter and upload it to the modern-day string (In our python code we set the cost as 50 and the value of distinction threshold as 20).
- otherwise, we dispose of the modern-day dictionary that have the matter of prediction of cutting-edge symbol to avoid the chance of predicting a wrong letter.
- The depend of a blank is detected exceeds an accurate value and further to it, if the contemporary buffer is empty no areas are diagnosed.
- In different case it detects the cease factor of a phrase by using displaying a space and the cutting-edge gets appended to the sentence.

Autocorrect function:

There's a python library for checking and autocorrecting the spellings knowns as Hunspell_suggest, which propose synonyms or correct options for the word which has to be revealed.

The consumer can modify or alternate his phrase with the aid of deciding on an appropriate phrase from the given pointers by means of the Hunspell module. As a result, there is reduction in mistakes which could come across in spellings and guide in detecting complicated phrases.

3.6 Training and Testing:

First, we convert our RGB images into greyscale and apply a gaussian blurring filter to remove unwanted noise from the image. We then use a flexible limit to extract only the details of our hand back and resize our images to 128 x 128 pixels. After all the image processing steps, the prediction layer evaluates and determines which image will be classified and at what stages. As a result, the output is usually from 0 to 1 so the total value in each class is added to 1. With the help of the SoftMax project, we were able to achieve this. First, the effect of the forecast layer will be slightly different than the actual value.

Using labeled data, we have trained networks to make them better. Performance measurement used in cross entropy classification. It gives a positive value that is not the same as a labeled value and the value is exactly zero if it is the same as the labeled value, all done on an ongoing basis.

To reduce the value close to zero, so we have improved cross-entropy. For this purpose, we adjust the weight of our neural network to do this in our network layer. To calculate cross-entropy, TensorFlow has a built-in function. At last, cross-entropy activity is calculated. upgrading with optimizer called Adam Optimizer with gradient downtime.

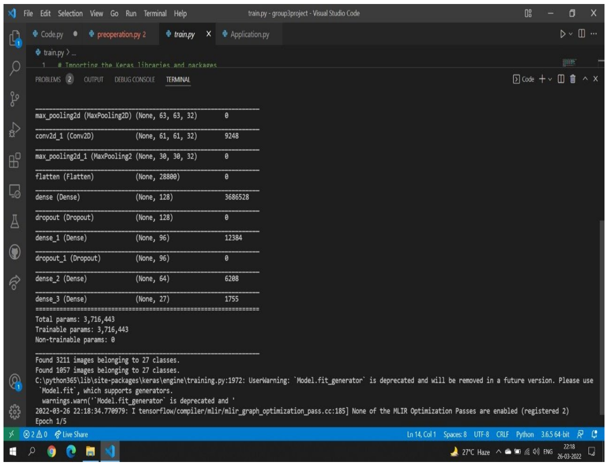

Fig.5. Training Results for CNN Model

IV. RESULTS AND DISCUSSION

We found 95% accuracy in our model using only 1 layer of our algorithm, and using a combination of layer 1 and layer 2 we get 98.0% accuracy, which is better than most current American Sign Language research papers.

One thing to note is that our model does not use any background removal algorithm while the other models above do just that. Therefore, when we try to use a different domain for our project the accuracy may vary. Many other projects use sensory devices but our main goal was to create a project that could be used with easily accessible resources. Below is a demo of the demo that shows how the sign-to-text feature works.

						P	r	e	d	i	c	t	e	d		V	a	l	u	e	s					
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
	A	147	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	2	0	0
	B	0	139	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0
	C	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	D	0	0	0	145	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	E	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	F	0	0	0	0	0	135	0	0	0	0	0	4	0	0	0	0	0	1	0	0	2	10	0	0	0
C	G	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
o	H	1	0	0	0	0	0	7	143	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1
r	I	0	0	0	33	0	0	0	0	108	0	2	0	0	0	0	0	0	0	0	7	1	0	0	0	0
r	J	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	K	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c	L	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0
t	M	0	0	0	0	0	0	0	0	0	0	2	0	152	0	0	0	0	0	0	0	0	0	0	0	0
	N	0	0	0	0	0	0	0	0	0	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0
V	O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	154	0	0	0	0	0	0	0	0	0	0
a	P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0
l	Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	147	1	0	0	0	0	0	0	0
u	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0
e	S	0	0	0	0	1	0	0	0	0	0	0	0	0	1	10	0	0	0	132	0	0	0	0	8	0
s	T	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	151	0	0	0	0	0
	U	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	35	0	0	115	0	0	0	0
	V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151	1	0	0
	W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	149	0	0
	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	148	0
	Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151
	Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Algo 1

Fig.6. Results for Algorithm 1

						P	r	e	d	i	c	t	e	d		V	a	l	u	e	s					
		A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
	A	147	0	0	0	0	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	0	2	0	0
	B	0	139	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0
	C	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	D	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	E	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	F	0	0	0	0	0	135	0	0	0	0	0	4	0	0	0	0	0	0	0	0	3	10	0	0	0
C	G	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
o	H	1	0	0	0	0	0	7	143	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	1
r	I	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
r	J	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
e	K	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0	0
c	L	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0	0	0	0
t	M	0	0	0	0	0	0	0	0	0	0	2	0	152	0	0	0	0	0	0	0	0	0	0	0	0
	N	0	0	0	0	0	0	0	0	0	0	0	0	0	152	0	0	0	0	0	0	0	0	0	0	0
V	O	0	0	0	0	0	0	0	0	0	0	0	0	0	0	154	0	0	0	0	0	0	0	0	0	0
a	P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0
l	Q	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	147	1	0	0	0	0	0	0	0
u	R	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0	0	0	0
e	S	0	0	0	0	1	0	0	0	0	0	0	0	0	0	10	0	0	0	133	0	0	0	0	8	0
s	T	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	151	0	0	0	0	0
	U	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0	0	0
	V	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151	1	0	0
	W	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	149	0	0
	X	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	148	0
	Y	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	151
	Z	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Algo 1 + Algo 2

Fig.7. Results for Algorithm 1&2

Fig.8(a): demo sign to text

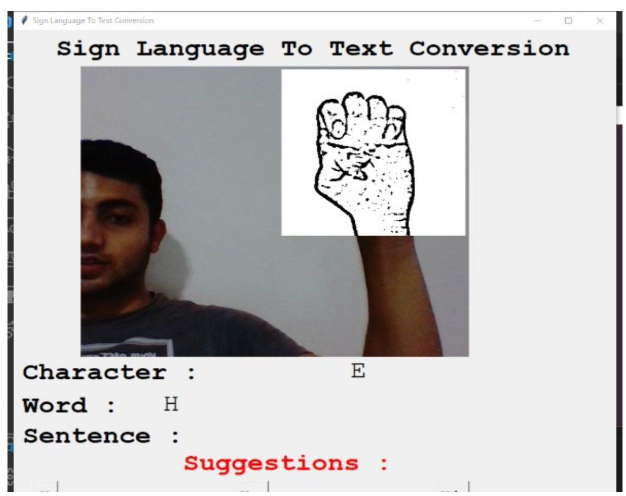

Fig.8(b): demo sign to text

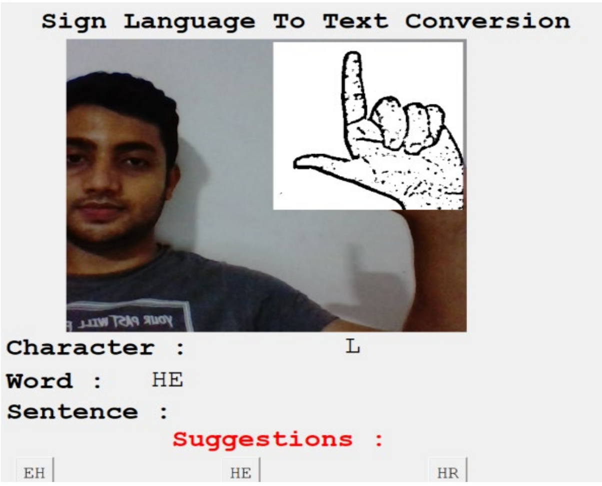

Fig.8(c): demo sign to text

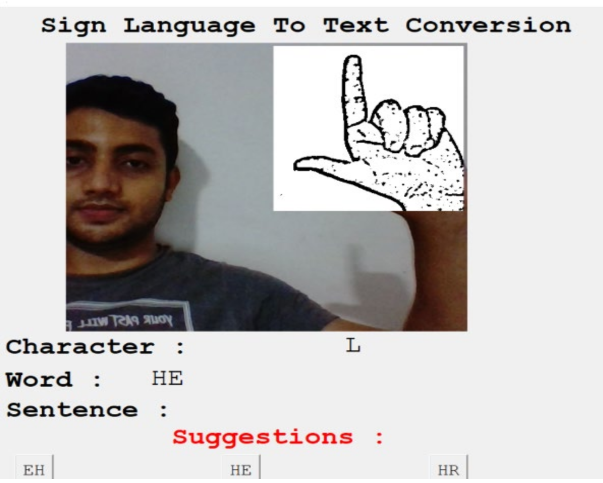

Fig.8(d): demo sign to text

Fig.8(e): demo sign to text

Fig.8(f): Text output for input signs gestures.

V. Conclusion

In this project, a real-time practical concept based on American Sign Language recognition for D&M people developed the alphabet ASL. We have found 98.0% final accuracy to detect the gesture and construct sentence using this model. Model accuracy is improvement by implementing two layers of algorithms in situation, if we come across verifying and predicting very identical signals. In this way we can see almost all the signs as long as they are properly displayed, there is no background noise and adequate lighting. We plan to achieve high accuracy even if there is a complex domain by trying backlink removal algorithms. We also consider enhancing pre-processing to predict touch conditions in low light conditions with high accuracy.

References

[1]. Cao Dong, Ming C. Leu, Zhao Zheng Yin, “American sign language Alphabet Recognition Using Microsoft Kinect”, Computer Vision and pattern Recognition workshop, IEEE conference, pp: 2015.

[2]. Yuan Yao, Yun Fu, “Contour Model based Hand-Gesture Recognition Using Kinect Sensor”, IEEE Transaction on Circuits and System for video Technology, pp: 1-6,2013.

[3]. Rao, G. A., Syamala, 1<., Kishore, P. V. V., Sastry, A. S. C. S. "Deep convolutional neural networks for sign language recognition", (2018).

[4]. Shanableh, Tamer, T. Khaled, "Arabic sign language recognition in user independent mode", IEEE International Conference on Intelligent and Advanced Systems, 2007, pp 597-600.

[5]. Umang: P. Karthick, N. Pratibha, V.B. Rekha, S. Thanalaxmi, “Transforming Indian Sign Language into Text

Using Leap Motion", International Journal of Innovative Research in science, Engineering and Technology, pp. 10906-10908,2014.

[6]. Leigh Ellen potter, Jake Araullo, Lewis Carter," The leap Motion Controller: A view on sign language", Proceedings of the 25th Australian Computer-Human Interaction Conference: Augmentation, Application, Innovation, Collabration.pp.1-4,2013.

[7]. Cao Dong, Ming C. Leu, Zhaozheng Yin, "American sign language Alphabet Recognition Using Microsoft Kinect", Computer Vision and pattern Recognition workshop, IEEE conference, pp: 2015.

[8]. Yuan Yao, Yun Fu, "Contour Model based Hand-Gesture Recognition Using Kinect Sensor", IEEE Transaction on Circuits and System for video Technology, pp: 1-6,2013.