Indian Sign Language Translator

Dr. K. Anitha Sheela[1]
Electronics & Communication Engineering
JNTUH University College of Engineering Hyderabad (JNTUH)
Hyderabad, Telangana
kanithasheela@jntuh.ac.in[1]

Chevella Anil Kumar[2]
Electronics & Communication Engineering
VNR Vignana Jyothi Institue of Engineering & Technology
Hyderabad, Telangana
chevellaanilkumar@gmail.com[2]

Jella Sandhya[3]
Electronic & Communication Engineering
JNTUH University College of Engineering Hyderabad (JNTUH)
Hyderabad, Telangana
sanju.436@gmail.com[3]

S Nadia Begum[4]
Electronics & Communication Engineering
JNTUH University College of Engineering Hyderabad (JNTUH)
Hyderabad, Telangana
nadiabegumshaik@gmail.com[4]

Gaddam Ravindra[5]
Electronics & Communication Engineering
JNTUH University College of Engineering Hyderabad (JNTUH)
Hyderabad, Telangana
gaddamravindra1999@gmail.com[5]

***Abstract*— Communication is crucial in our day-to-day lives. It is the basis for all human interactions. It is how people pass on information to other people and receive information from them. A person with a speaking or hearing problem cannot communicate properly with others. A person who is deaf or dumb usually uses Sign Language to communicate with others. Normal individuals who would not know Sign language, on the other hand, cannot communicate better with them. Indian Sign Language acts as a mode of communication for deaf and dumb people who constitute a hefty portion of the Indian population. Most people struggle to apprehend ISL gestures. As a result, communication gap is created between the hearing and speech impaired and those who do not understand ISL. Translation systems are required to help bridge the gap between the community of people with hearing and speech impairment and the general population. This paper focuses on developing an end-to-end system that can recognize the spoken language and interpret the corresponding speech to animated sign language gestures and that is also capable of converting Indian Sign Language to speech.**

Keywords—Indian Sign Language, Translator, Text to Speech, Speech Recognition, Animation.

I. INTRODUCTION

In India there are about 7 million people who suffer from hearing and speaking disabilities, according to the 2011 Census [1]. The majority of these people have trouble in communicating. So, they use sign language to communicate with others. Because sign languages lack a proper grammar or structure, these signs have very less acceptance outside of the narrow world of these differently abled persons [2]. Communication is difficult for hearing-impaired or speaking-impaired people at public locations like bus stops, railway stations and hospitals because a normal person who is hearing might not comprehend the sign language used by a deaf person to communicate. Normal people cannot share any information to a deaf/dumb person as they might not know gestures of sign language. To improve the interaction of the disabled and non-disabled communities, a sign language translator is a must.

The deaf/dumb community in India uses Indian Sign Language (ISL). Over the last century, Indian Sign Language (ISL) has evolved. It has been taught since 2001 and dates back over a century. Indian Sign Language, like other languages, has its unique grammar. It is independent of the spoken languages like Hindi, English, etc. In ISL, both word level gestures and fingerspelled words are included. The sign language differs from the manual depiction of spoken languages such as English or Hindi. It has some unique and distinguishing features such as: all the sign representations for numbers are fingerspelled, the interrogative sentences having terms like how, what, where, etc. are implied by interjecting these questions at the completion of sentences, the signs 'male or man' and 'female or woman' are preceded in case of family relationship signs.

II. LITERATURE SURVEY

Ali and Sankar [10] designed a system in which the English text is fed as input. The entered text is changed to ISL string which is then converted into ISL gestures. The system architecture has the following components:1) An input module for text translation. 2) To separate words in a sentence a tokenizer is used. 3) A railway enquiries specific ISL symbols repository. The synonym sign will be used if a term doesn't have a corresponding sign mapped to it. 4) A specially designed translator maps all the words to their appropriate symbols. It also screens the words to be translated by removing any that are derogatory, abusive, or do not have a symbol saved. 5) Accumulator accumulates the mapped gestures in the same sequence as present in ISL string.

Pankaj Sonawane [7] in their 2021 ICCCIS paper, proposed a system or an android app in which, upon clicking a button, the Google Speech-to-Text service, which is built-in to Android, was called upon to transform spoken input into parseable strings. Network access is required in the Speech-to-Text service. The Text generated is parsed based on ISL rules. Simultaneously, a humanoid mesh is mapped on a fbx 3D avatar in Blender 3D. The animations are made by capturing motion data and importing them onto the mesh for some of the commonly used Indian Sign Language gestures.

Nishi and Arkav [11] developed an Indian Sign Language to text converter for alphabet and numeric gestures. The alphanumeric Indian Sign Language gestures are static images. For the pre-processing, Grab Cut algorithm is used to find the region of interest consisting of hands and to segment that region. Then, to classify the gestures, the images are trained on a convolutional neural network.

Jayshree and Aishwarya [12] designed a real-time American Sign Language to Text and Speech conversion system where the region of hands is extracted by performing

image processing operations such as contour masking, skeletonization, Canny edge detection and classification of the gestures is done using Support Vector Machine (SVM) on the American Sign Language dataset.

Sarfaraz and Harish [14] worked on the video gestures of Argentinean Sign Language. In the Argentinean Sign Language video gestures (LSA-64) dataset the gesture videos consist of hands covered with coloured gloves. Colour based segmentation is performed to pre-process the dataset i.e., to identify the region of hands based on the colour of the glove. The video is converted to frames and then the frames are trained on CNNs and RNNs to convert Argentinean Sign language to Text.

III. DATASETS

a. *Indian Sign Language Recognition Dataset*

To train the network and create a ISL recognition model, a dataset consisting of gesture videos of 76 ISL signs are recorded in the anechoic chamber of ECE Department, JNTUHUCEH. 50 videos for each sign were recorded, which sum to a total of 380 videos.

The specifications of the dataset are as follows:

- Dataset size: 70.5GB
- Number of gestures: 76
- Number of persons: 10
- Number of videos per person: 5
- Video details:
  - Frame width: 1920
  - Frame height: 1080
  - Duration: 2-4 seconds
  - Frame rate: 50 fps

The gestures are categorized into 4 categories: numbers (0-9), alphabets(A-Z), greetings, and medical terms.

Table I LIST OF GREETINGS DATA IN DATASET

All the best	Bye	Excuse me	Good afternoon	Good evening
Good morning	Good night	Hello	How are you	I am fine
My name is	Nice to meet you	No	Please	Sorry
Thank you	Welcome	What is your name	Yes	

Table II LIST OF MEDICAL TERMS DATA IN DATASET.

Accident	Allergies	Asthma	Blood pressure	Breathe
Cancer	Diabetes	Doctor	ECG	Emergency
Fever	Headache	Health insurance	Heart attack	Hospital
Medicine	Operation	Stomachache	Thermometer	Virus
Vomit				

b. *Animation Dataset*

As a part of this paper implementation, an animation video dataset of 100 Indian Sign Language gestures with resolution 1080p and a frame rate of 60fps is created using "Blender" manually. These gestures are created based on Doctor-Patient conversations. These videos also have audio of the gestures and captions with gesture names. These gestures mainly belong to the following scenarios.

- Greetings
- Body Parts
- Hospital usage word
- Alphabets (A-Z) and
- Numbers (0-9)

The list of all the gestures belonging to different scenarios are tabulated in the table I, II and III. Out of 64 gestures (except alphabets and numbers), 20 gestures are greetings, 22 gestures are body parts and 22 gestures are hospital usage words.

Table III LIST OF SIGNS DATA IN DATASET.

Arm	Beard	Belly	Bones	Ear
Eye	Face	Feet	First aid bag	Hair
Hand	Head	Leg	Lungs	Mouth
Nose	Ribs	Skeleton	Skin	Skull
Spine	Symptoms	Teeth	Throat	

The video gestures of some gestures are depicted in the figures below.

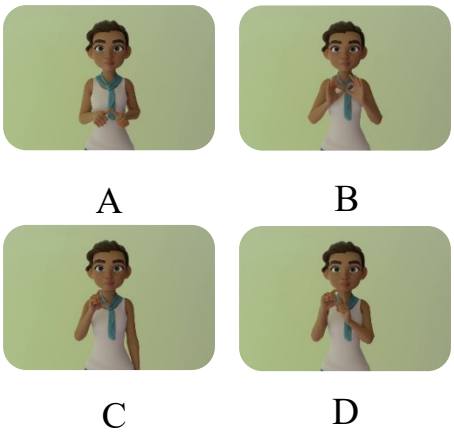


Fig. 1. Animation gestures of some alphabets.

IV. METHODOLOGY

The proposed system is capable of recognizing the Indian Sign Language (ISL) gestures and gives the output speech of corresponding gesture. It is also capable of recognizing the spoken words and displays corresponding ISL gestures as output.

A. *Pre-processing (Mediapipe Hands)*

The gesture video consists of hands and other background too. As only the region of hands is needed, it is necessary to extract the region of hands from the gesture video. So, we pre-process the video frames to extract hands

only. In the pre-processing stage, the video gestures are converted into frames and from each frame, we extract hand landmarks of both hands using Mediapipe Hands model.

MediaPipe Hands is an efficient hand and finger tracking solution [15]. The MediaPipe Hands is a combination of two models working combinedly.

- A palm detector model (called BlazePalm) that operates on the input image and outputs an oriented hand bounding box.
- A hand landmark model that returns 21 high fidelity landmarks based on the input bounding box region defined by the palm detector.

The palm detector model, BlazePalm Detector is also known as Single Shot Detector (SSD). The feed-forward convolutional network serves as the foundation for the Single Shot Detector method. It generates a collection of fixed-size bounding boxes and scores for the presence of hands, then performs a non-maximum suppression to generate the final detections [16].

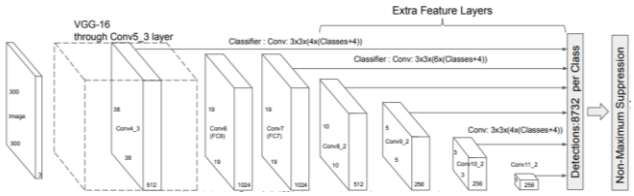
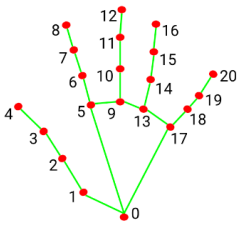


Fig. 2. Architecture of BlazePalm detector

The architecture of the BlazePalm Detector is as shown in the figure 2. For high quality image classification, base network VGG-16 network, which is a standard architecture, serves as the foundation for the early network layers. Instead of using the original VGG fully connected layers, a set of auxiliary convolutional layers are added to allow for the extraction of features at various sizes and to reduce the amount of the input at each succeeding layer.

As palms are smaller objects, the non-maximum suppression algorithm is effective even in situations where the hands are occlusive, such as handshakes.

The hand bounding box output of the BlazePalm detector is fed to the Hand Landmark model. The hand landmark model uses regression, or direct coordinate prediction, to locate 21 3D hand-knuckle coordinates precisely inside the identified hand regions.



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

Fig. 3. Hand landmarks

B. *Inception v3*

Convolution Neural Networks are used to extract the features. In this system we used Inception V3, which is a pre-trained CNN. As large amount of data needed to be processed, Inception V3 is used.

A deep neural network generally consists of different layers i.e., convolution layer, pooling layer, activation layer, fully connected layer, softmax layer, etc. A predefined neural network, namely Inception V3, is widely used for classification purpose. The architecture of Inception V3 is as follows.

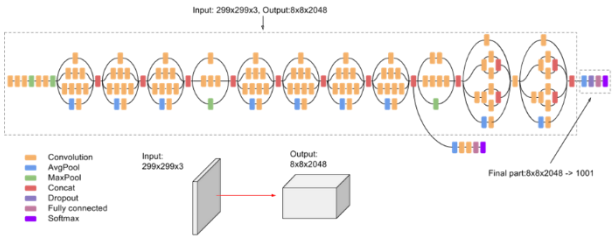


Fig. 4. Architecture of Inception V3

Inception V3 differs from other pretrained image classification networks because of the following [17]:

- Factorization to create more compact convolutions
- Asymmetric convolutions by spatial factorization
- Effective reduction in grid size.

C. *Recurrent Neural Networks and LSTM*

Recurrent Neural Networks (RNN) are artificial neural networks that use sequential data or time series data. Ordinal or temporal problems are frequently addressed by these deep learning systems. RNNs often have problems with "exploding gradients," "vanishing gradients," and their inability to learn "long-term dependencies." Hence, LSTMs (Long Short-Term Memory) are used for our network.

With the use of the feedback, LSTMs are able to handle whole data sequences (such as time series) by retaining relevant information about earlier data points to aid in the processing of later data points.

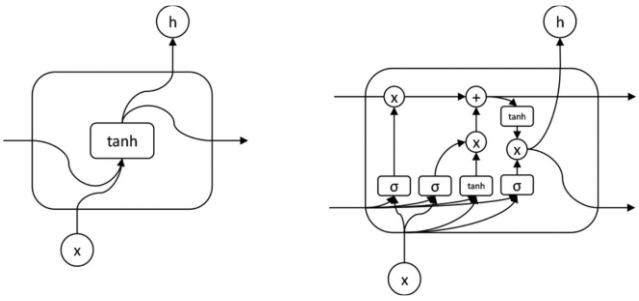


Fig. 5. Left: A unit of RNN. Right: A unit of LSTM

The LSTM outputs at a particular point in time step is determined by three factors:

- Cell state, which refers to the network's present long-term memory.
- Previous hidden state and previous time step output
- The current time step's input data.

A group of "gates" are used by the LSTM module to control the entry, storage, and exit of data in a sequence from the network. It also contains a cell state. A typical LSTM unit consists of three gates: an output gate, an input gate, and a forget gate, as shown in figure below.

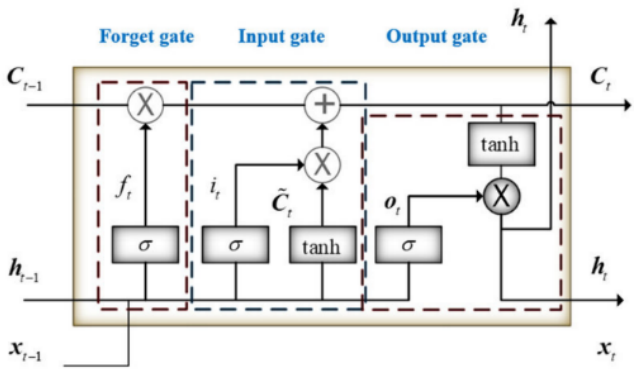


Fig. 6. LSTM Architecture

D. *Text to Speech*

Edinburgh Speech Tools offers a collection of executables that gives standalone application access to speech tool features. The sources of festival, festvox, speech tools, and SPTK are needed to create new synthetic voices using the festival speech synthesis system tool.

The festival architecture is depicted in the figure below.

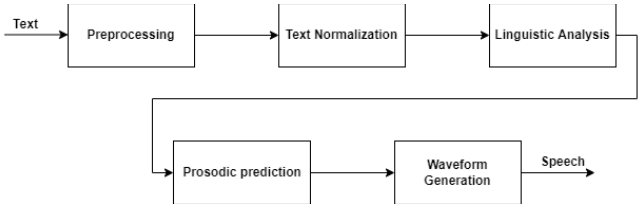


Fig. 7. Festival Architecture for text to speech synthesis

In the context of festival framework, utterance is crucial to produce synthetic speech. The components of utterance are a collection of elements connected by a collection of relations. Words or segments are represented by items. Items are connected in meaningful ways via relations. There may be one or more relations for a given item.

The pre-processing is the initial task. It creates the pronounceable format from the raw input text. The acronyms and numbers are defined in accordance with the context of the text. Depending on the white spaces and punctuation, the text is translated to tokens. Whitespaces can be thought of as separators. Festival transforms the text into an ordered list of tokens, each of which has its own preceding whitespace and following punctuation as token characteristics.

The next stage is to normalize non-standard words. All of the dictionary-available terms are considered as standard words. Non-standard terms include numbers, symbols, abbreviations, and others that do not have their pronunciations listed in dictionaries.

The examples below demonstrate how the raw text is converted into a sequence of pronounceable words.

- She borrowed $50 from her friend-> She borrowed fifty dollars from her friend.
- He bought 12 sheep on 18 Aug 2016 -> He bought twelve sheep on eighteenth August Two thousand sixteen.

The next stage of this speech synthesis system is linguistic and prosodic processing. Pronounceable words are inputs to this stage. The system needs phones, durations, and tune to turn these pronounceable words into segments with prosody (F0 contour). In this stage the pronounceable words are mapped into phone segments with prosodic features. Dictionary lexicons are used to find the appropriate phonetic symbols for input words. For words that are not present in the dictionary, the Lexicon list uses letter-to-sound rules to extract phonetic symbols. There are two methods for generating these rules. One is handwritten, while the other uses the Classification and Regression Tree (CART) algorithm to generate models.

Prosody refers to the combination of intonation, durations, and post-lexical norms. In order to produce natural speech as an output, prosody is crucial. Accent and F0 contour, which are drawn from current voice models, are all that constitutes intonation. The energy of the output speech is determined by these intonational parameters. Duration denotes the length of the phone in the phrase.

The festival speech synthesis system's final and most crucial component is waveform synthesis. From the preceding block and existing voice models, this receives phone information and prosody for synthesis. It will generate synthetic speech as an output by combining all of these factors. The waveform synthesizer accesses relevant and required information from voice models differently depending on the voice models and produces synthetic speech.

E. *DeepSpeech*

The Speech Recognition stage is implemented using DeepSpeech Network. The goal of this model is to create an open, and simple speech recognition model. Simple since the model shouldn't operate on server-class hardware. The main component of the network architecture is RNN (Recurrent Neural Network) trained to accept speech spectrograms (MFCC) and produce English text transcription [9].

The Complete architecture of the Network is illustrated in the figure below. The number of nodes in each layer can be customized while training the network using the variables called geometric constraints [9].

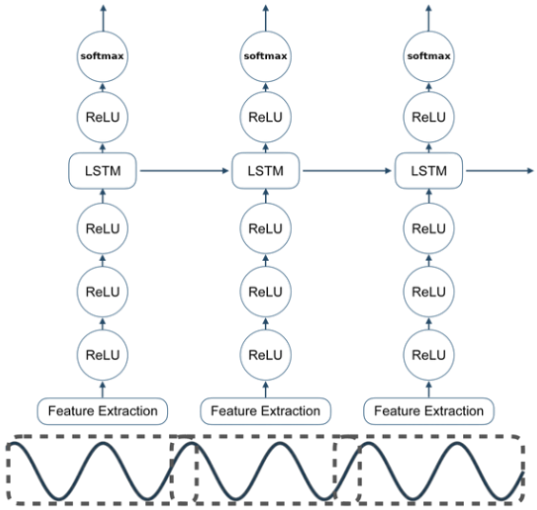


Fig. 8. Architecture of DeepSpeech network

Let a training set be used to sample an utterance x and a label y

$$S = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots\} \quad (1)$$

Each utterance, $x^{(i)}$, is a time-series of length $T^{(i)}$, with each time-period containing an audio features vector, $x_t^{(i)}$

where t=1,….,$T^{(i)}$. MFCC's are fed into the network as features; so $x_{t,n}^{(i)}$ denote the n^{th} MFCC feature in the audio frame at time t. The goal of the RNN is to convert an input sequence x into a sequence of character probabilities for the transcription y, with $\hat{y}^t = \mathbb{P}\big((c_t|x)\big)$, where for English $c_t \in$ *{a, b, c,...,z, space, apostrophe, blank}* [9].

V. IMPLEMENTATION

The block diagrams of the systems given below explain the overall implementation of the paper.

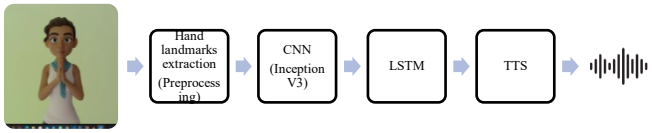


Fig. 9. Block diagram of ISL recognition system.

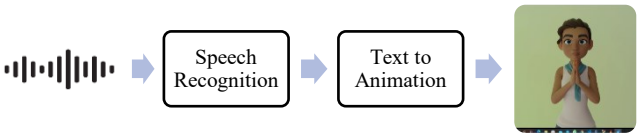


Fig. 10. Block diagram of Speech to ISL translation system.

In design of the Indian Sign Language to Speech conversion system, the gesture videos are converted to frames the Mediapipe is applied to it and the landmarks of hands are extracted. The frames are trained on Inception v3 to extract spatial features. The output of CNN is trained on LSTM to extract temporal features. The LSTM then outputs text. A text to speech synthesis system is trained using speech dataset tagged with corresponding text. The text output of LSTM is passed to the text to speech synthesis system, which converts input text to speech.

In design of the Speech to Indian Sign Language translation system, the DeepSpeech API (a Speech to Text model or ASR block) is integrated with the Text to Animation block (TTA). In this process, the real-time speech of the user through the microphone is given to DeepSpeech API (Speech to Text) model to generate the text of spoken words. The words that are obtained from speech to text model are split into multiple phrases based on the availability of the phrases in the dataset. If the spoken words like names or numbers are not present in the dataset, then the words are sequentially played character-by-character (Fingerspelling). The video gestures for corresponding phrases are obtained from the dataset and are aggregated into a single video file. The aggregated videos are displayed on the screen.

VI. RESULTS

The design of the ISL to text conversion system is done by tuning the hyperparameters, i.e., epochs and batch size. By changing the number of epochs and batch size, training has been done and the accuracies are noted down in below table.

Table IV ISL RECOGNITION SYSTEM TRAINING RESULTS

CNN			LSTM		
Batch Size	Training Steps	Accuracy	Batch Size	Epochs	Accuracy
200	4000	64.6%	32	75	73.13%
			32	100	82.34%
500	4000	64.7%	32	75	78.89%
			32	100	86.55%
1000	4000	64.9%	32	75	78.60%
			32	100	90.13%
1000	7000	70.5%	32	75	75.65%
			32	100	87.65%
1000	10000	79.8%	32	75	96.26%
			32	100	94.26%

The Quality of output of text to speech synthesized system, i.e., Synthesized speech is measured using Mel-Cepstral Distrotion (MCD). MCD objective measure is used for testing the generated TTS voice model and the values are 5.31 for around 3.5hrs data and 5.24 for around 4hrs of data.

The Speech to ISL system has a GUI design with 2 buttons and a text box as shown in figure 13. The mic button enables the mic after which the user spoken words are recorded till a pause and the text generated by DeepSpeech API is displayed in the textbox. The Video of the resulted text is played when the play button is pressed

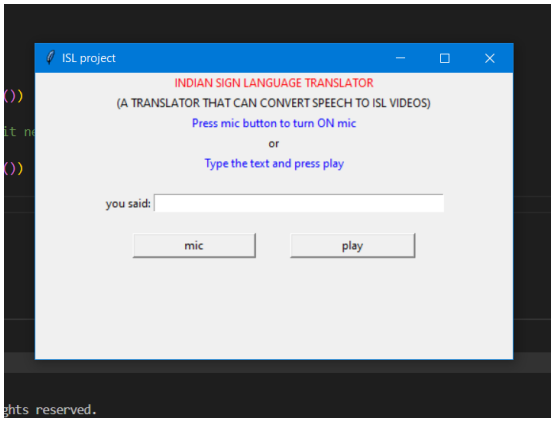


Fig. 11. GUI design of Translator

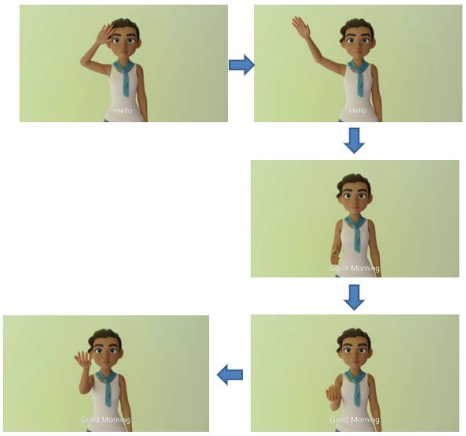


Fig. 12. Sequence of images of played video for input "Hello good morning."

VII. CONCLUSION

This paper's major goals are to illustrate the value of ISL translation systems and to create a viable solution that can translate speech to ISL and ISL to speech. While ISL has received less attention in this area, significant study has been conducted in other widely used sign languages including American Sign Language and British Sign Language. The grammar of sign languages like BSL and ASL allows rule-based systems, syntax analysis, and semantic interpretation to be done in order to get the appropriate translation. Contrarily, there are no precise grammar rules in Indian Sign Language, hence there are no standards to compare the English text with, which makes syntax and semantic analysis challenging.

The designed model is the basic version to convert Indian Sign Language to Speech. But more features are yet to be added to use it in the real-life application. The trained network converts the video gestures to text and the text to speech synthesizer converts the text to speech accurately and on the other hand speech is converted to text and then the recognized text is translated to Sign Language gestures accurately. As the trained model has few limitations to overcome, it can be further improved and can be integrated on the hardware and can be used in real-time for communication between a hearing/speech impaired person and the one who cannot apprehend Indian Sign Language gesture. As there are many similarities among few gestures, for example: cancer and virus gestures. It becomes difficult for our created model to differentiate between them.

Here we used an open-source Speech Recognition system (DeepSpeech), which has a WER (Word Error Rate) 7.06% on Libri speech clean test corpus. We can also use a different API for this. DeepSpeech is used because it is an open-source project.

VIII. FUTURE SCOPE

The network has to be trained intensively using dataset which contains more weightage for gestures that look more similar. Then, the model will be able to convert Indian Sign Language gesture to Speech accurately. That type of network which can identify all types of ISL gestures will really be a boon to the society.

There are more than 10000 gestures in ISL. In this paper, we were able to create dataset for 76 gestures and animations dataset for 100 gestures which can help interaction in hospital in a Doctor-Patient scenario. We want to include as many gestures as we can in our system, and animations can be made for a wide variety of scenarios. In order to make our product genuinely comprehensive and useful, our gesture library needs to be updated consistently. All around India, various languages are spoken. The fact that each state and province have its unique language makes this translation more challenging than before. Provisions for all the different languages must be developed using machine learning and other techniques. Multi Lingual support is essential for any system to be successful in India. There isn't a real, complete ISL database out there.

This paper is a sub part of the project entitled “Design and Implementation of Sign Language Translator (SignXter) using Hierarchical Long Short Term Memory (LSTM)” funded by AICTE under Research Promotion Scheme (RPS).

REFERENCES

[1] Office of the Registrar General & Census Commissioner, Sample registration survey of India (SRS: 2018). Ministry of Home Affairs, Government of India

[2] Goyal, L., & Goyal, V. (2016). Development of Indian Sign Language Dictionary using Synthetic Animations. Indian Journal of Science and Technology, 9(32).

[3] Kendon, A. Gesture: Visible Action as Utterance; Cambridge University Press: Cambridge, UK, 2004.

[4] Community, B.O., 2018. Blender - a 3D modelling and rendering package, Stichting Blender Foundation, Amsterdam. Available at: http://www.blender.org.

[5] Muthu Mariappan H, Dr Gomathi V, “Real Time Recognition of Indian Sign Language”, International Conference on Computational Intelligence in Data Science (ICCIDS) IEEE 2019.

[6] Nishi Intwala, Arkav Banerjee, Meenakshi and Nikhil Gala, “Indian Sign Language converter using Convolutional Neural Networks”, International Conference for Convergence in Technology (I2CT) IEEE 2019.

[7] Pankaj Sonawane, Karan Shah, Parth Patel, Shikhar Shah, and Jay Shah, “Speech to Indian Sign Language (ISL) translation system” 2021 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS) -IEEE

[8] Mateen Ahmed, Mujtaba Idrees, Zain ul Abideen, Rafia Mumtaz and Sana Khalique, “Deaf talk using 3D animated Sign Language: A Sign Language Interpreter using Microsoft's Kinect v2” 2016 SAI Computing Conference (SAI) – IEEE

[9] Awni Hannun, Carl Case, Jared Casper, Bryan Catanzaro, Greg Diamos, Erich Elsen, Ryan Prenger, Sanjeev Satheesh, Shubho Sengupta, Adam Coates, Andrew Y. Ng, “Deep Speech: scaling up end-to-end speech recognition” arXiv:1412.5567

[10] S. F. Ali, Gouri Sankar Mishra, A. K. Sahoo, ―Domain Bounded English to Indian Sign Language Translation Model‖, International Journal of Computer Science and Informatics, Volume 4, Issue 1, Article 6 Community, Def-ISL – a learning platform for Indian Sign Language. Available at: https://www.def.org.in/

[11] Nishi Intwala, Arkav Banerjee, Meenakshi and Nikhil Gala, “Indian Sign Language converter using Convolutional Neural Networks”, International Conference for Convergence in Technology (I2CT) IEEE 2019.

[12] Jayshree Maloo, Aishwarya Ramesh, Kohsheen Tiku, Indra R, “Real Time Conversion of Sign Language to Text and Speech”, International Conference on Inventive Research in Computing Applications IEEE 2020.

[13] Kusurnika Dutta, Satheesh Kumar Raju K, Anil Kumar G, Sunny Arokia, “Double Handed Indian Sign Language to Speech and Text”, International Conference on Image Information Processing IEEE 2015.

[14] Sarfaraz Masood, Adhyan Srivastava, Harish Chandra Thuwal and Musheer Ahmad, “Real-Time Sign Language Gesture (Word) Recognition from Video Sequences Using CNN and RNN”, Intelligent Engineering Informatics, Advances in Intelligent Systems and Computing, Springer Nature Singapore Pte Ltd. 2018.

[15] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, Matthias Grundmann, “MediaPipe Hands: On-device Real-time Hand Tracking”, Google Research 1600 Amphitheatre Pkwy, Mountain View, CA 94043, USA.

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, “SSD: Single Shot MultiBox Detector”, arXiv:1512.02325, Computer Vision and Pattern Recognition 2015.

[17] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna, “Rethinking the Inception Architecture for Computer Vision”, arXiv:1512.00567v3, Computer Vision and Pattern Recognition 2015.