

# **SPEECH TO INDIAN SIGN LANGUAGE**

<sup>1</sup>V. RAVI KUMAR, <sup>2</sup>SAKETH REDDY, <sup>3</sup>M. AJAYKUMAR, <sup>4</sup>P. INDHU

*Dept. of Information Technology, TKR College of Engineering and Technology, Hyderabad, Telangana*

[ravivannoj@tkrcet.com](mailto:ravivannoj@tkrcet.com), [sakethreddyrebal@gmail.com](mailto:sakethreddyrebal@gmail.com), [ajaychinu02@gmail.com](mailto:ajaychinu02@gmail.com), [Indhu0501@gmail.com](mailto:Indhu0501@gmail.com).

## **ABSTRACT**

*People interact with each other mostly through speech. Birth abnormalities, accidents, and oral disorders have all contributed to the significant rise in the number of deaf and dumb people in recent years. Because deaf and dumb persons are unable to interact with others, they must rely on visual communication. Sign language is a kind of communication in which meaning is conveyed through visually transmitted sign patterns. It is mostly used by persons who are deaf or hard of hearing to communicate fluently. This initiative is designed to assist these persons with special needs in participating in society on an equal footing. The Speech to Indian Sign Language system is built as a website where the user inputs audio (or) text into the system. The system then applies natural language processing to the input and creates a video with the relevant Indian Sign Language (ISL). The video is created by comparing the preprocessed text to the Indian Sign Language (ISL) Dataset for the presentation of sign symbols. The user is shown the video as well as the Indian Sign Language (ISL) sentence. We suggest a method to assist those who are unable to communicate with ordinary people due to a lack of advanced gadgets such as power, data gloves, and colored finger caps, among other things. This technique allows hearing individuals to communicate with deaf people simply.*

## **1. INTRODUCTION**

Deaf people use sign language to communicate, due to this it is difficult for people to communicate with them therefore in order to eradicate this problem the application “Speech to Indian sign language” is created. It is a real-time system that takes both speech and text as input and produces respective signs as output to convey information to deaf people. The Speech to Indian Sign Language system is implemented as a website where the user gives the audio input (or) text input to the system. The System then performs NLP to

the input and generates a video containing the corresponding Indian Sign Language (ISL). The video is constructed by matching the pre-processed text with Indian Sign Language (ISL) Dataset for the sign symbol to be displayed. The video as well as the Indian Sign Language (ISL) sentence is shown to the user. Thus, the system is used to overcome the hurdles faced by normal people to share their thoughts with hearing-impaired people and it will be an ear for the hearing-impaired.

## **2. LITERATURE REVIEW**

“Railway Stations Announcement System for

Deaf” This paper was written by Rakesh Kumar, Vishal Goyal, and Lalit Goyal published by International Conference on Natural Language Processing: System Demonstrations in 2020. Hearing-impaired persons cannot hear announcements given at railway stations, such as which train is traveling to which destination. This Machine Translation system receives announcements in the form of English text as input and creates synthetic animations in Indian Sign Language (ISL) as output. The technology, created for a particular area of railway reservations, transforms Hindi strings entered by reservation clerks into Indian Sign Language (ISL) gloss strings that are then converted using HamNoSys to the animated human avatar.

“Audio to Sign Language Translation for Deaf People” This paper was written by Ankita Harkude, Sarika Namade, Shefali Patil, and Anita Morey published by International Journal of Engineering and Innovative Technology (IJEIT) in 2020. Sign language is the mother tongue of deaf individuals, according to this research. This comprises hand gestures, arm or body movements, and facial expressions. This project makes use of Indian Sign Language. This technology enables the deaf population to participate in activities that typical people perform, such as everyday interaction and

information access. This application accepts voice as input, translates it to text, and then shows graphics in Indian Sign Language. The system's front end is built with EasyGui, while speech input through the microphone is handled with the PyAudio module. The Google Speech API is used to recognize the speech NLP is then used to pre-process the text (Natural Language Processing)

“Speech To Sign Language Translator For Hearing Impaired” This paper was written by Ezhumalai P, Raj Kumar M, Rahul A S, Vimalanathan V, and Yuvaraj published by Turkish Journal of Computer and Mathematics Education in 2021. The Speech to Sign Language translator is used in this study to break down communication barriers between normal and hearing-impaired persons. This translator makes it easier and faster for hearing-impaired persons to communicate their thoughts to regular people. Natural Language Processing techniques are used by the translator to transform voice or text to Sign Language. This technology is intended to convert each word received as input into sign language. The words in this project are translated using Indian Sign Language.

### **3. METHODOLOGY**

#### ***Data Collection***

We will be using the <http://www.indiansi-gnlanguage.org/> to download the video

clips of each and every word. We will manually label each of the video and remove videos that we will find irrelevant. We will like to maintain an unfiltered input that covers a wide range of words.

***Algorithm Design***

The system consists of 5 modules:

- English parser for parsing the English text
- Sentences reordering module based on ISL grammar rules
- Eliminator for eliminating stop words
- Stemming for getting the root words of each word and synonym replacement for words not in dictionary.
- Video conversion module.

The input to the system is a written English text which is parsed to create a phrase structure based on its grammar representation. Then reordering is done to meet ISL grammar needs since, English text follows Subject-Verb-Object structure whereas ISL follows Subject-Object-Verb structure along with variation of negative and interrogative sentences. After which unwanted words are removed, as ISL will only those words which have meanings and all helping words like linking verbs, articles etc are not used. The output of which is sent to lemmatization module which reduces each of the words to its root

form. The words not present in dictionary are replaced with their synonyms.

**Solution Structure**

***i. Parsing of the Input English Text*** To carry out rule based conversion of one language to another, grammatical structure of both the source and target language must be known. Parsing is the answer to acquiring this grammatical structure. Stanford parser is capable to produce three different outputs, part-of-speech tagged text, context free grammar representation of phrase structure and type dependency representation. The parser uses Penn tree tags for parsing the English sentence.

***ii. Grammar rules for conversion from English to ISL***

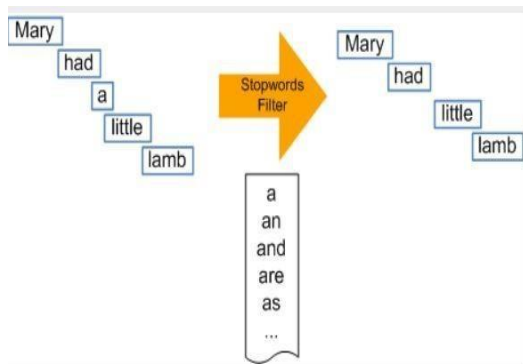
Since both spoken language and sign language have different grammar rules. The complexity to translate them is increased many folds. Comparison between ISL and English grammar is listed below:

English Grammar	Indian Sign Language Grammar
English grammar is well structured and a lot of research work has been carried out to define the rules for it. English grammar follows the subject-verb-object order.	ISL is invented by deaf and a little work has been done to study the grammar of this language. The structure of sentences of ISL follows the subject-object-verb order[13].
English language uses various forms of verbs and adjectives depending upon the type of the sentence. Also, a lot of inflections of the words are used in English sentences.	ISL does not use any inflections ( gerund, suffixes, or other forms ), it uses the root form of the word.

English language has much larger dictionary	Indian sign language has a very limited dictionary, approximately 1800 words[10].
Question word in interrogative sentences is at the start in English	In Indian sign language, the question word is always sentence final
A lot of helping verbs, articles, and conjunctions are used in the sentences of English	In Indian sign language, no conjunctions, articles or linking verbs are used

**iii. Elimination of Stop Words**

Since ISL deals with words associated with some meaning, unwanted words are removed these include various parts of speech such as TO, POS(possessive ending), MD(Modals), FW(Foreign word), CC(coordinating conjunction), some DT(determiners like a, an, the), JJR, JJS(adjectives, comparative and superlative), NNS, NNPS(nouns plural, proper plural), RP(particles), SYM(symbols), Interjections, non-root verbs

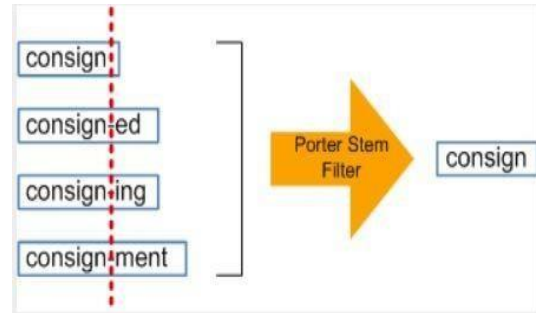


**iv. Lemmatization and Synonym replacement**

Indian sign language uses root

words in their sentences. So we convert them to root form using Porter Stemmer rules. Along with this each word is checked in bilingual dictionary, if word does not exist, it is tagged to its synonym containing the same part of speech.

**Stemming of Words**



**v. Video Conversion Stage**

After the completion of above stages we get the ISL transformed text, the program will then find matches from the dataset available for each of the word. This will be based on the basic string matching algorithm between the processed input text and labels of videos. Finally a display of set of videos as a sequence one after the other can be seen on the screen.

**Output Generation**

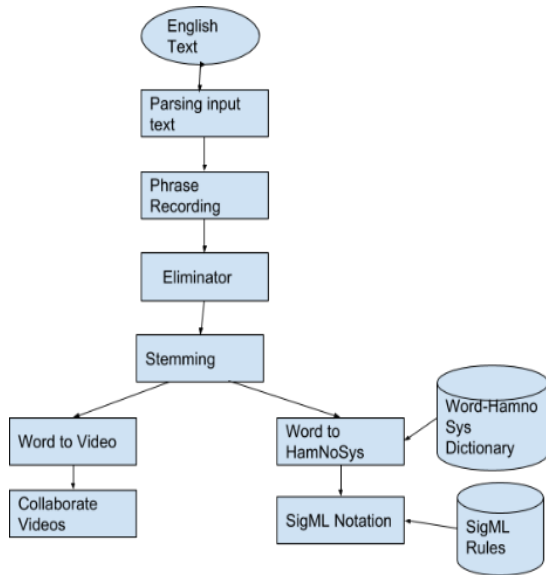
The output of this module will be a movie clip of ISL translated words. The database will be having video for each and every separate words and the resultant video will be a merged video of such words.

**4. IMPLEMENTATION**

**a. Design Document and Flowchart**

Our first terminology, BIG DATA, fits here.

Big data is nothing but any data which is too big/complex to handle. Big data does not necessarily mean data that is large in science. Big data is characterized by 4 different properties and if your data exhibits this property then it is qualified to be called Big data.



These properties are defined by 4 V's.

- Volume: Data in terabytes
- Velocity: Streaming data with high throughput
- Variety: Structured, semi-structured, and unstructured
- Veracity: quality of the data that is being analyzed

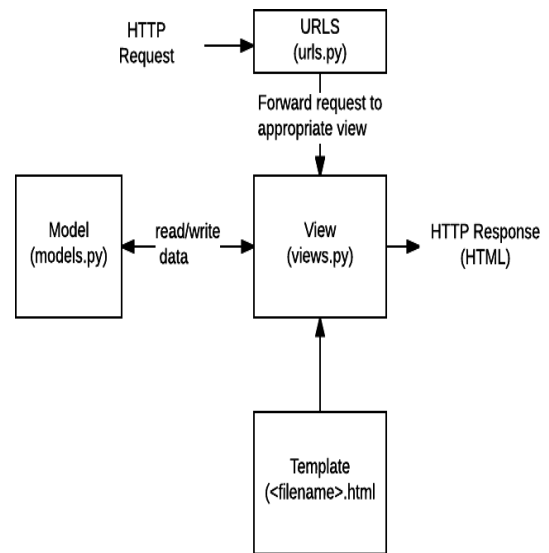
Django is an open-source framework for backend web applications based on Python — one of the top web development languages. Its main goals are simplicity, flexibility, reliability, and scalability.

Django has its own naming system for all

functions and components (e.g., HTTP responses are called “views”). It also has an admin panel, which is deemed easier to work with than in Lavarel or Yii, and other technical features, including:

- Simple syntax;
- Its own web server;
- MVC (Model-View-Controller) core architecture;
- “Batteries included” (comes with all the essentials needed to solve solving common cases);
- An ORM (Object Relational Mapper);
- HTTP libraries;
- Middleware support; and
- A Python unit test framework.

Django web applications typically group the code that handles each of these steps into separate files:



- *URLs*: While it is possible to process requests from every single URL via a single

function, it is much more maintainable to write a separate view function to handle each resource. A URL mapper is used to redirect HTTP requests to the appropriate view based on the request URL. The URL mapper can also match particular patterns of strings or digits that appear in an URL, and pass these to a view function as data.

- *View*: A view is a request handler function, which receives HTTP requests and returns HTTP responses. Views access the data needed to satisfy requests via models, and delegate the formatting of the response to templates.
- *Models*: Models are Python objects that define the structure of an application's data, and provide mechanisms to manage (add, modify, delete) and query records in the database.
- *Templates*: A template is a text file defining the structure or layout of a file (such as an HTML page), with placeholders used to represent actual content. A view can dynamically create an HTML page using an HTML template, populating it with data from a model. A template can be used to define the structure of any type of file; it doesn't have to be HTML!

### ***Set up a database for a Django project***

Django in its 'out-of-the-box' state is set up to communicate with SQLite -- a lightweight relational database included with the Python

distribution. So by default, Django automatically creates a SQLite database for your project.

In addition to SQLite, Django officially supports (i.e. included in Django itself) three other popular relational databases that include: PostgreSQL, MySQL and Oracle. And unofficially (i.e. with third party packages) Django supports connectivity to other relational databases that include: SAP (Sybase) SQL Anywhere, IBM DB2 and Firebird, as well as the ADO (ActiveX Data Objects) and ODBC (Open Database Connectivity) interfaces, the last two of which are standard for connecting to Microsoft SQL Server and the latter is supported by most relational database brands. The Django configuration to connect to a database is done inside the settings.py file of a Django project in the DATABASES variable. If you open the settings.py file of a Django project you'll notice the DATABASES variable has a default Python dictionary with the values.

```
# Build paths inside the project like this:
os.path.join(BASE_DIR, ...)

import os
BASE_DIR = os.path.dirname(os.path.abspath(__file__))

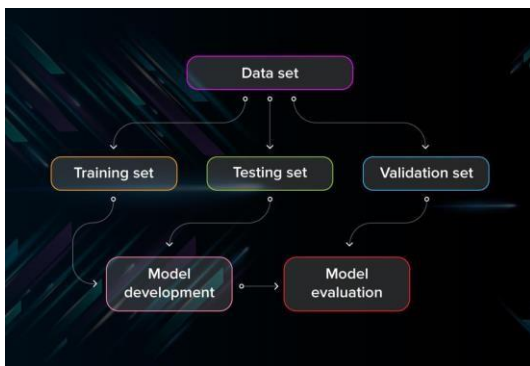
DATABASES = {
    'default': {
```

```
'ENGINE': 'django.db.backends.sqlite3',  
'NAME': os.path.join(BASE_DIR,  
'db.sqlite3'),  
}  
}
```

### How to write model tests:

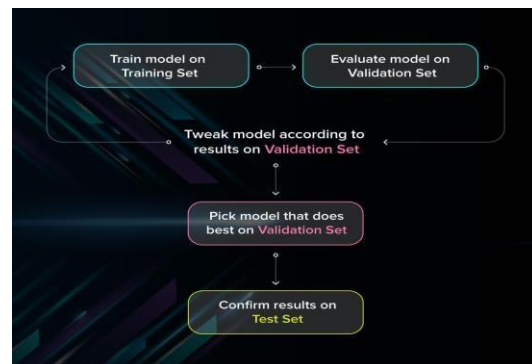
So, to write model tests, we need to cover several issues:

- Check the general logic of the model (not possible in the case of deep neural networks so go to the next step if working with a DL model).
- Control the model performance by manual testing for a random couple of data points.
- Evaluate the accuracy of the ML model.
- Make sure that the achieved loss is acceptable for your task.
- If you get reasonable results, jump to unit tests to check the model performance on the real data.



First of all, you split the database into three non-overlapping sets. You use a training set to train the model. Then, to evaluate the performance of the model, you use two sets of data:

- **Validation set.** Having only a training set and a testing set is not enough if you do many rounds of hyperparameter-tuning (which is always). And that can result in overfitting. To avoid that, you can select a small validation data set to evaluate a model. Only after you get maximum accuracy on the validation set, you make the testing set come into the game.
- **Test set (or holdout set).** Your model might fit the training dataset perfectly well. But where are the guarantees that it will do equally well in real-life? In order to assure that, you select samples for a testing set from your training set examples that the machine hasn't seen before. It is important to remain unbiased during selection and draw samples at random. Also, you should not use the same set many times to avoid training on your test data. Your test set should be large enough to provide statistically meaningful results and be representative of the data set as a whole.



But just as test sets, validation sets “wear out” when used repeatedly. The more times you use the same data to make decisions about hyperparameter settings or other model

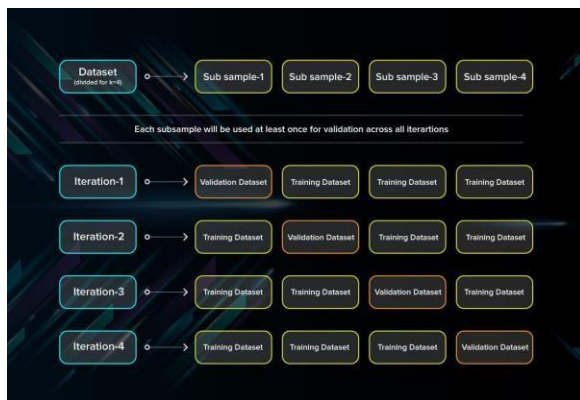
improvements, the less confident you are that the model will generalize well on new, unseen data. So it is a good idea to collect more data to 'freshen up' the test set and validation set.

### Cross-validation

Cross-validation is a model evaluation technique that can be performed even on a limited dataset. The training set is divided into small subsets, and the model is trained and validated on each of these samples.

### K-fold cross-validation

The most common cross-validation method is called k-fold cross-validation. To use it, you need to divide the dataset into  $k$  subsets (also called folds) and use them  $k$  times. For example, by breaking the dataset into 10 subsets, you will perform a 10-fold cross-validation. Each subset must be used as the validation set at least once.



This method is useful to test the skill of the machine learning model on unseen data. It is so popular because it is simple to apply, works well even with relatively small

datasets, and the results you get are generally quite accurate. If you want to learn more about how to cross-validate the model.

### Leave-one-out cross-validation

In this method, we train the model on all the data samples in the set except for one data point that is used to test the model. By repeating this process iteratively, each time leaving a different data point as a testing set, you get to test the performance for all the data.

The benefit of the method is low bias since all the data points are used. However, it also leads to higher variation in testing because we are testing the model against just one data point each time.

Cross-validation provides for more efficient use of the data and helps to better assess the accuracy of the model.

### Hyper parameter Tuning

Ill-adjusted hyperparameters can be the reason for the poor performance of the model. Here are the metrics you should usually check:

- **Learning rate.** Usually, ML libraries pre-set a learning rate, for example, in TensorFlow it is 0.05. However, it might not be the best learning rate for your model. So the best option is to set it manually between 0.0001 and 1.0 and play with it, seeing what gives you the best loss without taking hours to train.

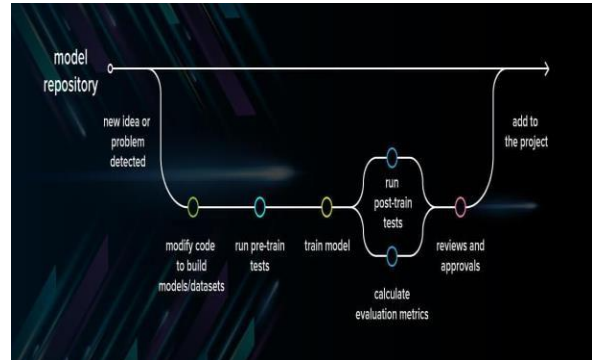


- **Regularization.** You should conduct regularization only after you have made sure that the model can make predictions on the training data without regularization. L1 regularization is useful if you need to reduce your model's size. Apply L2 regularization if you prefer increased model stability. And, in the case of neural networks, work with dropout regularization.
- **Batch size.** Models trained on smaller batches usually generalize better. A batch should usually contain 10-1000 samples where the minimal size depends on your model.
- **Depth of layers.** The depth describes the number of layers in a neural network: the more layers it has, the deeper it is. Start from 1 layer and gradually increase the number if you feel like the model should be deeper to solve your problem. This approach helps to not overcomplicate the model from the very beginning.

### Model development pipeline

The 'agenda' of your model development should include evaluation, pre-train tests, and post-train tests. These stages should be organized in one pipeline that looks something like in the following figure. Performing ML tests is necessary if you care about the quality of the model. ML testing has a couple of peculiarities: it demands that you test the quality of data, not just the model, and go through a couple of iterations

adjusting the hyperparameters to get the best results.



However, if you perform all the necessary procedures, you can be sure of its performance.

## 5.DATA ANALYSIS AND DISCUSSION

### a. Output Generation

For a given english text the system aims at generating its equivalent sign language depiction. Our system generates these outputs in the following two ways:

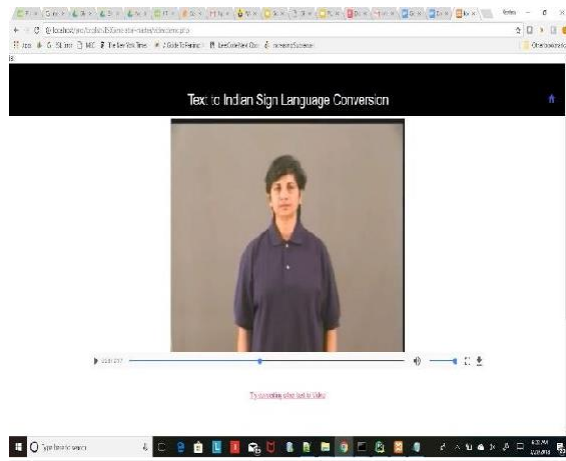
- i. Video generation - Output from the ISL conversion phase of input sentence is passed to video generation phase wherein for each of the words in the sentence are looked up in the database for its corresponding video file and then these files are all concatenated to to produce a more structured, informative and easy to understand visual depiction of Indian Sign Language.
- ii. Synthetic Animation generation - In this approach the ISL converted text is checked for its corresponding SigML file wherein these files are generated by a

process called Hamnosys to SigML conversion to generate a markup language that is compatible with JASigning tool to generate equivalent synthetic animations.

Consider the following input sentence to which the generated ISL, video output and animation output are as follows:

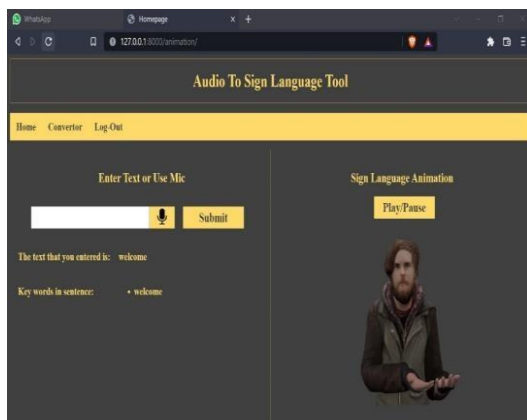
**English input text:** *I am going to the University.*

**ISL parsed text:** *I University going*



**Video Generated Output:**

**Animation Generated Output:**



### **b. Output Analysis**

The given input text undergoes parsing

using the StanfordParser to identify the subject, verb and object in the given sentences and which according to Indian Sign language grammar requires its verb and object positions to be swapped. This process fairly returns proper results and after this we require the system to lemmatize each of the words as sign language does not follow tense, however lemmatization follows a corpus and we chose wordnet corpus which not necessarily stem each of the verb forms, due to which there is a requirement to modify video and sigml files to support this issue. The generated output is then checked for stopwords and they are eliminated if present. After which the ISL sentence is then passed through two methods to generate sign language

Each of these methods were being analysed and we came up with the following analysis for each of them:

#### **i. Video**

1. The process of finding the relevant videos and creating a collaboration of these videos is a time taking task.
2. The video output couldn't ensure consistency due to the lack of research going on this field.
3. All words not necessarily find their equivalent word file, thus such words are skipped.

## **ii. Synthetic Animation**

1. The process of mapping the sign notations to generate animation is comparatively less time consuming
2. This ensures consistency as the entire sentence is enacted by the same character throughout.
3. However, these animations do not provide a realistic feel as is provided by videos.
4. Finger spelling are used for those words which are not present in the database.

## **CONCLUSION**

The Speech to Indian Sign Language system is implemented as a website where the user gives the audio input (or) text input to the system. JavaScript Web Speech API is used for speech recognition. The System then perform NLP i.e. tokenization, lemmatization and filtering stop words to the input and generate a video containing the corresponding Indian Sign Language (ISL). The video is constructed by making use of Indian Sign Language (ISL) Dataset. The video as well as the Indian Sign Language (ISL) sentence is shown to the user. The output is given as video instead of images, gifs to generate a realistic and vibrant appeal. This Project presents an interactive system for hearing impaired people for impactful

communication and reduce the gap between the normal speaking people and the deaf people. This system can be used at different areas such as hospitals, public meetings etc., in the place of an interpreter, to convey message to the deaf people. Additionally this system can also be implemented as E-Learning tool to teach and learn Sign Language.

## **REFERENCES**

1. *Ankita Harkude, Sarika Namade, Shefali Patil and Anita Morey (2020) "Audio To Sign Language Translation For Deaf People"*
2. *Ezhumalai P, Raj Kumar M, Rahul A S, Vimalanathan V and Yuvaraj A (2021) "Speech To Sign Language Translator for Hearing Impaired"*
3. *Rakesh Kumar, Vishal Goyal, and Lalit Goyal (2020) "Railway Stations Announcement System for Deaf"*