KIET
GROUP OF INSTITUTIONS
Connecting Life with Learning

A Project Report

on

Sign-O-Voice

submitted as partial fulfillment for the award of

BACHELOR OF TECHNOLOGY DEGREE

SESSION 2024-25

in

Computer Science and Engineering (AI & ML)

By

Diya Bhatia (2100291530020)

Prashant Kumar Singh (2100291530041)

Shubham Bhatt (2100291530053)

Vanshika Goyal (2100291530069)

Under the supervision of

Mr. Rajeev Kumar Singh

KIET Group of Institutions, Ghaziabad

Affiliated to

Dr. A.P.J. Abdul Kalam Technical University, Lucknow

(Formerly UPTU)

DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which to a substantial extent has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Signature:

Name:

Roll No.:

Date:

CERTIFICATE

This is to certify that Project Report entitled “Sign-O-Voice” which is submitted by Student name in partial fulfillment of the requirement for the award of degree B. Tech. in Department of CSE(AIML) of Dr. A.P.J. Abdul Kalam Technical University, Lucknow is a record of the candidates own work carried out by them under my supervision. The matter embodied in this report is original and has not been submitted for the award of any other degree.

Mr. Rajeev Kumar Singh **Dr. Rekha Kashyap**

(Assistant Professor) **(Head of Department)**

Date:

ACKNOWLEDGEMENT

It gives us a great sense of pleasure to present the report of the B. Tech Project undertaken during B. Tech. Final Year. We owe special debt of gratitude to Mr. Rajeev Kumar Singh, Department of CSE(AIML), KIET, Ghaziabad, for his constant support and guidance throughout the course of our work. His sincerity, thoroughness and perseverance have been a constant source of inspiration for us. It is only his cognizant efforts that our endeavors have seen light of the day.

We also take the opportunity to acknowledge the contribution of Dr. Rekha Kashyap, Head of the Department of Computer Science & Engineering, KIET, Ghaziabad, for her full support and assistance during the development of the project. We also do not like to miss the opportunity to acknowledge the contribution of all the faculty members of the department for their kind assistance and cooperation during the development of our project.

We also do not like to miss the opportunity to acknowledge the contribution of all faculty members, especially faculty/industry person/any person, of the department for their kind assistance and cooperation during the development of our project. Last but not the least, we acknowledge our friends for their contribution in the completion of the project.

Date:
Signature:
Name :
Roll No.:

ABSTRACT

Sign language is an essential tool of communication for the Deaf and Hard of Hearing (DHH) community. However, a significant linguistic barrier exists within our society. Existing solutions excel at character or letter level translations, but research for conversational-level translation systems has been insufficient. Our report presents a system for seamless bidirectional translation between sign language and English. We leveraged key-point extraction, Long Short-Term Memory networks (LSTM), Natural Language Processing (NLP), and Generative AI for the execution. We focused on Indian Sign Language (ISL) and employed MediaPipe for extracting Pose and Hand landmarks, and the data was trained on a Bidirectional-LSTM (Bi-LSTM) network to identify each sign. A Large Language Model (Llama 3) aided in converting raw sign inputs into structured English sentences. By reversing the process i.e. translating speech into sign language visualizations - our system offers a comprehensive solution. The results highlight the potential for real-time application, significantly improving accessibility for the DHH community and advancing automated sign language translation.

TABLE OF CONTENTS

Page No.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

S. No	Abbreviation	Definition
1	ASL	American Sign Language
2	CNN	Convolution Neural Network
3	DHH	Deaf and Hard of Hearing
4	ISL	Indian Sign Language
5	ISLRTC	Indian Sign Language Research and Training Center
6	LLM	Large Language Model
7	LSTM	Long Short Term Memory
8	NLP	Natural Language Processing
9	PCA	Principal Component Analysis
10	RNN	Recurrent Neural Network
11	SVM	Support Vector Machine

CHAPTER 1
INTRODUCTION

1.1 INTRODUCTION

Sign language has been a primary mode of communication for the Deaf and Hard of Hearing (DHH) community, facilitating effective interaction and information exchange. However, a large majority of the speaking and listening population cannot communicate through signs. The linguistic barrier between sign language users and those unfamiliar with it poses significant challenges. Thus, to create a medium between the two communities has been the aim for decades.

1.2 PROJECT DESCRIPTION

Our project addresses the need for a seamless translation system that converts sign language to spoken language and vice versa, leveraging key point extraction, Long Short-Term Memory (LSTM) networks, and Natural Language Processing (NLP) techniques as well as Generative AI.

Furthermore, this approach aims to create a bidirectional medium i.e. sign to speech as well as speech to sign. So, a reversal approach is used to allow us to generate sign visualizations from speech or text. By developing an automated and efficient sign-to-speech and speech-to-sign translation system, this study aims to bridge the communication gap and enhance accessibility for the DHH community.

CHAPTER 2

LITERATURE REVIEW

A plethora of research has been going on in this domain for decades. There are primarily two categories of approaches to classify the signs and translate them - Vision Based and Non-Vision Based. One of the earliest propositions for sign or gesture recognition can be traced back to 1995 when Sidney Fels and Geoffrey Hinton [1] wrote about Glove-TalkII.

More recent methods have leveraged Computer Vision alongside Machine Learning to produce significantly better results. For static signs various studies have shown excellent results. The study conducted by Tharwat et al. [2] in 2015 focused on SVM and K-NN. A similar use of the K-NN classifier along with PCA was demonstrated by Dewinta Aryanie and Yaya Heryadi [3] in 2015 for American Sign Language.

More works [4], [5], [6] have used CNN, SVM and LSTM for translating ASL and ISL with accuracies as high as 99%. Even though CNN performs highly accurate classification of static signs, the previous approaches have not been implemented for dynamic signs with movement to represent the sign (video data). Furthermore, they have discussed translation only on alphabets and numbers, not entire words or phrases. TABLE I contains major previous works and their contributions for translations of various sign languages.

Jayanthi P, Ponsy R K Sathia Bhama & B Madhubalasri [6] opted to implement video classification using CNN. The dataset contained complete words unlike [1]-[4] and had the same linguistic properties as spoken languages and were expressed either by hand movements or hand movements along with facial expressions.

TABLE I

PREVIOUS WORKS

S. No	Year	Author	Work	Result
1.	1995	Sidney Fels and Geoffrey Hinton [1]	Used Glove-TalkII(Cyberglove) embedded with 18 flex sensors. Generated real-time speech by using an adaptive interface with neural networks to map hand movements to control parameters of a speech synthesizer	Mean Square Error on testing data was 0.01
2.	2015	Tharwat, A., Gaber, T., Hassanien, A. E., Shahin, M. K., & Refaat, B. [2]	Focused on SVM and K-NN on Arabic Sign Language	Accuracy of 99%
3.	2015	Dewinta Aryanie and Yaya Heryadi [3]	Implemented K-NN classifier along with PCA over ASL (American sign language)	Accuracy of 99.8%
4.	2016	Garcia, B. and Viesca [4]	Implementation of CNN for ASL by fine-tuning a pre-trained GoogLeNet and Caffe. They were able to produce a robust model for letters a-e, and a modest one for letters a-k (excluding j)	Validation accuracy was of nearly 98% with five letters and 74% with ten letters
5.	2022	Shagun Katoch, Varsha Singh and Uma Shanker Tiwary [5]	Approach uses the Bag of Visual Words model (BOVW) to identify Indian sign language alphabets (A-Z) and digits (0–9) from a live video stream. SURF (Speeded Up Robust Features) were extracted from the images and the signs were mapped to their corresponding labels	Accuracy greater than 99%

6.	2023	Jayanthi P, Ponsy R K Sathia Bhama & B Madhubalasri [6]	Implemented video classification using CNN. They performed keyframe extraction and leveraged 3D ConvNet. Furthermore, Long Short-Term Memory (LSTM) was utilized for predicting the next word in a sequence of gestures representing sign language	Accuracy of 89.99%
7.	2019	Mittal, A., Kumar, P., Roy, P. P., Balasubramanian, R., & Chaudhuri, B. B. [7]	Offered a modified long short-term memory (LSTM) model for continuous sequences of gestures or continuous SLR that identifies a sequence of connected gestures	Accuracy of 72.3% on signed sentences and 89.5% on isolated sign words
8.	2022	Sundar B and Bagyammal [11]	Used google's mediapipe hand landmark detection and LSTM to recognize alphabets in American sign language that captured 21 key points	Accuracy of 99%
9.	2024	Unnathi, E., Sreeja, A. K., Teja, R., & Vinutha, L. V. [12]	Integrated real-time sign language detection with a smart glove for gesture-to-text and voice output. Hybrid CNN and RNN algorithm was used	Accuracy of 92%

CHAPTER 3

PROPOSED METHODOLOGY

3.1 Data Acquisition

The sign dataset thus created consists of 26 classes (27, if "Where" is also considered) and were made with the reference of Sign Dictionary released by Indian Sign Language Research and Training Centre (ISLRTC) [13], [14].

We ensured that compound signs were broken down into their constituent meaningful signs, e.g. "Where" is made up of 2 signs, "Place" and "What". Thus, separate classes were made for these signs as this provides us the flexibility for combining signs to form different phrases and sentences, which is an essential part of any natural language with its grammar and syntax.

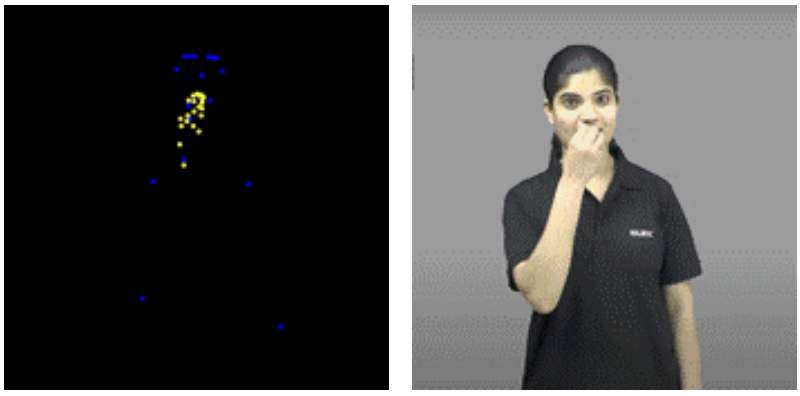

Fig. 1 "Eat" in ISL (a. Dataset Element, b. Reference Used [13])

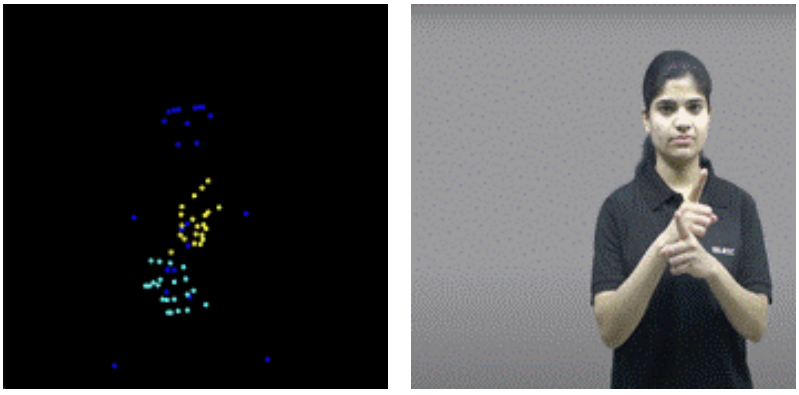

Fig. 2 "Teacher" in ISL (a. Dataset Element, b. Reference Used [14])

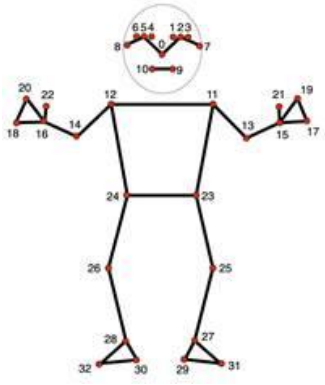

Fig. 3 Pose Landmarks [15]

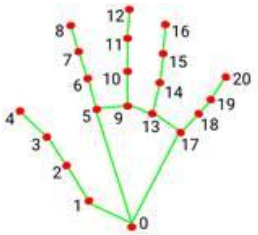

Fig. 4 Hand Landmarks [16]

Unlike the conventional approach of using video snapshots or entire videos for creating the dataset we used a different approach, leveraging MediaPipe [10], [11] to store the relevant information of the sign. MediaPipe is an open-source framework designed to create pipelines that enable computer vision inference on various types of sensory data, including video and audio. It was used to create the dataset ensuring different camera angles and covered variations of each sign. Each class contained 100 elements and within each recording, 30 frames were extracted and stored as a NumPy array. The NumPy arrays store the key-points extracted from each frame i.e. the Pose Contours and Hand Connections.

This approach reduced the size of the dataset significantly (~5% of a similar video dataset) as only relevant vector points are stored in the NumPy arrays.

TABLE II

VOCABULARY OF DATASET

Greetings	Hello	Thank You	
Affirmatives	Yes	No	
Nouns- People	Student	Teacher	
Nouns- Places	School	Home	Place
Nouns- Objects	Book	Pizza	
Nouns- Time	Morning	Yesterday	Tomorrow
Personal Pronouns	I/Me	You	
Demonstrative Pronouns	This		
Interrogative Pronouns	What	Where (Place +What)	
Adjectives	Hungry	Good	
Verbs	Eat	Read	Do
Verbs	Go	Want	
Adverb	Not		

3.2 Data Preprocessing

The collected key-points of the ISL were pre-processed before being fed into the neural network. The NumPy arrays of the Pose Contours were resized and the key-points below the shoulders were removed. This needed to be done because the lower body is not involved in sign language. This removed useless data and simplified the features of the sign. This further allowed the user to use the model while sitting or standing, ensuring flexibility in use.

Since the user could be at different positions in the camera frame, e.g. close to the camera, farther, to the left or to the right, the key-points needed to be normalized. Thus, using the “nose” as a central key-point, all the other key-points of the pose landmarks were normalized along the x-y plane. To normalize along the z- axis, the Euclidean Distance between the left

and right shoulder was used. To ensure that the relations between fingers and their positions is not lost, the hand landmarks were normalized with respect to the wrist.

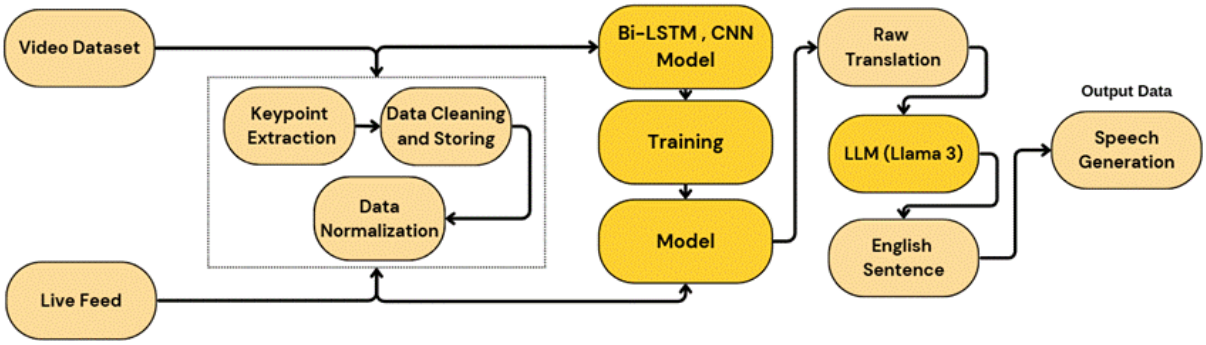

Fig. 5 Data Flow Diagram

3.3 Training

The model was created with TensorFlow's Sequential model. It was composed of a CNN input layer for feature extraction at spatial level. It was followed by 3 Bi-LSTM layers with 128, 128, 64 neurons respectively. After each Bi-LSTM layer, we have a Batch Normalization layer, as well as a Dropout layer (40% and 30% dropout rate) for regularization and to prevent overfitting.

A sign can have multiple variations, and minor errors in signing by the users is certainly expected during the real application of the code. To avoid that we added L2 regularization or ridge regularization to prevent the overfitting of weights by adding a small error to the loss, in order to make the model less sensitive to small changes. Unlike L1 regularization, it does not push the weights all the way to zero, which encourages the model to take in account all the features and miss none, even the one with the small contribution towards proper classification.

For compilation of the model, Adam optimizer was used for the categorical classification of the signs. The training ran for 100 epochs. The resultant training accuracy came out to be 99.94%. The testing accuracy came out to be 99.69%.

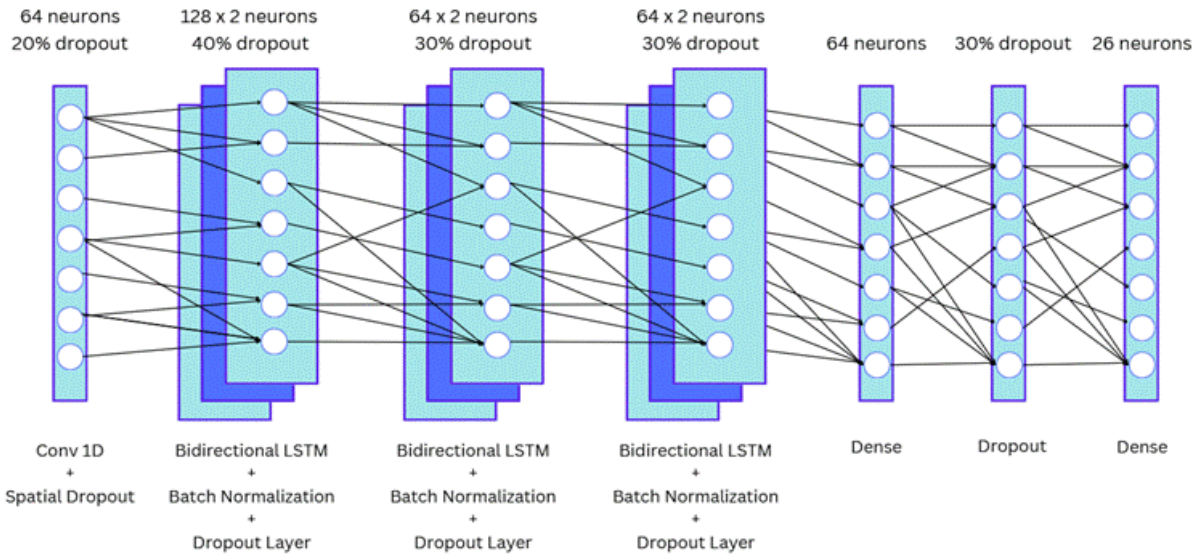

Fig. 6 Model Architecture Diagram

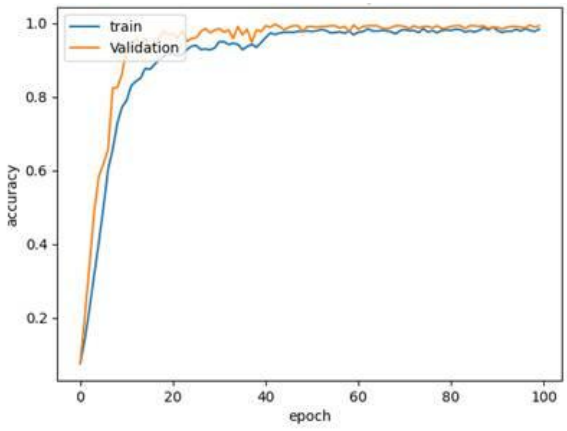

Fig. 7 Training and Validation Accuracy

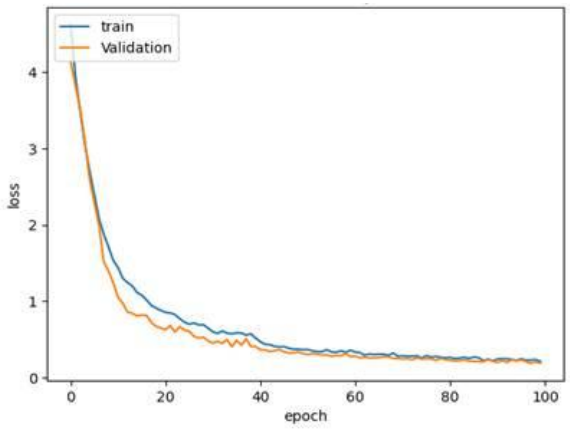

Fig. 8 Training and Validation Loss

```
model2.evaluate(X_test, Y_test, verbose=1)
```

```
21/21 [==============================] - 1s 29

[0.16593891382217407, 0.9969230890274048]
```

Fig. 9 Testing Loss and Accuracy

	precision	recall	f1-score	support
Book	0.96	1.00	0.98	23
Do	1.00	1.00	1.00	19
Eat	0.96	1.00	0.98	22
Go	1.00	1.00	1.00	18
Good	1.00	0.86	0.93	22
Hello	1.00	1.00	1.00	18
Home	1.00	1.00	1.00	20
Hungry	1.00	1.00	1.00	21
I	1.00	1.00	1.00	24
Morning	0.96	1.00	0.98	24
No	1.00	0.94	0.97	16
Not	0.93	1.00	0.97	14
Pizza	1.00	1.00	1.00	20
Place	1.00	1.00	1.00	21
Read	1.00	1.00	1.00	21
School	1.00	1.00	1.00	21
Student	1.00	0.88	0.93	8
Teacher	1.00	1.00	1.00	26
Thank You	1.00	1.00	1.00	15
This	0.96	1.00	0.98	23
Tomorrow	1.00	0.94	0.97	18
Want	1.00	1.00	1.00	24
What	1.00	1.00	1.00	22
Yes	1.00	0.95	0.97	20
Yesterday	0.96	1.00	0.98	25
You	0.94	1.00	0.97	15

Fig. 10 Classification Report

3.4 English Sentence Generation

The Bi-LSTM model was used to classify each word of ISL. However, communication in natural language takes place in the form of sentences and phrases. The syntax of ISL is different from English language and thus, to convert the sentences from Raw Sign Sentences into Structured English Sentences a medium is required. Conventionally NLP [9] techniques like language modeling, lemmatization, syntax restructuring was used to achieve this. However, these approaches fail when dealing with more complex sentences.

We decided to leverage Generative AI for this solution. We used LLM (Llama 3, 8 billion parameters) along with Few Shot Prompting to give the model examples of translations from raw signs into English Sentence. The LLM learns the relationships between the two languages

through these examples. On providing new Raw Signs, the LLM is able to translate effectively, surpassing all previous NLP techniques. The translated sentences always contained the semanticity of the raw ISL.

TABLE III
FEW SHOT PROMPT EXAMPLES

S.No	Raw ISL	English Sentence
1.	"HOME RAIN HEAVY"	"It is raining heavily in my home area"
2.	"CLASS STUDENTS SIT"	"There are students sitting in the class"
3.	"YOU FOOD FINISH?"	"Have you finished your food?"
4.	"I TONIGHT HOME GO LATE"	"I will go home late tonight."
5	"TONIGHT HOME LATE YOU?"	"Will you be late coming home tonight?"
6.	"I LIKE EAT APPLE APPLE"	"I like to eat apples."

3.5 Incorporating Non Manual Components

In Sign Language, Non-manual components such as face expressions play an important role in determining the tone, intent and the intensity of the verb.

We have used a media-pipe library to detect positions of eyebrows and eye corners for each eye. Landmark coordinates differ for different positions of the speaker, therefore we normalized the key-points. Normalizing the points of eyes and eyebrows gives us a scaled value for all the users regardless of their distance from the camera. Determining the threshold value for eyes to detect significant eyebrows raise, or squinting of eyes, helps to distinguish between a simple sentence and an interrogative sentence. To avoid any misclassification, or reduce the human induced error, if the model detects the frowned state, in more than 15 frames, it marks the sentence as interrogative. Otherwise a simple sentence.

3.6 Text to Speech and Reversed Translation

All the signs after being translated through the Bi-LSTM model and structured into English text by the LLM are converted into speech through the gTTS (Google Text to Speech) API.

The process of translation when reversed can provide conversion of English Speech into ISL signs. First the spoken words will be converted into text by the Speech Recognition library. The text will then be converted into Raw ISL signs using the LLM. Finally, each word can be visualized as the sign key-points using OpenCV.

3.7 User Interface

The frontend implementation focuses on creating a responsive and user-friendly interface. HTML5 provides the structural foundation, incorporating video elements for camera feed display and file upload capabilities. CSS3 styling ensures a modern, responsive design that adapts to different screen sizes. JavaScript handles dynamic interactions, managing video streams, file uploads, and real-time updates. The interface includes clear visual feedback mechanisms, loading indicators, and result displays to enhance user experience.

3.8 Backend Integration

Flask serves as the backend framework, managing HTTP requests, file processing, and model integration. The backend implements RESTful APIs for video processing and result generation. It handles video file uploads, processes frames using MediaPipe for landmark detection, and integrates with the trained model for sign language prediction. The system includes robust error handling and validation mechanisms to ensure reliable operation.

3.9 Data Flow Management

The system implements efficient data flow between frontend and backend components. Video data, whether from live camera feed or file upload, is processed through a streamlined pipeline. The frontend sends data via HTTP requests, which the backend processes asynchronously. Results are returned to the frontend for display. Real-time processing is achieved through efficient stream management and WebSocket implementation for live feedback.

CHAPTER 4

RESULTS AND DISCUSSION

Our approach incorporated the following features which have been absent in previous techniques:

1. Comprehensive dataset containing signs with complex movements as well as signs with no movement
2. Normalization of pose landmarks in order to make it resilient to changes in dimension of the video or the position of speaker in the video
3. Normalization of hand landmarks with respect to the wrist to enhance the features and the relations between the fingers
4. Bi-LSTM network with input Convolutional Layer to ensure that both spatial and temporal features are extracted
5. Incorporation of LLMs for sentence level translation which exhibits significantly better results than NLP techniques
6. Bidirectional translation system i.e. Sign to Speech and Speech to Sign, ensuring complete communication

All these steps have resulted in an accuracy of 99.69% and when compared to previous works shows better performance for both static [3]-[5] and dynamic [6], [7], [10] signs.

TABLE IV
COMPARISON OF APPROACHES

S.No	Approach	Sign Type	Accuracy
1.	CNN, SVM [5]	Static (Character)	99.17%, 99.64%
2.	CNN + LSTM [6]	Dynamic (Word)	89.99%
3.	LSTM [7]	Dynamic (Sentence)	72.3%
4.	RNN + CNN [12]	Dynamic (Word)	92%
5.	Our Approach	Static and Dynamic (Word and Sentence)	99.69%

CHAPTER 5

CONCLUSION AND FUTURE SCOPE

This is a stepping stone towards a supportive and inclusive society, and needless to say it has scope for better results. The future work could entail the following:

- Identifying word boundary transitions, with such a vast audience it is difficult to maintain a constant rate at which the user signs. This will create an uneven time series that will be difficult to classify
- Overcoming regional variations in language, just like a change in accent in speaking it is difficult to include variations of the same signs for better and usability for all audiences
- To incorporate intensity of the verb which is induced by non-manual components such as facial expression. which is further reflected in sentiment of the sentence generated

REFERENCES

[1] Fels, S., & Hinton, G. (1995, May). Glove-TalkII: an adaptive gesture-to-formant interface. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 456-463).

[2] Tharwat, A., Gaber, T., Hassanien, A. E., Shahin, M. K., & Refaat, B. (2015). Sift-based arabic sign language recognition system. In Afro-European Conference for Industrial Advancement: Proceedings of the First International Afro-European Conference for Industrial Advancement AECIA 2014 (pp. 359-370). Springer International Publishing.

[3] Aryanie, D., & Heryadi, Y. (2015, May). American sign language-based finger-spelling recognition using k-Nearest Neighbors classifier. In 2015 3rd International Conference on Information and Communication Technology (ICoICT) (pp. 533-536). IEEE.

[4] Garcia, B., & Viesca, S. A. (2016). Real-time American sign language recognition with convolutional neural networks. Convolutional Neural Networks for Visual Recognition, 2(225-232), 8.

[5] Katoch, S., Singh, V., & Tiwary, U. S. (2022). Indian Sign Language recognition system using SURF with SVM and CNN. Array, 14, 100141.

[6] Jayanthi, P., Bhama, P. R. S., & Madhubalasri, B. (2023). Sign Language Recognition using Deep CNN with Normalised Keyframe Extraction and Prediction using LSTM: CONTINUOUS SIGN LANGUAGE GESTURE RECOGNITION AND PREDICTION. Journal of Scientific & Industrial Research (JSIR), 82(07), 745-755.

[7] Mittal, A., Kumar, P., Roy, P. P., Balasubramanian, R., & Chaudhuri, B. B. (2019). A modified LSTM model for continuous sign language recognition using leap motion. IEEE Sensors Journal, 19(16), 7056-7063.

[8] Aditya, C. R., Shraddha, C., & Hegde, R. (2020). ENGLISH TEXT TO INDIAN SIGN LANGUAGE TRANSLATION SYSTEM. Turkish Journal of Computer and Mathematics Education (TURCOMAT), 11(3), 1418-1423.

[9] Jeyasheeli, P. G., & Indumathi, N. (2021). Sentence Generation for Indian Sign Language Using NLP. Webology, 18(SI01), 196-210.

[10] Han, J. S., Lee, C. I., Youn, Y. H., & Kim, S. J. (2022). A study on real-time hand gesture recognition technology by machine learning-based mediapipe. Journal of System and Management Sciences, 12(2), 462-476.

[11] Sundar, B., & Bagyammal, T. (2022). American sign language recognition for alphabets using MediaPipe and LSTM. Procedia Computer Science, 215, 642-651.

[12] Unnathi, E., Sreeja, A. K., Teja, R., & Vinutha, L. V. (2024, July). Real-Time Sign Language Recognition and Translation with Glove and Mobile Integration. In 2024 Second International Conference on Advances in Information Technology (ICAIT) (Vol. 1, pp. 1-6). IEEE.

[13] Department of Empowerment of Persons with Disabilities, (2019). [Online]. Available:

https://divyangjan.depwd.gov.in/islrtc/search.php?type=list&id=4008&search=Eat

[14] Department of Empowerment of Persons with Disabilities, (2019). [Online]. Available:

https://divyangjan.depwd.gov.in/islrtc/search.php?type=list&id=6152&search=Teacher

[15] Google AI for Developers (2024), Pose Landmarks. [Online]. Available: https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker

[16] Google AI for Developers (2024), Hands Landmarks. [Online]. Available: https://github.com/google-ai-edge/mediapipe/blob/master/docs/solutions/hands.md