American Sign Language Fingerspelling Recognition

Jaideep Kotani
200070035
200070035@iitb.ac.in

Gnanendar Reddy Palagiri
200070053
200070053@iitb.ac.in

Ojas Patil
200070052
200070052@iitb.ac.in

***Abstract*—There are almost 1.5+ billion peoples in the world who are affected by hearing loss and more than 70 million of them use sign language to communicate. In sign language, different type of hand gestures are used to match individual words and expressions. Many Deaf smartphone users can fingerspell words more quickly than they can use mobile keyboards. It can really be more faster to fingerspell in ASL than to type on.**

In this paper, we present a comprehensive overview of ASL fingerspelling recognition, including the history of the field, the different approaches that have been proposed, and the current state-of-the-art. We also discuss the challenges of ASL fingerspelling recognition and the future directions of research.

***Index Terms*—ASL(American Sign Language).**

I. Introduction

The ASFLR (American Sign Language Fingerspelling Recognition) project represents a pioneering effort in the realm of accessibility technology, aiming to significantly impact the lives of the Deaf and Hard of Hearing community. This academic paper presents an in-depth exploration of the ASFLR project, focusing on the development and implementation of cutting-edge AI and machine learning techniques to enhance the recognition and usability of American Sign Language (ASL) fingerspelling. The paper elucidates the project's objectives, methodologies, technological innovations, and potential implications for fostering inclusive communication through advanced digital interfaces. Through a rigorous analysis and discussion, this paper illuminates the ASFLR project's pivotal role in advancing accessibility and inclusivity for individuals reliant on sign language as their primary means of communication.

The competition's outcomes could lead to significant advancements, enabling DHH individuals to utilize fingerspelling as an alternative to traditional keyboard input. This technology has far-reaching implications, potentially facilitating faster communication for web search, map directions, texting, and even the development of applications capable of translating sign language inputs into speech for smoother interactions between signers and non-signers.

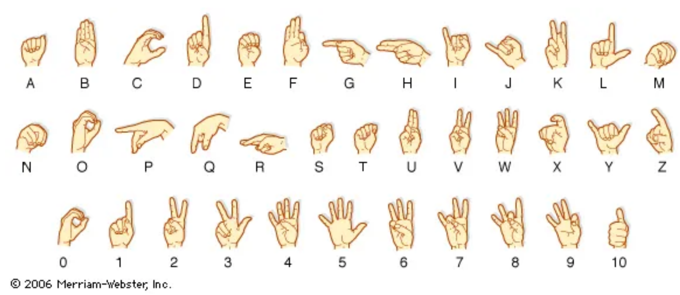

Fig. 1. ASL for different letters

II. Work Distribution

We all sat together and searched for the project ideas and finally found the google's ASL competition and its dataset. All three of us split the code work equally and Ojas and Gnanendar did the report work. Jaideep recorded the video Presentation for the project.

III. Dataset Description

The landmark data. The landmarks were extracted from raw videos with the MediaPipe holistic model. Not all of the frames necessarily had visible hands or hands that could be detected by the model. The landmark files contain the same data as in the ASL Signs competition (minus the row ID column) but reshaped into a wide format. This allows you to take advantage of the Parquet format to entirely skip loading landmarks that you aren't using.

sequence-Id - A unique identifier for the landmark sequence. landmark files contain approximately 1,000 sequences. The sequence ID is used as the dataframe index.
frame - The frame number within a landmark sequence. [x/y/z]_[type]_[landmark_index] - There are now 1,629 spatial coordinate columns for the x, y and z coordinates for each of the 543 landmarks. The type of landmark is one of ['face', 'left_hand', 'pose', 'right_hand'].

The labels for the landmark sequence. The train and test datasets contain randomly generated addresses, phone numbers, and urls derived from components of real addresses/phone numbers/urls. Any overlap with real addresses, phone numbers, or urls is purely accidental. The supplemental dataset consists of fingerspelled sentences. Note that some of the urls include adult content. The intent of this competition is to support the Deaf and Hard of Hearing community in engaging with technology on an equal footing with other adults. Details of the hand landmark locations can be found here. The spatial coordinates have already been normalized by MediaPipe.
Note that the MediaPipe model is not fully trained to predict depth so you may wish to ignore the z values. The landmarks have been converted to float32. Landmark data should not be used to identify or re-identify an individual. Landmark data is not intended to enable any form of identity recognition or store any unique biometric identification.

IV. Experimentation and Results

A. Dataset

Calculating the mean and standard deviation for different landmarks, separating them based on left and right hand indices, plots boxplots for each category, and saves the resulting statistics for use in a neural network model.

Taking data on the number of frames per character, converting it into a Pandas Series, which displays descriptive statistics along with a histogram plot to visualize the distribution and variation in the number of frames per phrase character, focusing on the 99th percentile.

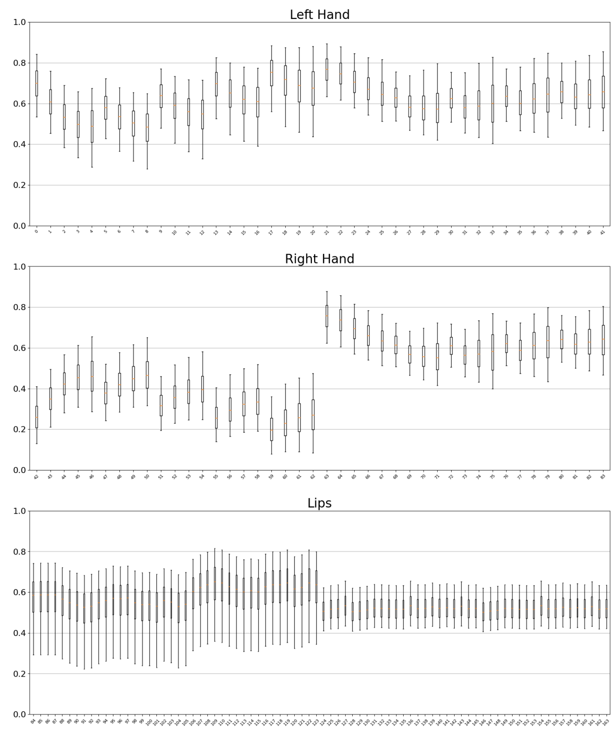

Fig. 2. Coordination Statistics

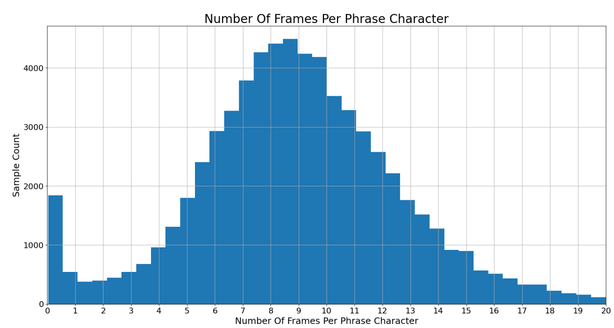

Fig. 3. Coordination Statistics

B. Model Descrition

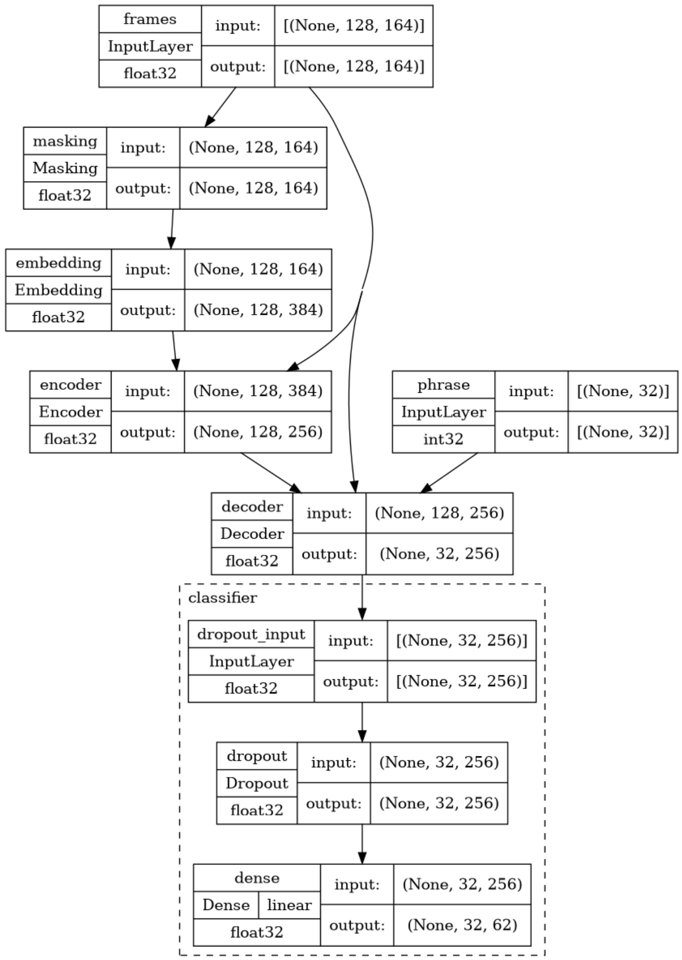

Fig. 4. Model Description

	phrase_true	phrase_true_len	phrase_pred	levenshtein_distance
0	3 creekhouse	12	+33-33--3-3-1-3-3-3-3-3	22
1	scales/kuhaylah	15	+32-20-3-3-3	15
2	hentaihubs.com	14	+44-444-4444	14
3	1383 william lanier	19	+99-99-9-999-9-9-9-99	21
4	988 franklin lane	17	+32-28-8-8-8	17
5	6920 northeast 661st road	25	+38-888-888-0-8-8-8	25
6	www.freem.ne.jp	15	+35-000-0-0-0-0	15
7	https://jsi.is/hukuoka	22	+32-21-1-1-1	22
8	239613 stolze street	20	www.thththththtttththtththththtt	28
9	242-197-6202	12	www.sthesthesth sth st	22
10	271097 bayshore boulevard	25	+38-888-8888	25
11	federico pearson	16	+32-22-2-0-1	16
12	/carpina/hope_&_faith/litle	27	+38-88-8-8-8	27
13	dine-in/code/	13	www.comaromaromaromadarada	24
14	+264-97-568-217-145	19	+32-22--2-8-2-1-1-1-1	13
15	+51-2721-208-63	15	+91-11-1-1-1	9
16	wildberries_ru	14	+22-22-2-2-2	14
17	leona owens	11	+99-399-3999	12
18	+220-557-859-04	15	+25-2-1-1-1-1-1-1-1-1	15
19	kati castro	11	+22-22-2-8-2	12
20	5566 hellertown road	20	+21-21-1-11-1-1-1-1	20
21	6867 granville drive	20	+99-29-2-999-9-9-999999999	26
22	1600 fire water	15	+22-2-1-2-1-1-1-1-1-1	20
23	+45-39-007-1887	15	+22-20-2-1-1	10
24	65634/tennessee%20river	23	+39-29-2-1-1	21
25	596-033-4046	12	+95-444-444-9-1-5	12
26	18 cutter ridge road	20	+34-44-1-1-1	20
27	tampa fl	8	+39-000-0-0-0-0-0-0	19
28	492288 west 28th terrace south	30	+39-8-8-8-8-	28
29	166 water power	15	+88-887-8777pom/pom	17

Fig. 5. Training and Validation loss/accuracy

C. *Levenstein Distance Train*

The Levenshtein distance is a string metric for measuring the difference between two sequences. Informally, the Levenshtein distance between two words is the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other. We use SVM and Random Forest classifiers on top of the transfer learning models to classify the pneumonia images. Extract features from the transfer learning models and use them as input for SVM and Random Forest classifiers and train the classifiers on the training set and validate them on the validation set. Tune the hyperparameters of the classifiers using grid search or other methods to improve their performance.

Using Fig. 5 a dictionary to count occurrences of each Levenshtein distance value in the training

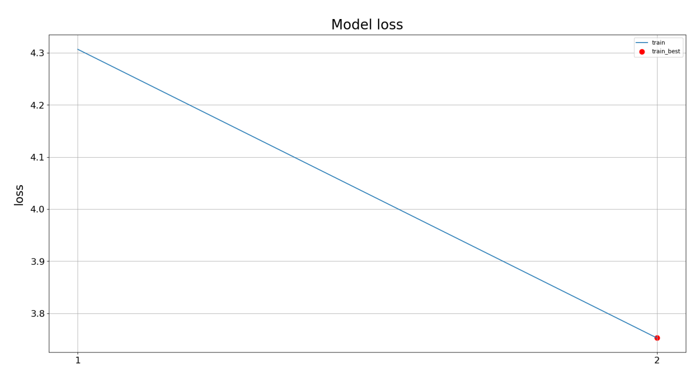

Fig. 6. Model loss

dataset, then we calculates the normalized Levenshtein distance as an evaluation metric. Finally, we visualizes the distribution of Levenshtein distances in the training data using a bar plot, incorporating the mean Levenshtein distance and the calculated normalized Levenshtein distance (NLD). In order to find NLD:

$$NLD = (N - D)/N$$

where,

$$N = \sum_{phrase\_length} l_d$$

$$D = \sum_{levenshtein\_distance} l_d$$

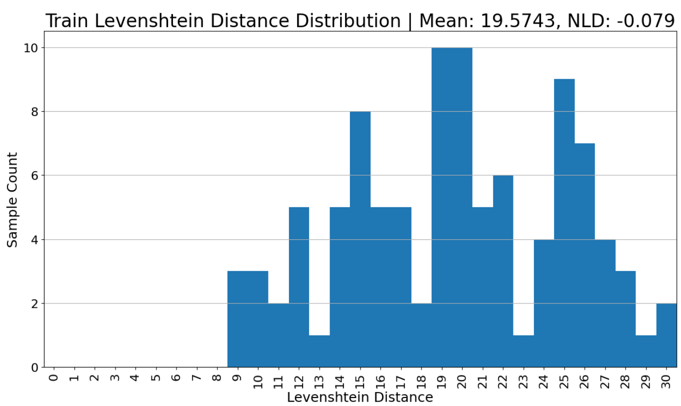

Fig. 7. Levistein distance distribution

Compare the performance of the models and select the best-performing model for further fine-tuning. Validate the final model on the test dataset and report its accuracy and other evaluation metrics. Show the experimentation results prospectively.

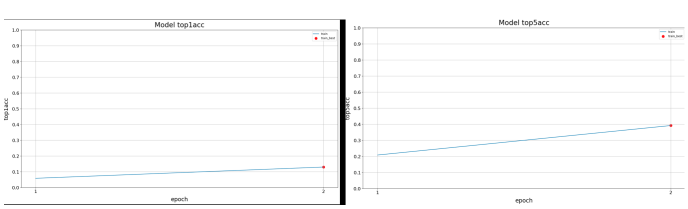

Fig. 8. Accuracy Evolution

V. Conclusion

The dataset includes over 200,000 frames of ASL fingerspelling video, and the code includes several different approaches to ASL recognition.

The results of the project suggest that ASL recognition is a feasible technology that can be used to develop real-world applications. These applications could include:

Real-time translation of ASL: This would allow people who are deaf or hard of hearing to communicate more easily with people who are hearing. ASL-based search and navigation: This would allow people who are deaf or hard of hearing to use search engines and navigation systems without assistance. ASL-based education and training: This would allow people who are deaf or hard of hearing to access educational and training materials that are currently not available to them.

VI. References

1) TensorFlow ASLFR
2) 1st Place Solution - Training
3) 1st Place Solution - Inference
4) Fingerspelling Detection in ASL paper
5) ASLFR using Hybrid Deep Learning
6) TopKAccuracy
7) Categorical Croosentropy
8) Learning Rate Scheduler
9) Levenshtein Distance

References