Translating Sign Language to Speech

Francisco del Castillo, Mervin McDougall, David Plotkin

Syracuse University
School of Information Studies

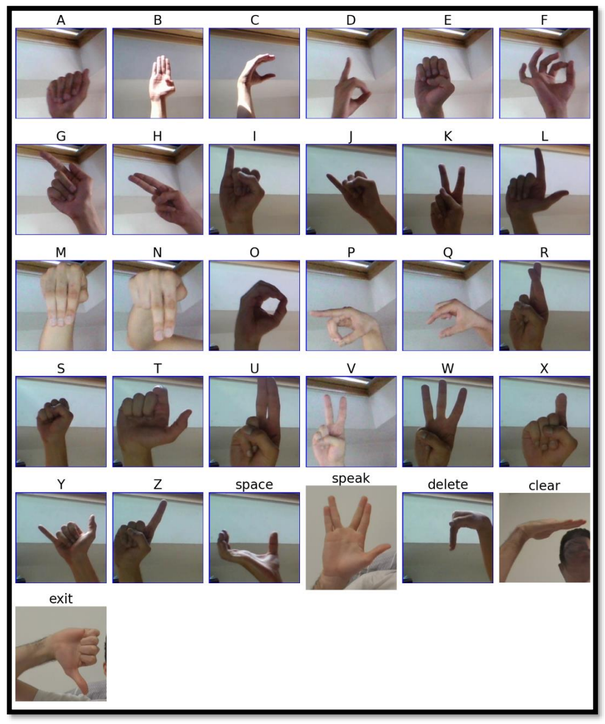
Introduction

Although most of us enjoy the convenience of verbal-driven AI assistants, some of us cannot participate and take advantage of this feature. Those who cannot speak are tethered to the device by a keyboard whether on-screen or physical. Through our application, we are proposing a different approach where Artificial Intelligence (AI) is leveraged to provide the convenience of fluid communication for those of us who are speech impaired by translating sign language into text or audio for a more familiar and natural manner of communication. Our application - the ASL (American Sign Language) Translator named “Inner Voice” - allows a user, through sign language, to issue commands to an AI, communicate with family or friends, and participate in social settings and society in a second-nature way.

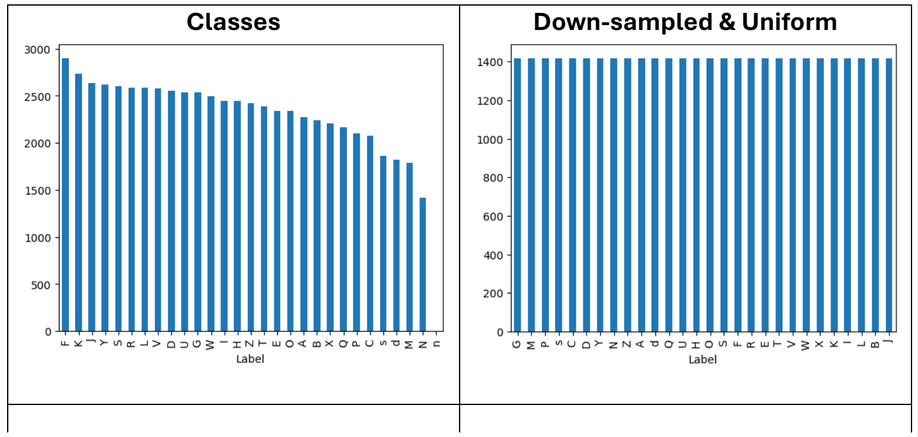
Data Description

The dataset, retrieved from Kaggle, contains 87,000 images of American Sign Language (ASL) gestures categorized into 29 classes: 26 alphabetic letters (A–Z) and three additional classes (“space,” “delete,” and “nothing”). However, the "nothing" class was dropped due to compatibility issues with Google’s MediaPipe API. Each alphabetic class contains 3,000 images, each sized 200x200 pixels in RGB format. To enhance interactivity in the application, three additional classes (“Speak,” “Clear,” and “Exit”) were introduced.

Google MediaPipe was utilized to preprocess the images, translating hand gestures into a series of x-y-z coordinates and normalizing them. This preprocessing step generated a secondary dataset where the gestures are represented in a tabular format with x and y coordinates. The resulting dataset, which includes 31 classes, was used to train the models. These classes are illustrated in the following picture.



Frequency Distributions

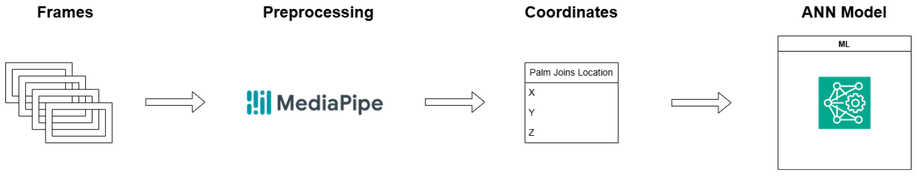


Model Selection

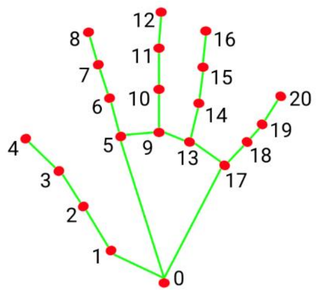
3 different neural network configurations were tested with increasing levels of accuracy. The final model consists of 42 Inputs, a hidden layer of 84 neurons, and 31 neuron SoftMax output (model 3).

Model	Configuration	Reasoning
Model1	Input (42), SoftMax (31)	Test the ANN at its most basic i.e. input and output minimums.
model2	Input (42), Dense (21), Dense (10), SoftMax (31)	Use an autoencoder architecture to reduce the number of neurons to a code of 10 and then inflate the number of neurons in the output.
model3	Input (42), Dense (84), SoftMax (31)	Increase the number of neurons to get more data representation.

Model Training Pipeline



The pipeline begins with each frame or hand image being processed through a call to the MediaPipe API. This API returns the (x, y, z) coordinates of 21 hand and finger landmarks by overlaying a geometric “skeleton” on the hand (next picture). These extracted coordinates are then used to train the Neural Network model.



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

Models Performance

All 3 Models were trained for 20 epochs on preprocessed data. We chose model 3 because it had the highest F1 score.

Model	Loss	Accuracy	F1
Model 1	24.02%	95.60%	95.70%
Model 2	14.03%	96.79%	96.70%
Model 3	6.51%	98.67%	98.45%

Final Implementation

The application interface features a video screen where the model identifies the hand, overlays the landmarks on it, and classifies the gesture with a confidence score. For instance, in the picture provided, the model classifies the gesture as the letter “L” with a 99.5% confidence.

Below the video feed, there is a text box where the predicted letters are spelled out to form words. For a letter to appear in the text box, the model must classify it with a confidence score higher than 70% for at least 2 seconds.

Finally, the interface includes quality-of-life buttons to clear the message, quit the application, and make the app "say" the message. The "Say it" option sends the message to the Google Text-to-Speech (GTTS) API, which returns an MP3 recording of the message in English and plays it out loud. These three actions (“clear”, “quit”, and "say it") can also be triggered using specific gestures.

Limitations & Future Research

One limitation of our project is the lack of data augmentation techniques due to time constraints. As a result, the model is restricted to recognizing signs performed with the right hand, which must remain straight and free of rotation. Future research could address these by incorporating data augmentation.

Additionally, future work might explore translating American Sign Language (ASL) through motion-based signs. While our focus was on the ASL alphabet, it is well-known that ASL also includes motion-based signs representing complete words and sentences. This new direction may require different architectural approaches, such as Convolutional Recurrent Neural Networks (CRNN), to effectively process and interpret dynamic gestures.