Sign Language Recognition

by

Varun - 02070206
Tilak - 02020599
Sanjana - 02075870
Ghanshyam - 02070793

Under the Guidance of
Prof. **AshokKumar R Patel**

Submitted to the Department of Computer Science.

University Of Massachusetts
North Dartmouth, Massachusetts

Abstraction

Sign language recognition is a project that uses technological breakthroughs to help those who are visually impaired. We intended to construct a sign detector for this reason. We are attempting to recognize and display messages based on hand movements with this project. The advancement of existing technologies, as well as extensive study, are employed to help the deaf and dumb. Because comprehending sign language is not something that many people have, this can be highly valuable for deaf and dumb people connecting with others. Python modules such as OpenCV, Tensorflow, and Keras are needed to complete this assignment.

Introduction

1.1. Motivation

It can be quite difficult to communicate with someone who has hearing loss. People who are Deaf or Mute use hand gestures to communicate, which makes it difficult for people outside of their society to understand the signs they use. As a result, there is a need for systems that can identify various indicators and inform regular people.

1.2. Problem Statement

When hand and other movements are employed in sign language, a sign detector is utilized to identify them. Based on these unique movements, the detector then shows signals for the persons who require special assistance. In this context, machine learning and computer vision ideas can be applied.

1.3. Sign Language Recognition

The process of turning a user's movements into text is known as sign language recognition. It aids those who are unable to engage with others, thereby minimizing the apparent divide between those who require special assistance and the broader population. Using image processing methods and neural networks, raw photos and videos are converted into text that can be read and understood. This ties the gesture to the text that corresponds to it in the training

data. Due to the fact that just a small number of their gestures are generally understood, it has been observed that dumb people frequently find it difficult to interact with ordinary people.

The desire for a computer-based system is quite great for the community of the dumb in this technological age. The objective is to familiarize the computer with human interaction so that it can recognize it and produce a user-friendly human-computer interface (HCI). Training your computer to recognize facial expressions and hand gestures is one step in this method. A person can use a number of different gestures at once in nonverbal communication. For computer vision researchers, this is a very exciting topic because human signals are perceived visually. To encode these movements in machine language and aid in the development of an HCI that can recognize human gestures, sophisticated programming techniques are needed.

Architecture Diagram

The architecture of the model for recognizing sign language offers a guide and suitable methods to use when creating a well-structured application that meets our requirements. Three phases primarily comprise this architecture:

Phase 1: Data Collection: A model is being developed, and it is being fed a series of photographs. This stage is important for subsequent symbol-based model training.

Phase 2: Training and Testing: In this phase, a set of model inputs are used, and a certain outcome is anticipated. The correctness of the model is being determined based on the results.

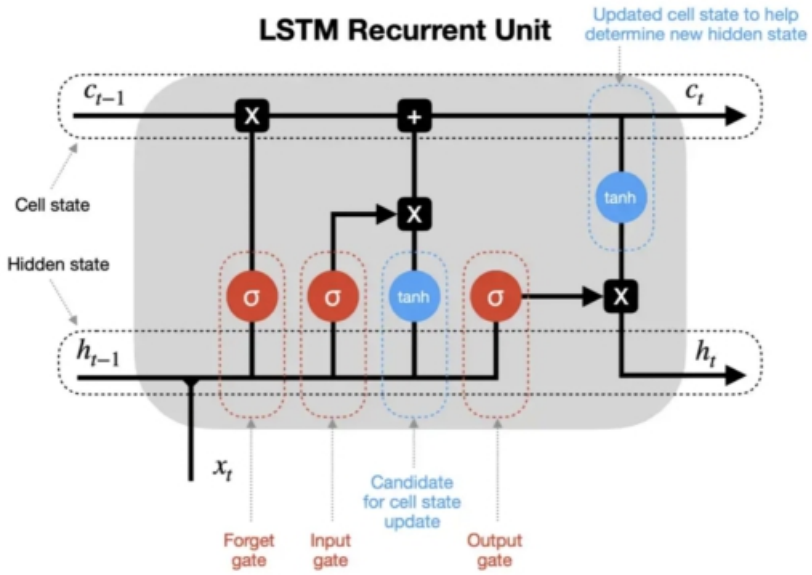
Phase 3: Output recognition: An image is provided to the model as input during this phase. It generates a message and matches each image with a certain output based on the photos being trained into the model. The sign being recognized in the previous step has a specific meaning, and the training phase is where a message is assigned.

Software requirement specifications

- Python
- IDE (Jupyter)
- Numpy
- CV2 (OpenCV)
- Keras
- Tensorflow
- Matplotlib
- Mediapipe

Model : LSTM Neural Network

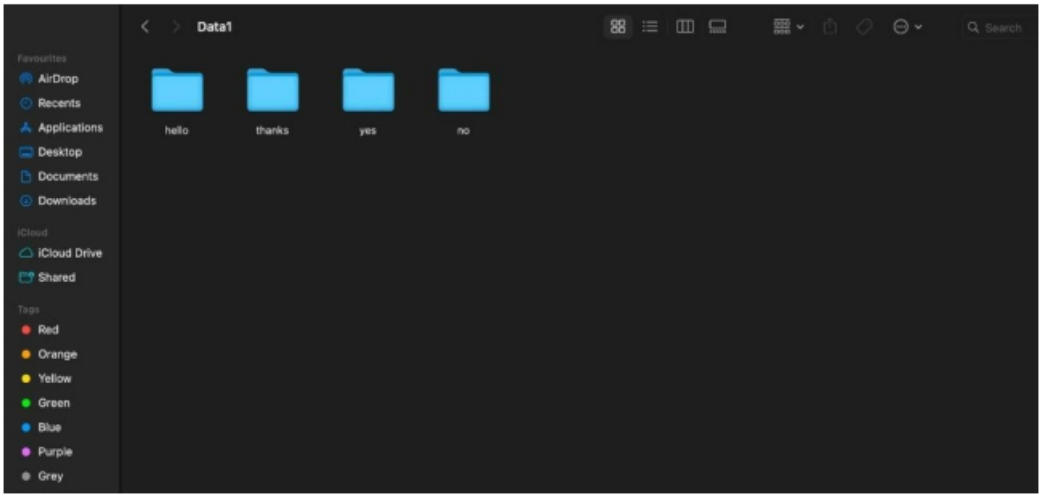
An artificial neural network called Long Short-Term Memory (LSTM) is employed in deep learning and artificial intelligence. LSTM features feedback connections as opposed to typical feedforward neural networks. Such a recurrent neural network (RNN) may analyze whole data sequences in addition to single data points (such as photos). For instance, LSTM can be used for applications like speech recognition, robot control, machine translation, networked, unsegmented handwriting recognition, video games, and healthcare.

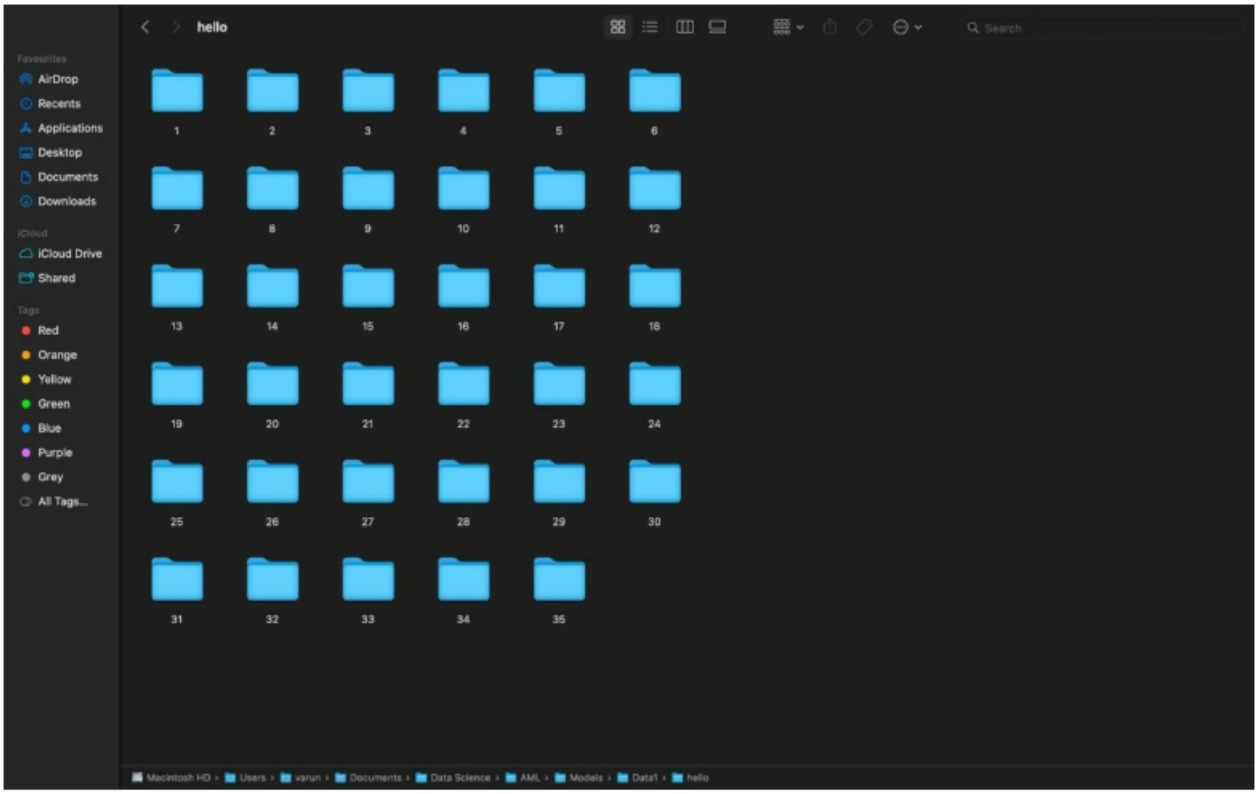


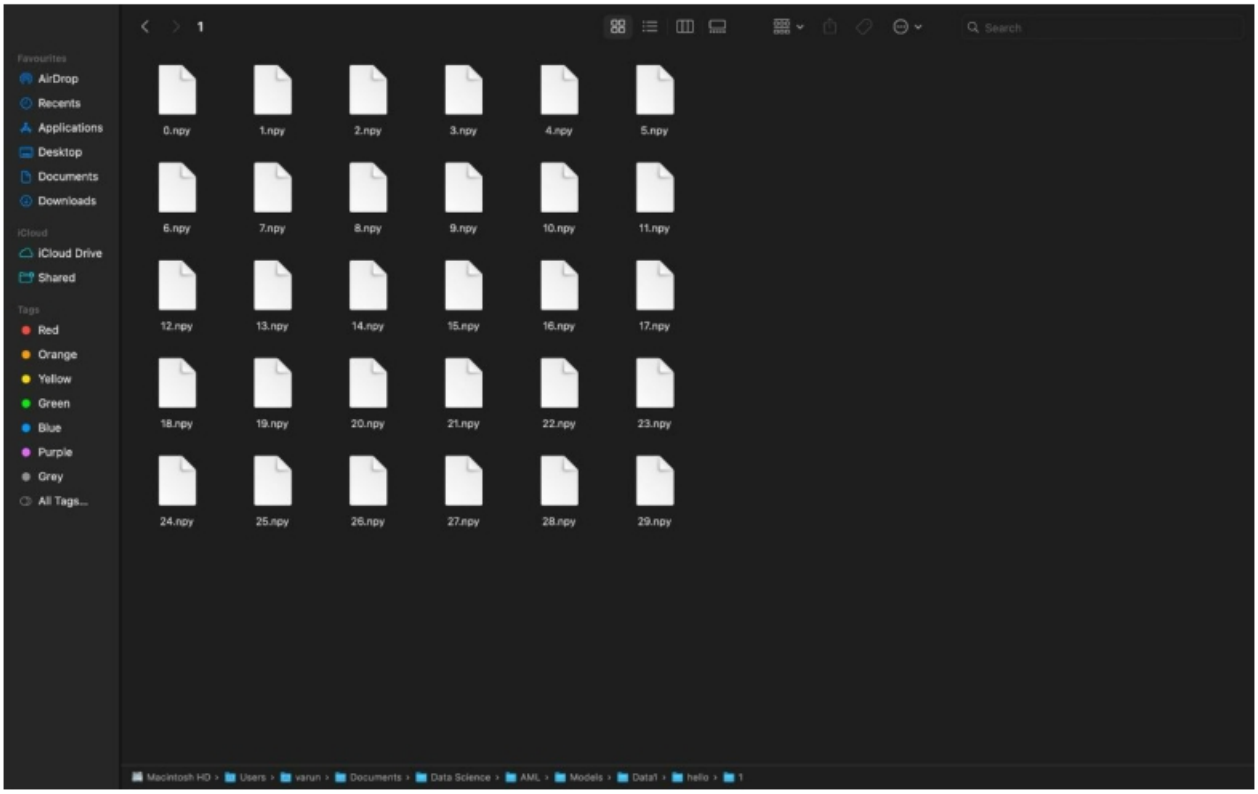
Since there may be lags of uncertain length between significant occurrences in a time series, LSTM networks are well-suited to categorizing, processing, and making predictions based on time series data. To solve the vanishing gradient issue that can arise when training conventional RNNs, LSTMs were created. In many cases, LSTM has an advantage over RNNs, hidden Markov models, and other sequence learning techniques due to its relative insensitivity to gap length.

Training and Testing

The dataset we require is reasonably easy to find online, however for this project, we will be building the dataset ourselves.
Every frame that recognizes a hand in the ROI (region of interest) will be streamed live from the camera, and every frame that does so will be saved in a directory. In our project, we will teach 4 sign languages (hello, thanks, yes, no), and each one will have its own folder. Each folder will include 35 videos, each of which has 30 frames.




hello
Search
Favourites
AirDrop
Recents
Applications
Desktop
Documents
Downloads
iCloud
iCloud Drive
Shared
Tags
Red
Orange
Yellow
Green
Blue
Purple
Grey
All Tags...
1 2 3 4 5 6
7 8 9 10 11 12
13 14 15 16 17 18
19 20 21 22 23 24
25 26 27 28 29 30
31 32 33 34 35
Macintosh HD › Users › varun › Documents › Data Science › AML › Models › Data1 › hello

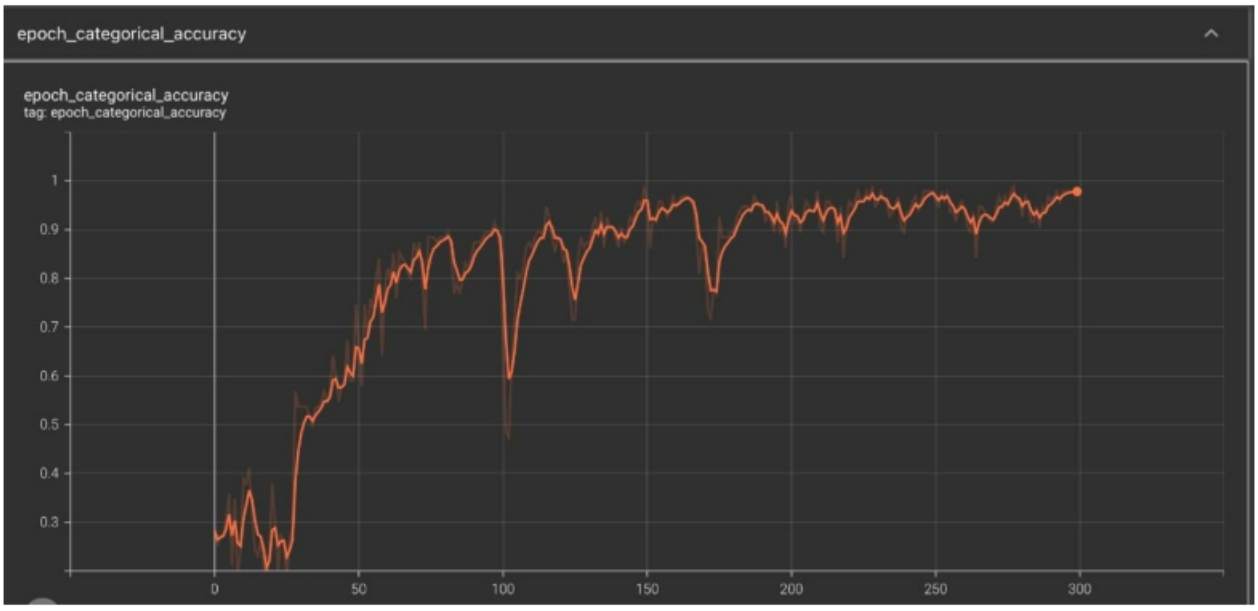

1
Search
Favourites
AirDrop
Recents
Applications
Desktop
Documents
Downloads
iCloud
iCloud Drive
Shared
Tags
Red
Orange
Yellow
Green
Blue
Purple
Grey
All Tags...
0.npy 1.npy 2.npy 3.npy 4.npy 5.npy
6.npy 7.npy 8.npy 9.npy 10.npy 11.npy
12.npy 13.npy 14.npy 15.npy 16.npy 17.npy
18.npy 19.npy 20.npy 21.npy 22.npy 23.npy
24.npy 25.npy 26.npy 27.npy 28.npy 29.npy
Macintosh HD › Users › varun › Documents › Data Science › AML › Models › Data1 › hello › 1

All videos were processed using the Google MediaPipe hand tracking technology once the video augmentation was prepared and stored. A Python script was developed to enter the videos. The videos are sent to MediaPipe by this Python script after it has searched for them in a folder. The outputs are placed in fresh directories by the final modified version of MediaPipe.

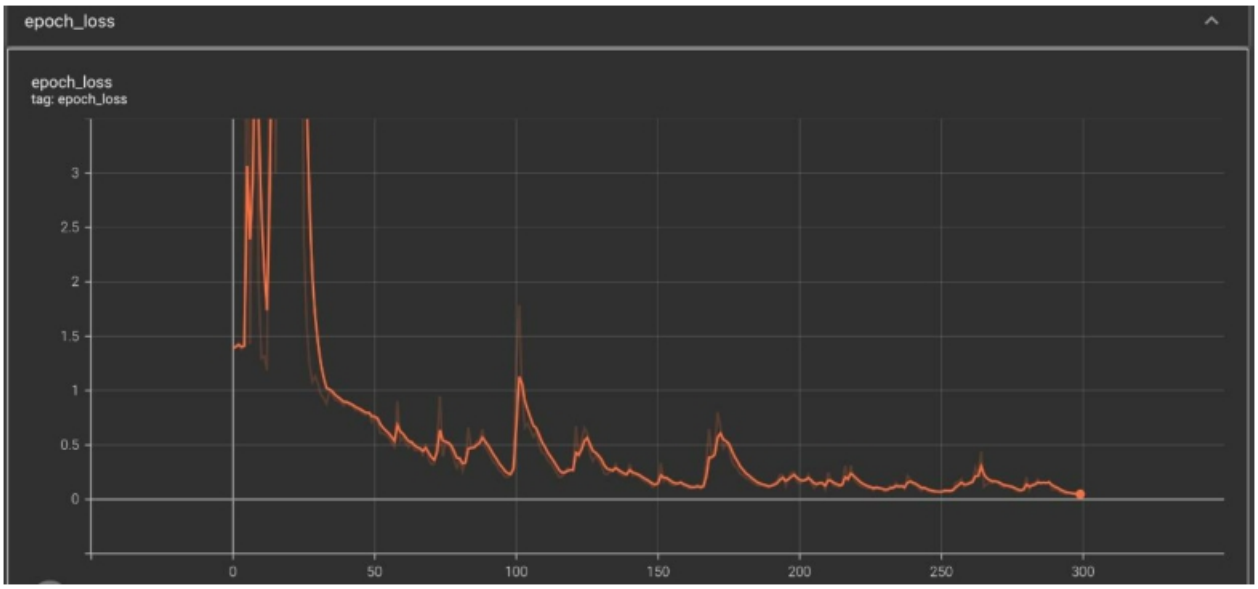
We built a Deep learning model where we had used five “LSTM” layers and three Dense layers. To make training faster “RelU” activation function is used. Optimizer we have used is “Adam” as it was taking less time for training. Have used “categorical_crossentropy” as it is a multiclass classification. The model has been trained on 300 epochs.

The Graphs that are shown below are from tensorboard. As you can see, the starting loss was high so the accuracy was very low. When the losses decreased eventually accuracy started to increase.

Training Accuracy Graph



Loss per epoch



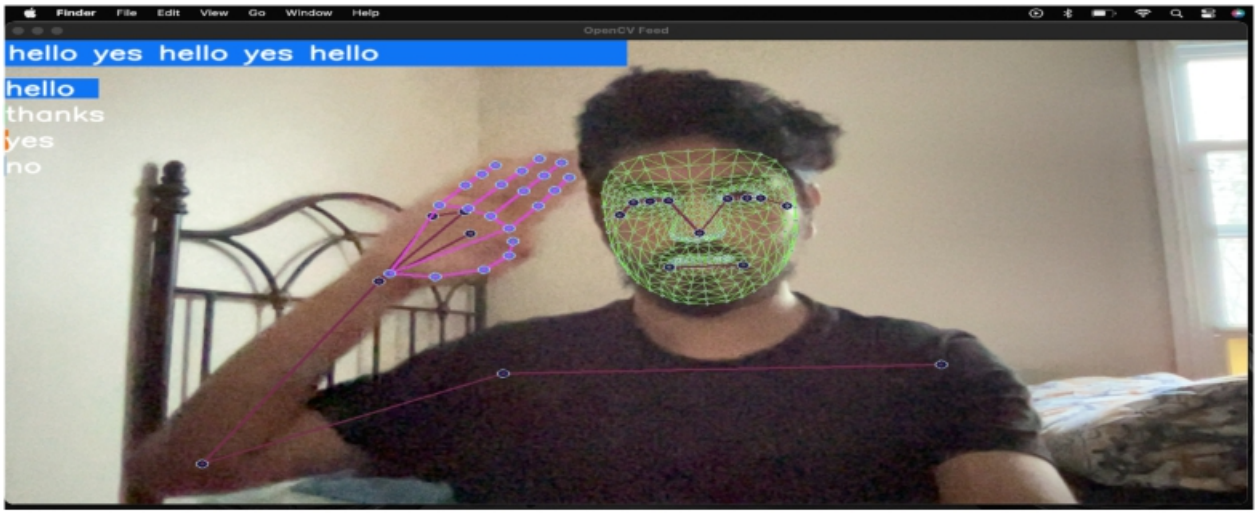
Accuracy:

For the above created model we got an accuracy of **97%** for training and **92%** for testing.

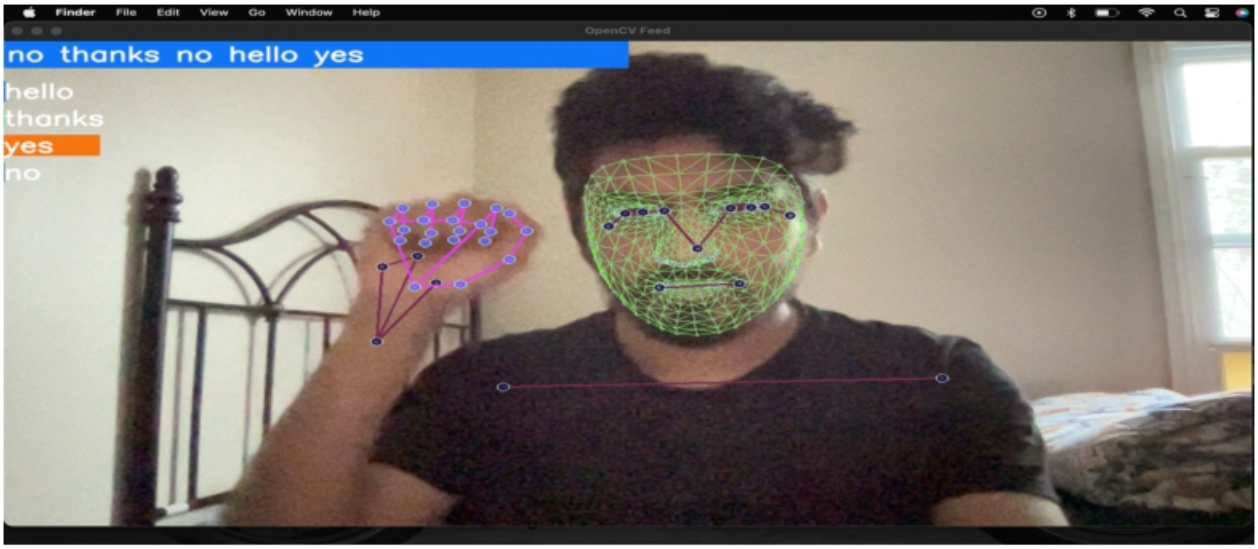
Result:

Below are the screenshot we have got from live feed.

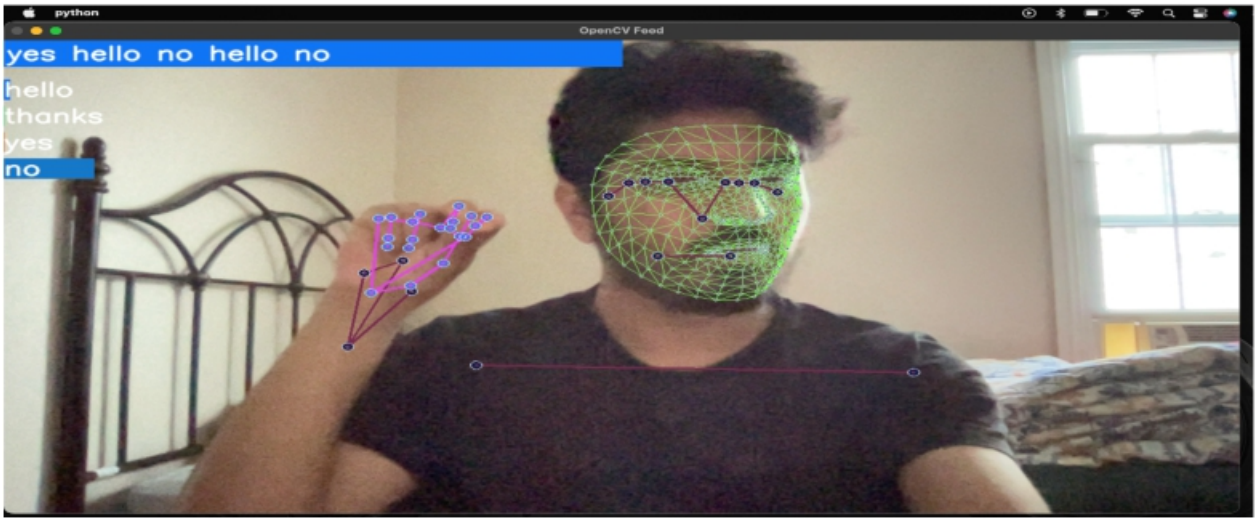
1. **Hello**



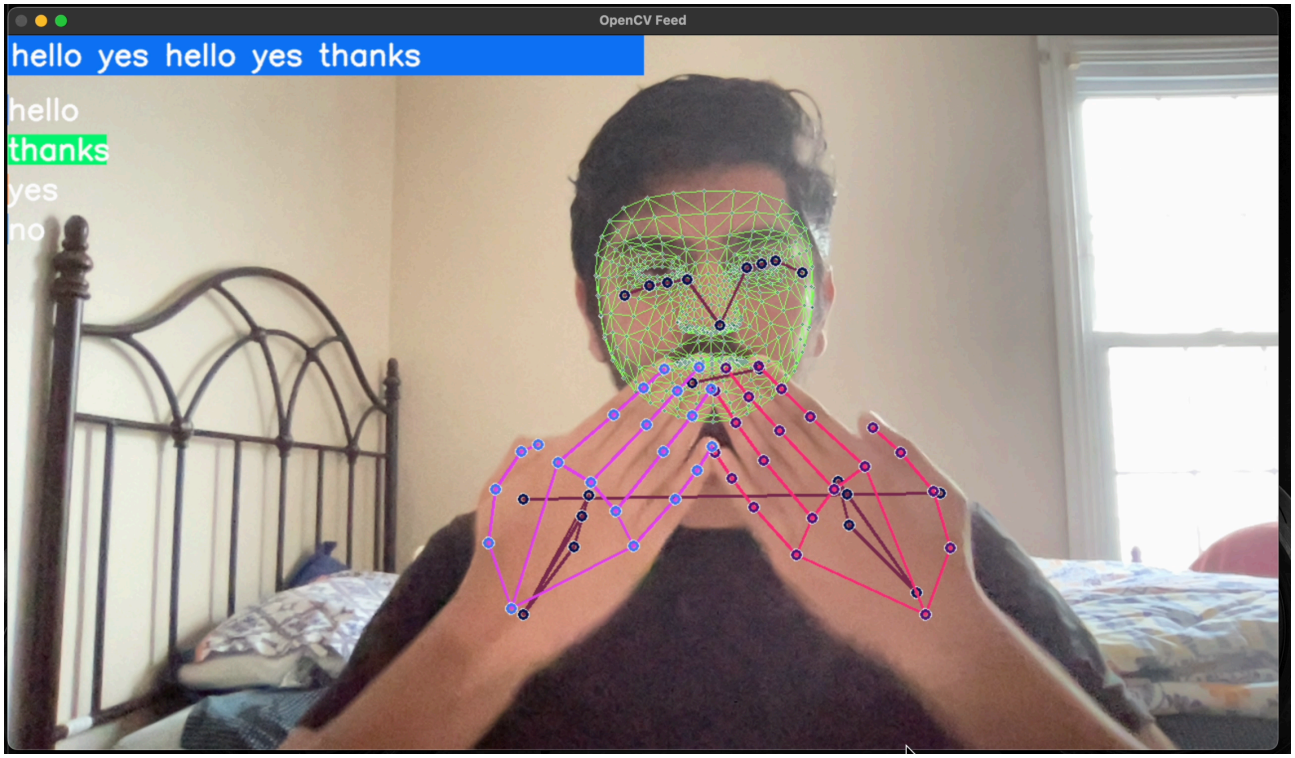
2. **YES**



3. NO


python
OpenCV Feed
yes hello no hello no
hello
thanks
yes
no

4. THANKS


OpenCV Feed
hello yes hello yes thanks
hello
thanks
yes
no

7. Conclusion

A sign language recognition model can be created using Python modules like OpenCV and Keras to help the disabled community. Here, a set of inputs, like photos, are used to train the created model before it is used to generate a series of messages after recognizing a particular hand motion.
This model has the unique ability to be trained to create a variety of messages using custom hand gestures, thereby bridging the communication gap between people who require special assistance and the general public.

References

1. https://google.github.io/mediapipe/solutions/holistic.html
2. https://docs.opencv.org/4.6.0/
3. https://www.analyticsvidhya.com/blog/2021/01/understanding-architecture-of-lstm/
4. https://mediapipe.dev/