SIGN TO SIGN TRANSLATION USING MEDIAPIPE

U.V. Sai Madhumita[1], Meghna Malla[2], Ponnada Nikheel Deo[3], S. Venkat Sai[4], Dr. Mrudula Owk[5]

[1,2,3,4] Student, Dept. Of Computer Science Engineering, GITAM Institute of Technology (GIT), GITAM University, AP, INDIA

[5]Assistant Professor, Dept. Of Computer Science Engineering, GITAM Institute of Technology (GIT), GITAM University, AP, INDIA

Abstract - *Communication is the cornerstone of human interaction, enabling individuals to express thoughts, emotions, and intentions. Whether verbal or non-verbal, language plays a pivotal role in fostering understanding and connection. However, the diversity of sign languages, such as American Sign Language (ASL) and Indian Sign Language (ISL), presents unique challenges for cross-cultural communication. To address this, our work aims to bridge the gap between ASL and ISL through innovative technology solutions. The model takes ISL gestures as input. Accurate translation between these languages is enabled by training the model on comprehensive datasets of ASL and ISL gestures. The model involves Machine Learning to capture data and create datasets (using OpenCV and Mediapipe), Deep Learning models to recognize the dynamic gestures (RNN architectures to deal with Sequential data), and Analysis tools such as NLTK (VADER for Sentiment Analysis). The model is trained on LSTM, LSTM+GRU, and GRU neural network architectures. Upon training, the network with the highest accuracy is used to implement the final gesture detection. The final result is given in ASL gestures (in GIF format) along with the tonality of the subject matter. We have noted from training and testing the model that the GRU model shows promising accuracy for the three models, followed by LSTM+GRU and LSTM.*

Keywords: *American Sign language (ASL), Gesture recognition, Indian Sign language (ISL), Machine learning, Mediapipe and Sign language.*

1.INTRODUCTION

Communication is a fundamental aspect of human life, enabling interaction and meeting basic needs. From making requests to sharing problems, various tasks require communication. While verbal communication is the primary mode for hearing individuals, non-hearing communities have developed regional sign languages. Sign language involves hand gestures and facial expressions representing words or phrases. Despite its local efficiency, global communication between sign language users faces challenges due to regional variations. Many deaf individuals are not taught to read; written languages differ by location. To address this gap, we have proposed a system that converts one language's sign language gesture to another. The focus was mainly on ASL – the most popular sign language, and ISL- our local sign language. This initiative aims to improve cross-cultural communication among sign language users worldwide.

1.1 American Sign Language (ASL)

ASL, a visual language predominantly employed by Deaf, Mute, and hard-of-hearing communities in the United States and parts of Canada, possesses its grammar and vocabulary. Communication in ASL involves the use of hand shapes, facial expressions, and body movements. Beyond North America, various ASL dialects are utilized in several West African and Southeast Asian countries. Originating nearly two centuries ago as a fusion of local sign languages and the French Sign Language (LSF), ASL exhibits a close relationship with LSF. ASL comprises 26 gestures corresponding to each letter in the alphabet and distinct signs for numbers 1-10. While precise statistics are elusive, an estimated half a million people in the US use ASL as their natural language.

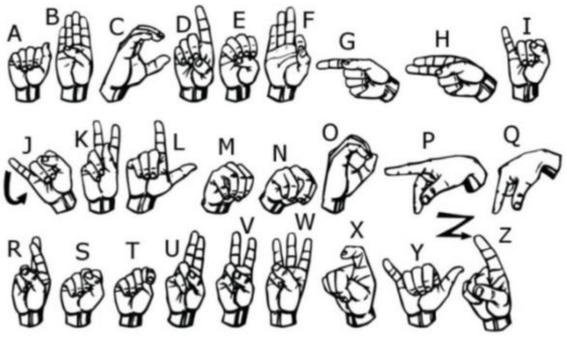


Fig -1: ASL alphabet gestures

1.2 Indian Sign Language (ISL)

ISL is the sign language used in India since it is a country of linguistic riches and countless regional dialects. This language is gestural and is expressed using the hands and face with origins in the early 20th century. Indo-Pakistani Sign Language is used by about 15 million deaf persons in the South Asian subcontinent. Therefore, ISL, like ASL, has unique signs for each letter of the alphabet. Legal recognition of the sign language in India shows the use and application of this language in the country's communicational growth process. ASL and ISL are vital tools in ensuring effective communication and supporting their community development.

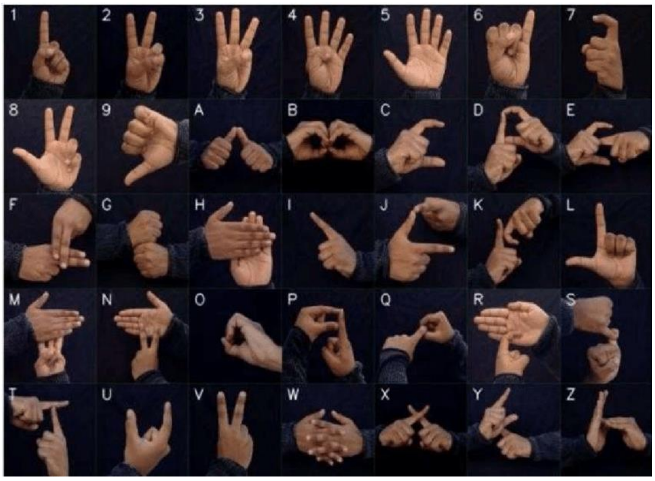


Fig - 2: ISL alphabet gestures

1.3 Mediapipe

Media Pipe, which is a Google's library-based system specifically for all sorts computer vision tasks, now features in the platform vision tasks. The Mediapipe is the platform which mastered the problem of holistically estimation of a pose. This consists of the precise locating of iconic features such as the eyes, nose, and the mouth as well as the body stance and hand movements. The utilization of Mediapipe in the advancement of the technology demonstrates its critical role in the technology transformation. This model takes a comprehensive pose estimation to a higher level, where it is applied to numerous applications like augmented reality facial recognition, gesture handling and an interactive interface. [16] On-device ML can be difficult. MediaPipe Solutions makes it easy. You can customize advanced solutions to your needs, quickly and seamlessly. These flexible tools are built on top of TensorFlow Lite for the best end-to-end on-device ML and hardware performance.

1.4 Sentiment Analysis

Sentiment recognition is an operational process that is employed for the purpose of describing the emotion that is communicated. It will trigger the changes in media sphere in the years to come. When it comes to the sentiment analysis mentioned, it is done with the help of VADER sentiment analysis library. This text is aimed to reveal the emotional state of the given sentence with the sentence one can consider the presence of the amount of positive, neutral, and negative emotional themes in the plot. The sentimentIntensityAnalyzer class from the VADER library is made to fulfil this purpose. The analysis results in a polarity score for each sentiment category: good-neutral-negative.

2. LITERATURE REVIEW

The research on hand gesture recognition consists of various methodologies and techniques which aim to achieve high accuracy. One notable approach discussed in [1] utilizes a vision-based methodology employing static and dynamic hand gesture recognition. This method uses the combination of Mediapipe and LSTM (Long Short-Term Memory) models, achieving an impressive accuracy of up to 99%. Similarly, [3] utilizes Mediapipe for landmark extraction but diverges by employing ML techniques such as the Random Forest algorithm for classification tasks. Meanwhile, [6] introduces a lightweight model suitable for deployment on smart devices, incorporating SVM (Support Vector Machine) algorithms alongside Mediapipe.

In exploring regional languages like Assamese, [2] proposes a method utilizing 2D and 3D images of gestures trained through a feedforward neural network. This novel approach seeks to bridge language gaps by enabling gesture-based communication. In contrast, [10] extends this concept by incorporating SVM and Mediapipe, focusing on recognizing alphabets, numbers, and commonly used words.

Many even experimented with specialized architectures and optimizations. For instance, [8] introduces MediaPipe optimized GRU (MOPGRU), modifying activation functions and layers to enhance performance. Additionally, [9] employs a transformer model to discern the most significant Mediapipe landmarks for prediction tasks, offering insights into feature importance.

Moreover, researchers have explored diverse models and datasets tailored to specific requirements. [4] adopts a straightforward yet comprehensive approach, detecting individual alphabet gestures in sign language and assembling them into sentences, subsequently converted into speech. Conversely, [11] focuses on dataset creation using media hand tracking and RNN (Recurrent Neural Network) training, emphasizing the importance of curated datasets for accurate recognition.

Furthermore, [5] and [7] delve into advanced models like LSTM, GRU, and feedback-based architectures, showcasing their effectiveness in capturing temporal dependencies for gesture recognition. Additionally, [12] compares CNN (Convolutional Neural Network) and LSTM models for static and dynamic gesture recognition, shedding light on their strengths and weaknesses.

3. PROBLEM IDENTIFICATION & OBJECTIVES

3.1 Problem Statement

Translating the signs from one sign language to another is very difficult because of the characteristic features of the language. There are dissimilarities among sign languages such as ASL and ISL. These languages are grammatically complicated, not only in structures but also in syntax, which hinders finding one standardized translation. Furthermore, effective communication entails a factor of culture that is also a complex concept. Moreover, a sign operating as an icon that implies one direct idea will take more work to understand in translation. We should also consider that deaf people from different nations can use entirely different sign languages for communication. While standard sign languages are generally used, the main problem is overcoming this gap by translating signs from one language to another for improved communication.

3.2 Existing System

The pre-existing methods in sign-language accessibility and inclusivity deal mostly with static gestures or alphabet datasets using CNNs. They focus on various fields of sign language, which include sign language recognition, sign language generation, avatar creation, real-time conversion of sign language, multi-sensor integration, sign language dialect recognition, etc. Although they present with high accuracy, it can be inconvenient to use as every word needs to be spelled out. Moreover, the existing technologies concert either the sign gestures to text or vice versa. This approach largely overlooks the fact that many deaf and mute people and people from the non-hearing spectrum are unable to read. They are either not literate or have not been able to learn to read due to their disability-related challenges. Most of the proposed models are also trained on ASL datasets, the most prevalent and popular sign language. Other less popular and vernacular sign languages are being overlooked. Therefore, our solution tries to do justice to our national Indian Sign Language.

3.4 Proposed Methodology

The proposed method is suitable for hard-of-hearing people who sign in different languages. It focuses on deaf-to-deaf community conversation rather than the existing methods that deal with deaf-to-hearing communities and vice versa. The learning model is created in a way that considers the nuances and complexities of sign language interpretation. Phase one of the process involves creating our own ISL dataset of 30 words with 50 videos worth of data for each word. The dataset is split into training and testing sets. Eighty percent of the data is used for training, and 20 percent is used for testing. The model is trained on three different architectures, and the best one out of the three is chosen in the second phase. In the final phase, the output sentence is translated into ASL through GIFs, and the polarity of the sentence is determined. ASL and ISL were chosen for the model since ISL is our locally used sign language and due to the abundance of data available for ASL. This method can be used for any sign language based on the available resources.

3.4 Objectives

The primary purpose is to ensure that the translation of ISL signs into ASL is being done accurately while also capturing ASL's cultural and social nuances. Respect towards ISL users' cultural backgrounds should also be considered. The system attempts to make a seamless, real-time translation from ASL to ISL, facilitating natural and spontaneous conversations among users. Exact gifs should be shown for the word predicted, and the tonality of the sentence should also be predicted precisely at the end. By addressing these objectives, the system promotes understanding and inclusivity among individuals using different sign languages, ultimately contributing to a more accessible and connected society.

4. METHODOLOGY

4.1 Architecture

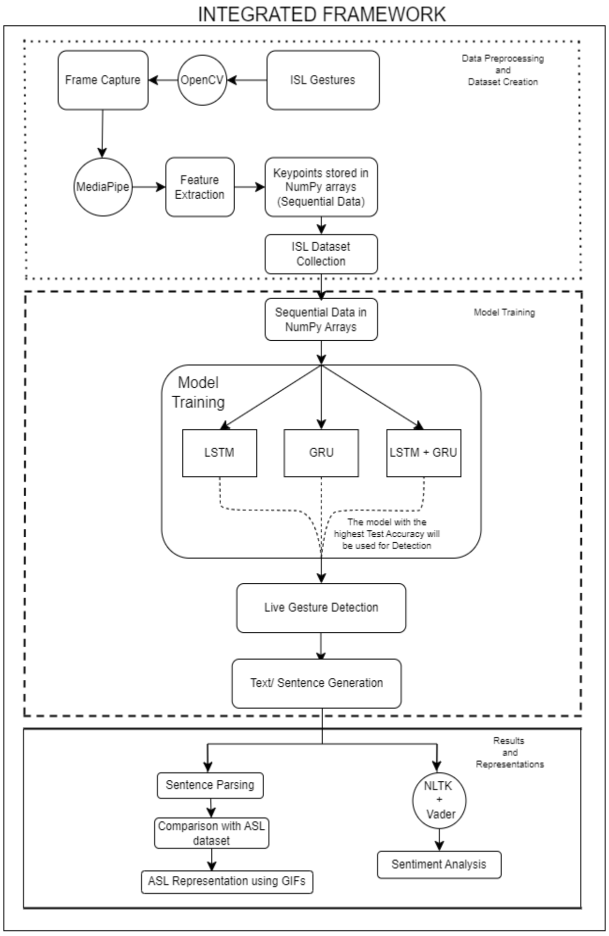

INTEGRATED FRAMEWORK
Data Preprocessing and Dataset Creation
ISL Gestures
OpenCV
Frame Capture
MediaPipe
Feature Extraction
Keypoints stored in NumPy arrays (Sequential Data)
ISL Dataset Collection
Model Training
Sequential Data in NumPy Arrays
Model Training
LSTM
GRU
LSTM + GRU
The model with the highest Test Accuracy will be used for Detection
Live Gesture Detection
Text/ Sentence Generation
Results and Representations
Sentence Parsing
Comparison with ASL dataset
ASL Representation using GIFs
NLTK + Vader
Sentiment Analysis

Fig -3: A integrated framework representing the conversion of ISL to ASL through GIFS and identifying the sentiments

4.2 Dataset Creation

OpenCV library is used to access live camera feed and to capture video data in the form of frames. Media Pipe model is used to detect all the landmarks and the obtained key-points are stored in NumPy arrays for respective frames. Sequential data is obtained by using this method and an ISL dataset is created. The list of words includes (Bathroom, Coffee, dislike, drink, eat, ExcuseMe, favorite, GoodMorning, GoodNight, happy, hello, HowAreYou, I, icecream, idontunderstand, like, meet, name, please, repeat, sad, sandwich, Sorry, Thanks, time, want, water, What, Where, you). There are a total of 30 words in the list. For each word, data worth of 50 videos is recorded during dataset creation with each video having 30 frames each.

4.3 Model Training

Once the dataset is created, it is used for training a deep-learning model. Since the data is sequential, RNN models are applied. Namely, LSTM and GRU. Three models have been trained over the given dataset - LSTM model, GRU model, LSTM+GRU model training Accuracy obtained across each of the 3 models is compared and the best one is chosen. The model with the highest training accuracy is used to implement the real-time detection. After importing the required libraries Sequential model is initialized.

The architecture of the stacked LSTM layers in addition to dense connection layers is responsible for hierarchical representation learning. For LSTM model, 3 LSTM layers are added. The first, second and third layers have 64, 128 and 64 units respectively. The 1st and 2nd layers return sequences which are given as input for the next layer except the 3rd layer. ReLU (Rectified Linear Unit) function used promotes non-linearity; neural networks with this activation can learn complex patterns easily. 3 dense layers are added after the LSTM layers to further process the extracted features for better classification. Adam optimizer is then used to compile the model and categorical_crossentropy is used as loss function. We choose the number of epochs for training the model and track the accuracy using TensorBoard. The architecture of the model is same for GRU, LSTM+GRU model as well, except in LSTM+GRU model the first and third layers consist LSTM and the third layer consists of GRU respectively.

4.3.1 LSTM

Long Short-Term Memory (LSTM) is a kind of recurrent neural network (RNN) architecture designed for action detection tasks, and therefore it is mostly suitable for actions such as sign language translation. It was introduced to address the vanishing gradient problem, which affects previous RNNs when training on long sequences of data. LSTM networks have memory cells and three gates: the input gate, the forget gate, and the output gate. These gates regulate the flow of information within the network, where it can choose which data to remember and what to forget. The input gate controls the flow of new information into the memory cell, the forget gate determines which information to remove from the cell's memory, and the output gate takes care of the information output from the cell.

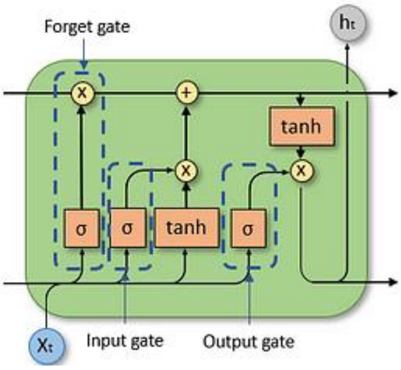
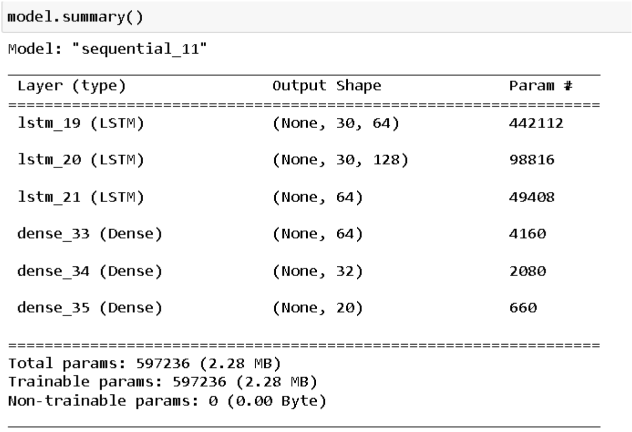


Fig -4: LSTM



```
model.summary()

Model: "sequential_11"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_19 (LSTM)              (None, 30, 64)            442112

 lstm_20 (LSTM)              (None, 30, 128)           98816

 lstm_21 (LSTM)              (None, 64)                49408

 dense_33 (Dense)            (None, 64)                4160

 dense_34 (Dense)            (None, 32)                2080

 dense_35 (Dense)            (None, 20)                660

=================================================================
Total params: 597236 (2.28 MB)
Trainable params: 597236 (2.28 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig -5: LSTM model summary

4.3.2 GRU

Gated Recurrent Unit (GRU) is a type of RNN architecture that is widely used for sequence modeling tasks, such as natural language processing and time series prediction. It was introduced as a simpler alternative to the LSTM architecture, designed to address some of the computational complexities of LSTM while maintaining similar performance. GRU has gating mechanisms that control the flow of information within the network, allowing it to capture long-range dependencies in sequences effectively. Unlike LSTM, which has separate memory cells and input/output gates, GRU combines these functionalities into a single update gate and reset gate. This simplification results in fewer parameters and computations compared to LSTM, making GRU models faster to train and often more efficient in terms of memory usage.

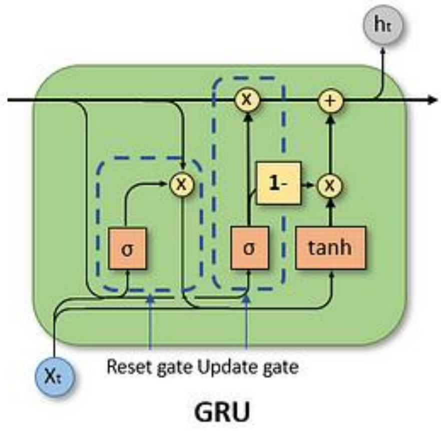
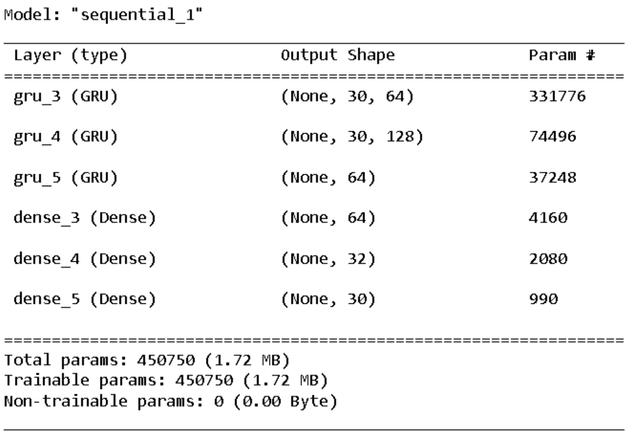


Fig -6: GRU

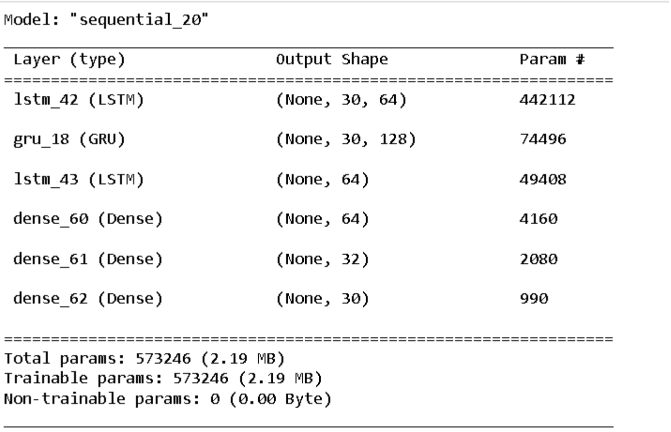


```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 gru_3 (GRU)                 (None, 30, 64)            331776

 gru_4 (GRU)                 (None, 30, 128)           74496

 gru_5 (GRU)                 (None, 64)                37248

 dense_3 (Dense)             (None, 64)                4160

 dense_4 (Dense)             (None, 32)                2080

 dense_5 (Dense)             (None, 30)                990

=================================================================
Total params: 450750 (1.72 MB)
Trainable params: 450750 (1.72 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig -7: GRU model summary

4.3.3 LSTM+GRU

This neural network implements the LSTMs and GRU layers together for sign language translation tasks. To this end, hybrid architecture can use the contextual relation of the gesture sequence well by the LSTM layer connected with the GRU layer. These repeating layers extract meaningful contents of sign language gestures as the translation's basis, increasing the truthfulness. Another LSTM layer is used to process the data, followed by more densely connected layers via the ReLU activation function, which allows for nonlinear learning. The model is taught to operate with the Adam optimizer, minimizing categorical cross-entropy loss. Category loss focuses on precision in the form of categorical accuracy during training. Another added feature is using TensorBoard to monitor training progress in real-time. After the training phase, the model summary section provides the details of its architecture and parameters.

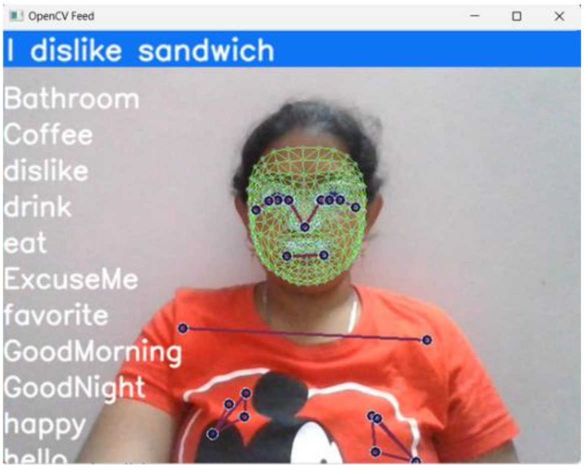


```
Model: "sequential_20"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_42 (LSTM)              (None, 30, 64)            442112

 gru_18 (GRU)                (None, 30, 128)           74496

 lstm_43 (LSTM)              (None, 64)                49408

 dense_60 (Dense)            (None, 64)                4160

 dense_61 (Dense)            (None, 32)                2080

 dense_62 (Dense)            (None, 30)                990

=================================================================
Total params: 573246 (2.19 MB)
Trainable params: 573246 (2.19 MB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Fig -8: LSTM+GRU model summary

4.4 Model Testing

The trained model undergoes real-time testing using OpenCV's live capturing feature. Upon initiation, an OpenCV window displays a list of words used to train the model. As the user begins making gestures in front of the camera, the system identifies and highlights the word that closely matches the gesture. In cases where the gesture corresponds to multiple words, all relevant words are partially highlighted. Additionally, a highlight bar accompanies each word, indicating the percentage accuracy of detection. A full bar denotes 100% accuracy, while a partially highlighted bar signifies partial accuracy or misidentification. This setup enables users to interact with the system seamlessly, despite potential misidentifications, by providing real-time feedback on gesture recognition accuracy.



I dislike sandwich

Bathroom
Coffee
dislike
drink
eat
ExcuseMe
favorite
GoodMorning
GoodNight
happy
hello

Fig -9: Real-time gesture recognition

4.5 Output and Representations

The final accuracy and performance metrics of the model are proven visually through a series of graphs and charts.

A TensorBoard graph presentations how the learning graph of the model in the course of education. Also, a confusion matrix is generated to recognize word identification accuracy, helping to perceive any kinds of erred category or confusion. In addition, the output text undergoes sentiment evaluation to understand the emotional tone of the message, whether or not it's positive, negative, or neutral. These results provide useful insights into the model's effectiveness and emotional effect, permitting similarly refinement and optimization for improved overall performance for users.

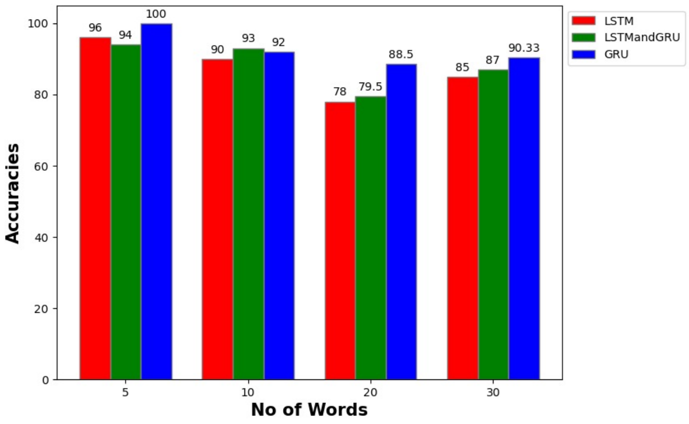


Fig -10: Bar-graph representing how the accuracies decrease with an increase in the number of words

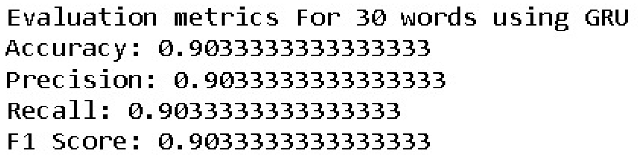
5. OVERVIEW OF TECHNOLOGIES

Machine learning and computer vision are the most important frameworks and libraries used in this work, which help in performing complex model construction, training and visualization. The main base used is TensorFlow: an open framework that supports designing ML models, including the LSTM and GRU networks for action detection and visceral data streams in particular. Being an end-to-end ecosystem, its flexibility allows users to fine-tune their models which drives advancement of numerous applications. Also, OpenCV functions as an integral component of computer vision. The tool performs multiple operations that include frame grabbing from camera images and videos, color space conversion, and feature superimposition. We have used OpenCV for collecting datasets in the form of frames for each dataset and at last we have used it for the detecting the asl signs.

The Scikit-learn features a wide range of machine learning tools that are great for data analysis and prediction because they offer classification, regression, clustering, and dimension reduction algorithms for sign language gestures and datasets. Matplotlib is in essence, the element bringing life to the visualization landscape in Python, presenting the ability to execute dynamic graph drawing coupled with diversified visualization opportunities to enhance the analysis and interpretation of data related to sign language gestures. TensorBoard is a tool which can be seen as a platform for traceability and visualization of model performance, real-time monitoring and visualization in model development and performance optimization are stressed. TensorBoard is used to enhance the analysis and interpretation of data related to sign language gestures. Collectively, these tools constitute the skeleton of the entire model's pipeline.

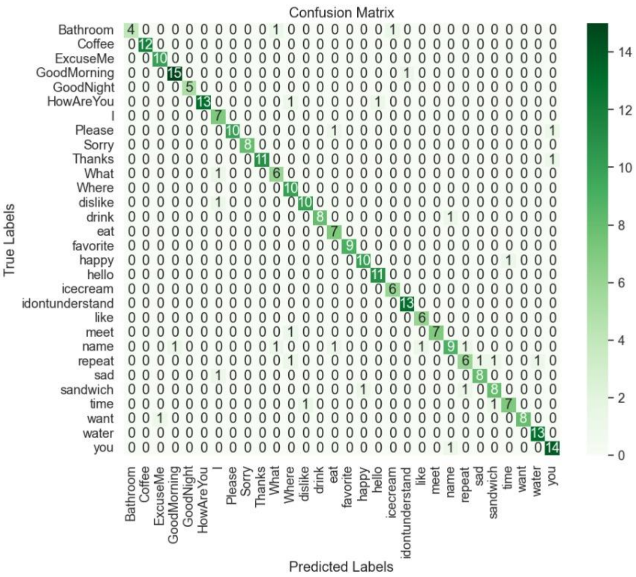
6. RESULTS AND DISCUSSIONS

The project was tested with various subsets of data. Initially the dataset was made of 30 words, but for checking the efficiency and performance of the model it has been split into 5, 10, 20 and 30 words respectively. These sets of words were trained on different neural network architectures namely — LSTM, LSTM+GRU, and GRU. After attentive trial and error method, we identified the architectures yielding the highest accuracies and used them to generate sentences from the input. These sentences were then transformed into ASL GIFs with the help of IPython libraries. Sentiment analysis was conducted using the VADER (Valence Aware Dictionary for sEntiment Reasoning) framework to determine the emotional tone of the sentences. The polarity of the sentiments—be it positive, negative, or neutral—was then determined, offering valuable insights into the conveyed messages.



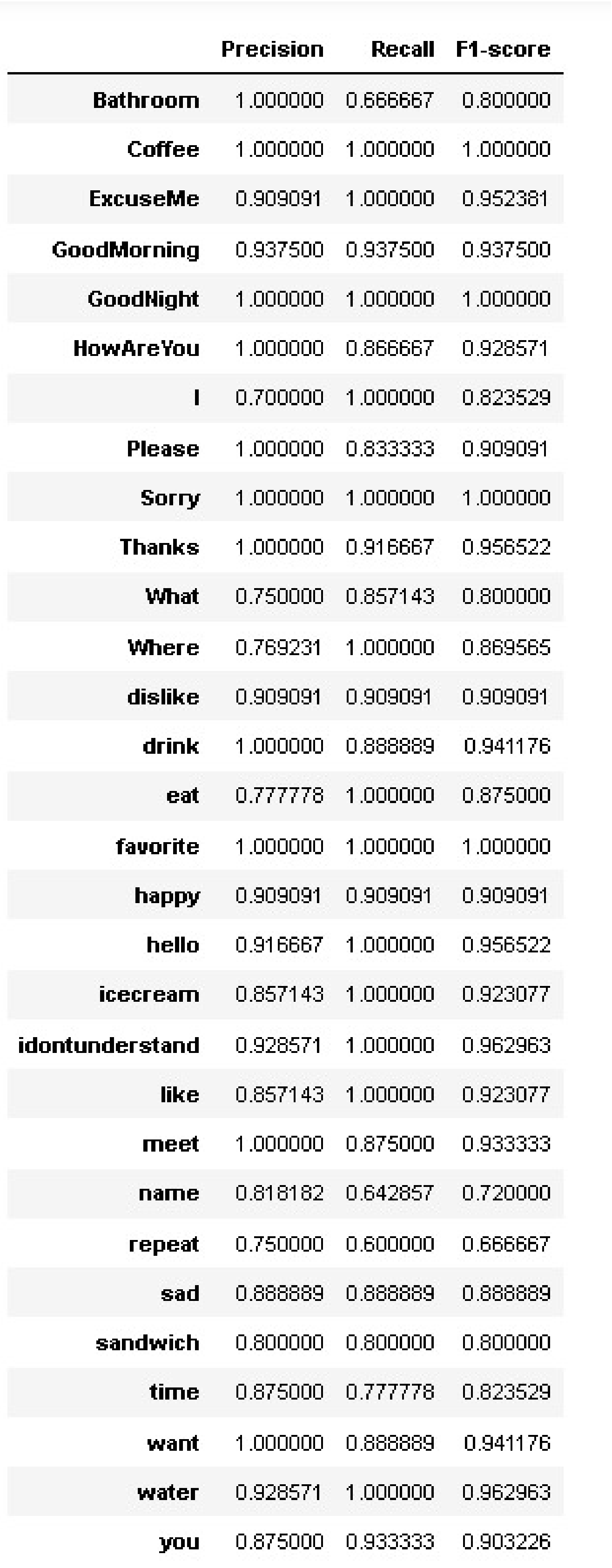
```
Evaluation metrics For 30 words using GRU
Accuracy: 0.9033333333333333
Precision: 0.9033333333333333
Recall: 0.9033333333333333
F1 Score: 0.9033333333333333
```

Fig -11: Evaluation metrics for 30 words using GRU



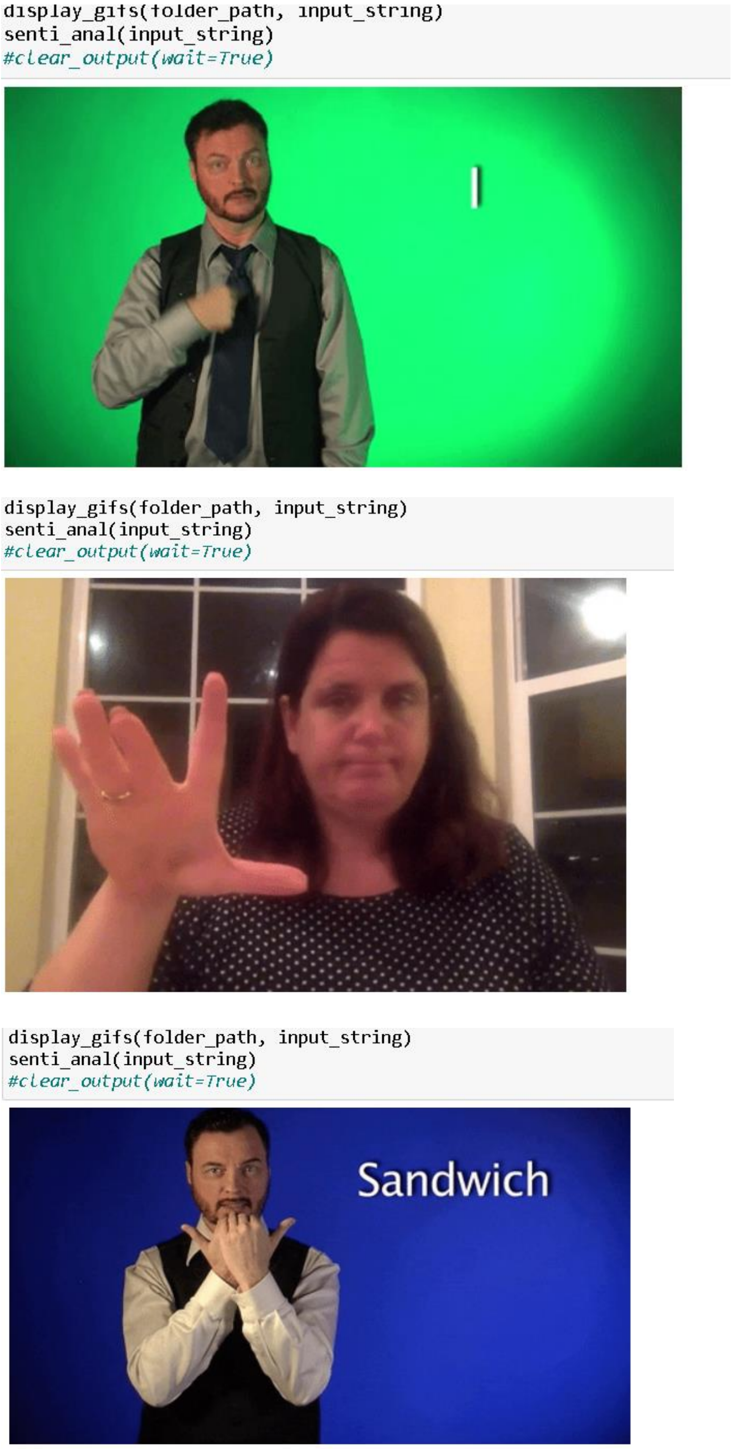
	Bathroom	Coffee	ExcuseMe	GoodMorning	GoodNight	HowAreYou	I	Please	Sorry	Thanks	What	Where	dislike	drink	eat	favorite	happy	hello	icecream	idontunderstand	like	meet	name	repeat	sad	sandwich	time	want	water	you
Bathroom	4	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
Coffee	0	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
ExcuseMe	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
GoodMorning	0	0	0	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
GoodNight	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
HowAreYou	0	0	0	0	0	13	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
I	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Please	0	0	0	0	0	0	0	10	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
Sorry	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Thanks	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
What	0	0	0	0	0	0	1	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Where	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
dislike	0	0	0	0	0	0	1	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
drink	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
eat	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
favorite	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
happy	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0	0	0	0	0	0	1	0	0	0
hello	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	0	0	0	0	0
icecream	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0
idontunderstand	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0	0	0	0	0	0	0	0	0	0
like	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0
meet	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0
name	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	1	0	9	1	0	0	0	0	0	0
repeat	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	6	1	1	0	0	1	0
sad	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	0
sandwich	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	8	0	0	0	0
time	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	7	0	0	0
want	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	0	0
water	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13	0
you	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	14

Fig -12: Confusion matrix for 30 words using GRU



	Precision	Recall	F1-score
Bathroom	1.000000	0.666667	0.800000
Coffee	1.000000	1.000000	1.000000
ExcuseMe	0.909091	1.000000	0.952381
GoodMorning	0.937500	0.937500	0.937500
GoodNight	1.000000	1.000000	1.000000
HowAreYou	1.000000	0.866667	0.928571
I	0.700000	1.000000	0.823529
Please	1.000000	0.833333	0.909091
Sorry	1.000000	1.000000	1.000000
Thanks	1.000000	0.916667	0.956522
What	0.750000	0.857143	0.800000
Where	0.769231	1.000000	0.869565
dislike	0.909091	0.909091	0.909091
drink	1.000000	0.888889	0.941176
eat	0.777778	1.000000	0.875000
favorite	1.000000	1.000000	1.000000
happy	0.909091	0.909091	0.909091
hello	0.916667	1.000000	0.956522
icecream	0.857143	1.000000	0.923077
idontunderstand	0.928571	1.000000	0.962963
like	0.857143	1.000000	0.923077
meet	1.000000	0.875000	0.933333
name	0.818182	0.642857	0.720000
repeat	0.750000	0.600000	0.666667
sad	0.888889	0.888889	0.888889
sandwich	0.800000	0.800000	0.800000
time	0.875000	0.777778	0.823529
want	1.000000	0.888889	0.941176
water	0.928571	1.000000	0.962963
you	0.875000	0.933333	0.903226

Fig -13: Evaluation metrics for each of the 30 words

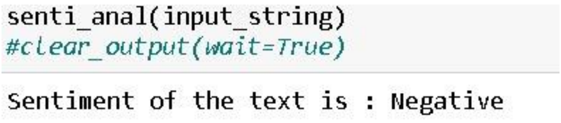


```
display_gifs(folder_path, input_string)
senti_anal(input_string)
#clear_output(wait=True)
```

```
display_gifs(folder_path, input_string)
senti_anal(input_string)
#clear_output(wait=True)
```

```
display_gifs(folder_path, input_string)
senti_anal(input_string)
#clear_output(wait=True)
```

Fig -14: ASL representation of words using GIFs



```
senti_anal(input_string)
#clear_output(wait=True)
```

```
Sentiment of the text is : Negative
```

Fig -15: Sentiment analysis result

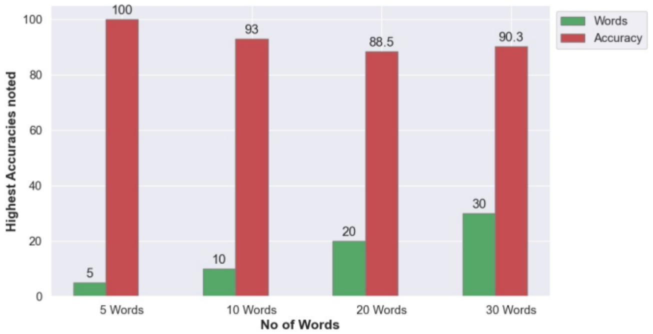


Fig -16: Bar graph representing the no. of words and highest accuracy obtained for each case

7. CONCLUSION & FUTURE SCOPE

From our comprehensive analysis, it's evident that the accuracy of our models is intricately linked to the number and nature of words considered for training. Different word types exhibit varying accuracies, underscoring the importance of meticulous data curation. Smooth training processes and high accuracies are observed for words with consistent data representations. Furthermore, the choice between LSTM, LSTM+GRU, or GRU architectures significantly influences both training speed and final accuracy. Notably, GRU requires the fewest parameters, while LSTM typically demands more. The inclusion of more words leads to a decline in final accuracy, attributed to the increased data volume and complexity of model parameters. Despite leveraging extensive datasets comprising 50 videos with 30 frames each, our models still achieve commendable accuracy values.

Looking ahead, the integration of both static and dynamic data presents an exciting avenue for further advancement. Incorporating NLP to further refine the sentence structures for a more seamless translation between languages also serves as a key point for future development. This holistic approach sets the stage for continued refinement and innovation in communication accessibility and sentiment analysis domains.

REFERENCES

[1] Sundar, B & Thirumurthy, Bagyammal. (2022). American Sign Language Recognition for Alphabets Using MediaPipe and LSTM. Procedia Computer Science. 215. 642-651. 10.1016/j.procs.2022.12.066.

[2] Bora, Jyotishman & Dehingia, Saine & Boruah, Abhijit & Chetia, Anuraag & Gogoi, Dikhit. (2023). Real-time Assamese Sign Language Recognition using MediaPipe and Deep Learning. Procedia Computer Science. 218. 1384-1393. 10.1016/j.procs.2023.01.117. M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[3] S. Adhikary, A. K. Talukdar and K. Kumar Sarma, "A Vision-based System for Recognition of Words used in Indian Sign Language Using MediaPipe," 2021 Sixth International Conference on Image Information Processing (ICIIP), Shimla, India, 2021, pp. 390-394, doi: 10.1109/ICIIP53038.2021.9702551.

[4] Akshatharani, B. K., & Manjanaik, N. (2021). Sign language to text-speech translator using machine learning. International Journal of Emerging Trends in Engineering Research, 9(7).

[5] Sheth, Pranav & Rajora, Sanju. (2023). Sign Language Recognition Application Using LSTM and GRU (RNN). 10.13140/RG.2.2.18635.87846.

[6] Tayade, Akshit & Halder, Arpita. (2021). Real-time Vernacular Sign Language Recognition using MediaPipe and Machine Learning. 10.13140/RG.2.2.32364.03203.

[7] Khartheesvar, G & Kumar, Mohit & Yadav, Arun Kumar & Yadav, Divakar. (2023). Automatic Indian sign language recognition using MediaPipe holistic and LSTM network. Multimedia Tools and Applications. 1-20. 10.1007/s11042-023-17361-y.

[8] Subramanian, B., Olimov, B., Naik, S.M. et al. An integrated mediapipe-optimized GRU model for Indian sign language recognition. Sci Rep 12, 11964 (2022). https://doi.org/10.1038/s41598-022-15998-7

[9] Cristina Luna-Jiménez, Manuel Gil-Martín, Ricardo Kleinlein, Rubén San-Segundo, and Fernando Fernández-Martínez. 2023. Interpreting Sign Language Recognition using Transformers and MediaPipe Landmarks. In Proceedings of the 25th International Conference on Multimodal Interaction (ICMI '23). Association for Computing Machinery, New York, NY, USA, 373–377. https://doi.org/10.1145/3577190.3614143

[10] Chandwani, Laveen & Khilari, Jaydeep & Gurjar, Kunal & Maragale, Pravin & Sonare, Ashwin & Kakade, Suhas & Bhatt, Abhishek & Kulkarni, Rohan. (2023). Gesture based Sign Language Recognition system using Mediapipe. 10.21203/rs.3.rs-3106646/v1.

[11] Duy Khuat, B., Thai Phung, D., Thi Thu Pham, H., Ngoc Bui, A., & Tung Ngo, S. (2021, February). Vietnamese sign language detection using Mediapipe. In Proceedings of the 2021 10th International Conference on Software and Computer Applications (pp. 162-165).

[12] Goyal, K. (2023). Indian sign language recognition using mediapipe holistic. arXiv preprint arXiv:2304.10256.

[13] https://www.researchgate.net/figure/Indian-sign-language-for-numbers-and-alphabets_fig1_357169928

[14] https://images.app.goo.gl/hoCvmmorVRTXQmBk7

[15] https://developers.google.com/mediapipe

[16] https://towardsdatascience.com/a-brief-introduction-to-recurrent-neural-networks-638f64a61ff4