Sign Language Interpreter using Deep Learning Techniques

[1]Vedanth P Bhar adwaj, [2]Ananya B, [3]Ms. Yashaswini B V

[1,2]Student, [3]Assistant Professor
Information Science and Engineering Department,
BNM Institute of Technology, Bengaluru, India

***Abstract*: The primary goal of man is to progress and grow in the field he chooses and communication is one of the key aspects for this growth. As we are aware that some of the people in this world are unable to speak/hear to convey their thoughts and feelings. The purpose of this system was to make a way for the common man to understand what the mute are trying to convey. One solution could be the presence of an interpreter but his is highly impractical in daily life. There has been many research work which have been conducted in this field but most of them have developed a system which captures the isolated images of hands. The system that has been proposed here is implemented with the MediaPipe holistic framework which upon capturing the videos of the signs, extracts the skeletal keypoints of the face, pose, left and right hands which is very helpful to collect more accurate data. Adding to this the LSTM model has helped in processing and classifying the short videos of signs and OpenCV module has been crucial to capture all the videos required for the system. It was found that this system, might be quite slow to recognize the signs given a larger dataset, it was more accurate as the background lights, colours of skin getting merged with the background did not matter as it was the case with some of the other systems. For the dataset that was trained, the accuracy was around 97% and this system has a great scope for improvement in the future for recognizing many signs and can also be made into an application to the mobile phones.**

***Keywords*: Indian Sign Language (ISL), Mediapipe, LSTM and OpenCV.**

I. INTRODUCTION

According to world health organization over 1 billion young adults are at a risk of permanent hearing loss due to unsafe listening practices and over 5% of the world's population – or 430 million people – require rehabilitation to address their 'disabling' hearing loss (430 million adults and 34 million children). Expertising human motions may be posed as a sample popularity trouble. If a computer can detect and distinguish the human motion patterns(signs/gestures), the desired message can be constructed. Human beings whose hearing aids are absolutely or partly damaged are termed as deaf. To speak with human beings with proper hearing, the deaf use lip-studying or sign language as a language that employs signs and symptoms made with the arms and different actions, inclusive of facial expressions and postures of the body. Communication is the process of exchanging ideas or messages with other people through gestures or text. The sign language is a powerful method of communicating our thoughts to others. Sign language is the daily language of communication between deaf and dumb people, which is the most comfortable and natural way of communication between deaf and dumb people, and is also the main tool for special education schools to teach and convey ideas.
The project titled Sign Language Interpreter using deep learning techniques, aims to design a sign language interpreter / detector and which can ultimately convert it into text. In the project MediaPipe framework and LSTM is used. The LSTM is an algorithm which helps in predicting the actions and is a part of the RNN model. Our system recognises gestures from The Indian Sign Language (ISL) in real time. To bridge the space between the ordinary human beings and listening to impaired people, person one ought to recognize sign language. We aim to expand an efficient system to assist recover from this barrier of conversation. A functioning sign language interpreter can provide an opportunity for a muted to communicate to a non-signer without the help of an interpreter.

II. OBJECTIVE AND SCOPE

Sign language is a visual language that makes use of hand gesture, exchange of hand form and track statistics to specific meaning and is the principle verbal exchange tool for humans with hearing and language impairment. This system will be proposed to recognize the hand gestures using a Deep Learning Algorithms, Computer Vision and LSTM (Long Short-Term Memory) to process the image and predict the gestures. By identifying the gestures, the system will be converting it into real time text.
The main scope includes as India has around 6.3 crore hearing-impaired people and of these, at least 50 lakhs are children. The way in which they converse with people is by using sign-language. Sign Language is not known to many, hence making it extremely difficult for the deaf and dumb to express their thoughts to people. One solution to this problem is having a person as an interpreter, but it becomes impractical when it comes to cost and convenience. The solution that is proposed is the "Sign-Language Interpreter using Deep Learning Techniques", a system that helps understand the sign-language, interprets the same and helps people know what the mute are trying to express.

III. RELATED WORK

A literature survey in a assignment report represents the take a look at completed to help in the final touch of a challenge. A literature survey also describes a survey of the preceding existing cloth on a topic of the record. A survey of related literature refers to a study done earlier than or after selecting a studies problem to recognize about the preceding studies paintings, ideas, theories, strategies, strategies, troubles going on during the research, and so forth. There is a significant work related in the field of Sign

Language Recognition. There are many different models built for recognizing the signs, with different methods such as using gloves or sensors, camera, motion accelerator which are used to recognize the movement of hands. A research work is done to build an efficient system that is robust than the rest. **Shaik Khadar Sharif. *et. al.*, [1]** proposed a system where CNN models were created for picture gathering, in which the model recognizes two-dimensional data addressing an image's pixels and concealing channels, in a strategy called highlight learning. This equal method can be applied to one-dimensional courses of action of data. An exceptionally new area that become attempting to triumph over the out-of-date problems is Human PC interactions which are accomplished with the aid of AI and profound gaining knowledge. **Mehreen Hurroo, *et. al.*, [2]** introduces a Sign Language recognition using American Sign Language. In this, the user must be able to capture images of the hand gesture using web camera and the system shall predict and display the name of the captured image. Used the HSV colour algorithm to detect the hand gesture and set the background to black. Usage of technique called Computer vision where the images undergo a series of processing steps. Also made use of CNN for training and to classify the images. **Shubhendu Apoorv, *et. al.*, [3]** proposed a system which was reliable communication interpretation program for interpreting Indian Sign Language and converting it to a reliable output. The task is carried out using photo processing and device mastering. Proposed project can find its applicability within the every day for communication, it may also work for gaining knowledge of various gestures in gesture based automatic structures. **Anshul Mittal, *et. al.*, [4]** has proposed a modified LSTM model for continuous sequences of gestures or continuous SLR that recognizes a sequence of connected gestures. It is primarily based on splitting of continuous symptoms into sub-gadgets and modelling them with neural networks. The proposed machine has been examined with 942 signed sentences of Indian Sign Language (ISL). **Gautham Jayadeep, *et. al.*, [5]** the main goal of the proposed method is to design an ISL (Indian Sign Language) hand gesture motion translation tool for helping the deaf-mute community to convey their ideas by converting them to text format. Proposed technique recognizes human movements thinking about remoted dynamic Indian symptoms associated with the bank as a unique approach. They used a self-recorded ISL dataset for training the model for recognizing the gestures. Larger lengthened video gestures were taken and actions have been recognized from a sequence of video frames. **Siming He, *et. al.*, [6]** presented an approach for hand locating and sign language recognition of common sign language based on neural network, and the main research contents include: 1. A hand locating network based on the Faster R-CNN 2. A three-D CNN function extraction network and a sign language popularity framework based totally on lengthy- and short-time memory (LSTM). **Saurabh Kumbhar, *et. al.*, [7]** proposed a system for sign language recognition of alphabets and numbers based on CNN. The technique used is CNN because it increases the accuracy of the system by recognizing hidden patterns and correlation in raw data. **Yuancheng Ye, *et. al.*, [8]** proposed a novel hybrid model, 3D recurrent convolutional neural networks, to recognize American Sign Language (ASL) gestures and localize their temporal boundaries within continuous videos, by fusing multi-modality features. The proposed 3DRCNN model integrates 3D convolutional neural network and enhanced fully connected recurrent neural network, where 3DCNN learns multi-modality features from RGB, motion, and depth channels, and FCRNN captures the temporal data amongst quick video clips divided from the unique video. The existing systems were studied as a part of the literature survey and many things were learnt from them.

• Many systems use 2D or 3D images as the dataset and some even use depth sensors for extracting features from the gestures.
• Some systems are dependent on the contrasting colours of the skin and the background in order to different the hands from its background. These systems capture only the hand movements which needs to be recorded separately unlike the real-world use case.
• Dataset used by many systems is limited to 26 or 36 classes. That is, 26 alphabets along with 10 digits, but these cannot be used for extensive purposes.

The above-mentioned points were some of the shortcomings of the systems existing currently and we hope to bring out a system which performs better than the existing ones.

IV. METHODOLOGY

Live perception of simultaneous human pose, face landmarks, and hand tracking in real-time on any devices which can enable various modern life application such as gesture control and sign language recognition, augmented reality try-on and effects. MediaPipe already gives rapid and correct, but separate, answers for these tasks. Combining all of them in real-time into semantically constant cease-to-give up solution is a uniquely tough hassle requiring simultaneous inference of multiple, established neural networks. The MediaPipe Holistic pipeline integrates separate fashions for pose, face and hand additives, each of which can be optimized for their precise domain. The pipeline is carried out as a MediaPipe graph that uses a holistic landmark subgraph from the holistic landmark module and renders the use of a committed holistic renderer subgraph. The holistic landmark subgraph internally uses a pose landmark module, hand landmark module and face landmark module.

Fig 1: Glimpse of Mediapipe Feature Extraction of Hands

Fig 2: Mediapipe Feature Extraction of Face

LSTM also plays a major role in processing the data, long short-term memory (LSTM) is an artificial recurrent neural network (RNN) architecture used in the field of deep learning. Unlike standard feedforward neural networks, LSTM includes feedback connections. A commonplace LSTM unit consists of a cellular, an enter gate, an output gate and an overlook gate. A cellular remembers values over arbitrary time durations and the three gates adjust the go with the flow of records into and out of the cellular. LSTM networks are well-suited to classifying, processing and making predictions based on time series data, since there can be lags of unknown duration between important events in a time series. An RNN using LSTM units can be trained in a supervised fashion on set of training sequences, using an optimization algorithm like gradient descent combined with backpropagation through time to compute the gradients wanted throughout the optimization technique.

OpenCV i.e., computer vision is a process by which we can understand the images/videos how they are stored and also, how can we manipulate and retrieve data from them. OpenCV is the massive open-supply library for the computer imaginative and prescient, machine learning, and photo processing and now it performs a prime function in actual-time operation which could be very crucial in today's systems. To identify image pattern and its various features we use vector space and perform mathematical operations on these features.

V. IMPLEMENTATION

The primary step of any proposed system is to collect the dataset. In many existing systems the data is collected using sensors or pre-recorded standard set of images as dataset. Whereas for our system, we made use of the web camera to record the signs. We first installed and imported six dependencies which are : TensorFlow, TensorFlow-gpu, OpenCV-python as it is an open computer vision library that enables us to work with the web cameras and makes it easier to build our feed and extracts the keypoints by accessing the web camera, then we make use of mediapipe holistic to extract the keypoints and saving it as frames, also imported scikit-learn (sklearn) for evaluation matrix as well as to leverage a training and testing split and finally, we used matplotlib which helps us to visualize images easier. The dataset is collected by recording signs/gesture in the form of very short videos which are converted to multiple frames. Then, keypoints are drawn which are then converted to numpy arrays. We collect keypoints for training and testing. The next step was data pre-processing wherein we label the data, later we go ahead and train our LSTM model for this we have used TensorFlow specifically keras. Predicting the designed model is the next step, in this we passed our test data and by giving the values it detects the right result. The next step is testing the model in real time where we test if the accuracy is good enough. If the accuracy is good then we stop, otherwise we again pre-process the data collected, train the LSTM model and lastly testing is done.

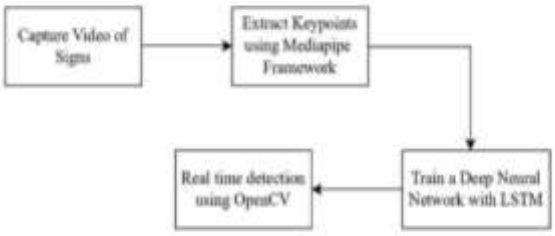

Fig 3: System Architecture

A. Dataset

We have created a dataset of Indian Sign Language of 15 signs. As it's a video dataset we have collected 30 videos per action. Then each one of those video sequences are going to contain 30 frames of data. Each frame contains 1662 landmark values. There are basically four landmarks which are face, pose, lefthand and righthand. The data is being collected using a media pipe loop, as we loop through the actions and collect set of frames/videos. We developed a code to recognize the videos of the dataset when the gesture is positioned in at different distance/position from the camera. We used OpenCV in our project to capture the video frames and to save them in a prescribed path. Computer vision is a process by which we can understand the images and videos how they are stored and how we can manipulate and retrieve data from them. Computer vision is the base for Artificial Intelligence. By using OpenCV, we can process videos and images (videos in our case) to identify objects, faces, or even handwriting of a human, but in our system OpenCV identifies 4 landmarks as discussed above. When it's far incorporated with diverse libraries, which includes NumPy, python is able to process the OpenCV array structure for evaluation. The 15 basic signs that we have created are

'Hello', 'Namaste', 'Good', 'Thank You', 'India', 'Yes', 'No', 'Time', 'Food', 'Strong', 'What', 'Why', 'Face', 'Teacher' and 'Hearing'. All these basic words are recorded based on the Indian Sign Language (ISL) standards.

B. Extract Keypoints

After detecting face, hand and pose landmarks we extract the keypoint values into a format and access the components to grab our face landmarks, left hand landmarks, right hand landmarks and pose landmarks. In order to extract the keypoints we first have concatenated into a numpy array, if there is no value in time, we just create a numpy zero array so that an array with the same shape we create the numpy array with value zero and later, going to substitute that in. In this section of our system design we extracted a set of landmarks for one of our keypoints, and after that we extract for each different landmark. Next step was to create a holder or placeholder array to find the length of a particular landmark.

C. Data Preprocessing

After collecting the keypoint sequences we pre-process the data and creating labels and features. To do that we imported a couple of additional dependencies specifically train test split from scikit learn, where it allows us to create a training and testing partition and also, we imported two categorical functions from keras utilities. After importing the train test split function allows us to partition our data into a training partition in a testing partition, ultimately it allows us to train on one partition and test it out on a different segment of our data and then the two categorical function is useful when we have to convert the data into one big encoded data. Next, we created a label map which is a label array or label dictionary to represent each one of our different actions, the dictionary contains the word (from the dataset) and each word in the dictionary is set to an id. These ids are used when the training and testing data are created, for achieving that we create a segment or a set of labels which represents those ids set to each word in the dictionary. Finally, we bring all of our data together and structuring it. We have previously collected all our different keypoint sequences which will be structured, there are 1662 values per sequence so, for this we create a big array which actually contains all our data. So effectively we end up having 90 arrays with 30 frames in each one of those arrays with 1662 values which represents our keypoints. After structuring we loop through each sequence of frames for processing the data. We have included 75% of the data for the training purpose and the remaining 25% for testing.

D. Train LSTM Model

After data preprocessing and labelling, we went ahead and trained our LSTM (long short-term memory) model, for this we have used TensorFlow specifically Keras. In this part we import certain key dependencies such as sequential model, LSTM layer and dense layer. In which LSTM layer gives the temporal component to build our neural network and allows us to perform action detection. The primary activation function used for the LSTM layer is ReLu. And for the Dense Layer it was Softmax. The model was run for around one thousand epochs in order to get a better accuracy. This model is saved for future use and for better accuracy.

```
1 model.summary()
```

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm_3 (LSTM)                (None, 30, 64)            442112
_________________________________________________________________
lstm_4 (LSTM)                (None, 30, 128)           98816
_________________________________________________________________
lstm_5 (LSTM)                (None, 64)                49408
_________________________________________________________________
dense_3 (Dense)              (None, 64)                4160
_________________________________________________________________
dense_4 (Dense)              (None, 32)                2080
_________________________________________________________________
dense_5 (Dense)              (None, 15)                495
=================================================================
Total params: 597,071
Trainable params: 597,071
Non-trainable params: 0
_________________________________________________________________
```

Fig 4: Model Summary

Fig 4. Describes the model summary with total parameters, trainable parameters and non-trainable parameters.

E. Real Time Detection

The model which was saved to the hard-disk is loaded. Then, using OpenCV we test the model by capturing the signs in real-time. The mediapipe points are drawn and are compared to the model that is already saved and the closest match is found which is displayed as text. In this part we have created two detection variables which are sequence and sentence. Sequence will collect our 30 frames in order to be able to generate a prediction. As we loop through the frame using OpenCV we will be appending the sequence and once we get the 30 frames, we pass it to the prediction algorithm to start our predictions. Whereas, sentence variable helps us to concatenate the history of detection together. This will help us to create a sentence by combining multiple words.

VI. RESULTS AND DISCUSSION

We have got the following results after implementing the code by providing the real-time input to the built model. We developed a model by applying deep learning approach that is used to translate the videos which are recorded that include face, left hand, right hand and pose to text. We have designed a simple user-friendly GUI, that would help user to operate for translating/communicate with one another. We faced a major challenge of having small dataset consisting Indian Sign Language, as it required high computing power. Our proposed model achieved an accuracy of around 97%. Used modules such as cv2 and TensorFlow. Implementation is done with the help of specific functions such as LSTM, OpenCV and Mediapipe. As discussed, earlier Mediapipe played a major role in processing time series data like video and LSTM helped in building the neural network and performed action detection. We were able to achieve a good accuracy and through video dataset we got a chance to make our sign language interpreter clear and high in functioning. Compared to few previous research work which we had studied, it was observed that image dataset was used which was limited and they also came across limitation of being dependent on the contrasting colours of the skin and background in order to differentiate the hands from its surrounding. Some of these systems captured only the hand movements which required to record it separately unlike the real-world use case. But through our proposed system we were capable of recording the videos without having any limitation of the background and colours of the skin. We were also able to capture some of the facial expressions which had an impact on the signs. The system that has been developed gives the output as a text and it is appended at the end of each word/sign. Hence, the last word that appears is the sign that is being depicted.

Fig 5: Output of the system.

VII. CONCLUSION

The main objective of our proposed system is to bridge the communication gap between the mute and normal person. We achieved this objective by building a Sign Language Interpreter using deep learning techniques having Indian Sign Language as the data. Our application is beneficial to a certain section of the society. As most people are unable to understand what mute are trying to convey, our proposed system converts the video dataset to real time text through a web camera which is helpful for the mute and the common people to communicate. Limitations from our study is to implement with large dataset, it requires high end systems for processing which are quite expensive.

VIII. FUTURE SCOPE

In the implementation of the project LSTM and dense layers were used, in which LSTM layer gives the temporal component to build our neural network and allowed us to perform action detection. Sequential model was also imported as key dependency. TensorFlow was used that allows neural networks to go hand in hand, which is predominantly used for prediction and classification of the sample data fed into the model or the network. We collected dataset of 15 words, which is a video dataset recorded through

web camera and converted the recognized gesture into real-time text. The dataset recorded was limited as it required high processing systems. We did not have any limitation on background lighting and dependence on contrasting colours of the skin. Future enhancement to this model is, the system can be built using large video dataset with high processing system that can recognize the video dataset and convert it into real time audio, which can ultimately help mute and blind people to communicate.

REFERENCES

[1] Shaik Khadar Sharif, Chava Sri Varshini, "Sign Language Recognition", Vol. 9, May- 2020, Hyderabad-500090, Telangana, India .
[2] Mehreen Hurroo , Mohammad Elham Walizad, " Sign Language Recognition System using Convolutional Neural Network and Computer Vision ", IJERT, Vol. 9 Issue 12, December- 2020, 406-415, Delhi, India.
[3] Shubhendu Apoorv, Sudharshan Kumar Bhowmick," Indian sign language interpreter using image processing and machine learning ", IOP Conference, Vol.7,2020, Issue 8, Chennai, India.
[4] Siming He, "Research of a Sign Language Translation System Based on Deep Learning", 2019 AIAM Conference, Canada
[5] Anshul Mittal, Pradeep Kumar an Team, "A Modified-LSTM Model for Continuous Sign- Language Recognition using Leap Motion", 2019 IEEE Sensors Journal, India
[6] Yuancheng Ye, Yingli Tian, Matt Huenerfauth, and Jingya Liu, "Recognizing American Sign Language Gestures from within Continuous Videos", CVF, IEEE Xplore, USA
[7] Saurabh Kumbhar, Abhishek Landge, Akash Kulkarni, Devesh Solanki, Vidya Kurtadikar, "Indian Sign Language Recognition System", Vol 6, Issue 6, pp 1375 1377, June-2021, IJISRT
[8] Gautham Jayadeep, Vishnupriya N V, Vyshnavi Venugopal, Vishnu S, Geetha M, "Mudra: Convolutional Neural Network based Indian Sign Language Translator for Banks", IEEE 2020, pp 1228-1232, ICICCS.