**PROJECT REPORT**

4/15/2024

***YUVAM KOTHARI*** (229309251)

***EESHAN BABLANI***(229309252)

***TUSHAR GARG***(229309270)

***LOCHAN MANGAL***(229309274)

**DIGITAL INTERPRETOR**

ISL TO TEXT RECOGNITION

1. **INTRODUCTION**

In our project, we address the significant communication barrier faced by the deaf and hard-of-hearing community due to limited effective sign language interpretation tools. This barrier restricts their access to essential services like education, employment, and healthcare, leading to social exclusion and reduced quality of life. To tackle this pressing issue, we have developed an Indian Sign Language (ISL) program using Long Short-Term Memory (LSTM) technology.

Our approach leverages LSTM, a type of recurrent neural network capable of recognizing and interpreting sequential data, making it ideal for capturing the temporal dynamics of sign language gestures. By harnessing LSTM's capabilities, our program aims to accurately interpret ISL gestures in real-time and translate them into text or spoken language, facilitating effective communication for the ISL community.

In the context of related work, our project aligns with ongoing efforts in the field of sign language recognition and translation using machine learning and artificial intelligence technologies. However, our focus on ISL and the use of LSTM technology allows us to create a more tailored and accurate solution that addresses the unique linguistic and cultural aspects of Indian sign language. By building upon existing research in gesture recognition, machine learning, and assistive technologies, our project contributes to the broader goal of enhancing accessibility and inclusivity for the deaf and hard-of-hearing community globally.

1. **RELATED WORK**

Our work has been inspired from the research paper published by two scholars named Yogeshwar I. Rokade and Prashant M. Jadav from Department of Computer Engineering, Dharmsinh Desai University,Gujrat.

In their problem they have defined a method is proposed for the automatic recognition of the finger spelling in the Indian sign language. Here, the sign in the form of gestures is given as an input to the system. Further various steps are performed on the input sign image. Firstly segmentation phase is performed based on the skin color so as to detect the shape of the sign. The detected region is then transformed into binary image. Later, the Euclidean distance transformation is applied on the obtained binary image. Row and column projection is applied on the distance transformed image. For feature extraction central moments along with HU’s moments are used. For classification, neural network and SVM are used.

In our problem , we have defined certain words and phrases instead of letters and alphabets and we have solved our problem using LSTM for real time sensing and testing the model , Sklearn machine learning model for collection and cleaning of data ; and KERAS, our main model in which training and testing has been there for the acquired dataset . Finally ,we have used MEDIAPIPE, a python library which uses real time hand detection using key points.

Our problem and method is better as there is a real usage of the sign language. Moreover where there model is converting the letters and alphabets into phases we are directly training the model with phrases so that it can understand easily and in faster and efficient way. Our model also have deep learning techniques which is absent in the proposed model of our project’s aquintance.

1. **ALGORITHIM & METHODOLOGY**
   1. ALGORITHIM

The algorithm of our interpetor works in a simple manner .First and foremost , importing and installing all the dependencies which includes PYTHON 3.7.9 , KERAS , MEDIA PIPE , OPENCV , MATPLOTLIB , NUMPY & SCIPY ,OS and TIME module. Then forming of the landmarks using Media pipe holistic method. After that extraction of the key points and setting up the data collection folders is our next step. Furthermore , Collected data is now treated as training and testing using SKLEARN ML MODEL. Stepping up , forming the LSTM model using KERAS , making predictions and evaluations is done in favor of accuracy using confusion matrix. Testing in real time is our Final step.

3.1.1PSEUDO CODE

Import: PYTHON 3.7.9, KERAS, MEDIA PIPE, OPENCV, MATPLOTLIB, NUMPY, SCIPY, OS, TIME

Function extract\_landmarks(): Capture video frames, Detect landmarks for each frame, Return landmarks

Function extract\_key\_points(landmarks): Extract key points from landmarks, Return key points

Function setup\_data\_collection\_folders(): Create folders for data collection, Split folders for training and testing, Return folder paths

Function collect\_data():

For each folder in data collection folders: Capture video frames, Extract landmarks using extract\_landmarks(), Extract key points using extract\_key\_points(), Save key points data in corresponding folder

Function train\_test\_model(data\_paths):

Using SKLEARN ML MODEL: Load data from data\_paths, Split data into training and testing sets, Train the model, Evaluate model accuracy using confusion matrix, Return trained model and evaluation results

Function build\_LSTM\_model(): Using KERAS: Define LSTM model architecture, Compile the model, Return LSTM model

Function make\_predictions(model, test\_data): Using trained LSTM model: Make predictions on test data, Return predictions

Function real\_time\_testing(model): Capture real-time video frames, Extract landmarks using extract\_landmarks(), Extract key points using extract\_key\_points(), Make predictions using make\_predictions(), Display predictions in real-time

Main Code:

Call setup\_data\_collection\_folders() to setup data collection folders

Call collect\_data() to collect and save data

Call train\_test\_model() to train and test SKLEARN ML MODEL

Call build\_LSTM\_model() to build LSTM model

Call make\_predictions() to make predictions using LSTM model

Call real\_time\_testing() for real-time testing

3.2 METHODOLOGY

In order to achieve our model’s evaluation we are using “CONFUSION MATRIX” to define the accuracy of the model. The accuracy of our model came out be ‘1.00’ which means 100% accurate as we are only using three phrases however in real time this came out to be ‘0.99921’which is 99.921% accurate as certain factors such as camera quality , gestures , background matters and differs as well.

In Experimental methodology , we installed and imported all the discrepancies. Then started collecting the data to form the dataset. For that we used land marks and keys to recognize the right hand. Figure 3.2(a) shows the above explanation A person with a hand raised up

Description automatically generated

Figure 3.2

(a) Framing the key points

Following we selected the 30 frames in 30 different folders to record 30 different gestures to make our dataset enriched with variation. Finally after cleaning and preprocessing our dataset using sklearn we build the LSTM model to train with our dataset and implanted the confusion matrix to formulate the training accuracy which came out to be 100% accurate. At the end, coming up with the real time testing of our model. Figure 3.2(b) shows the explanation.A person with a face recognition system

Description automatically generated with medium confidence

Figure 3.2 (b) Real Time testing

The independent variables in a sign language detection system are the factors or parameters that can be controlled, adjusted, or varied during the experiment or testing phase. On the other hand, the dependent variables are the outcomes or results that are measured, evaluated, or observed based on the changes or variations in the independent variables.

The Dataset used over here is human gesture of one of our team members for 3 different phrases which are [“HELLO”, “THANKS” , “I LOVE YOU”].The dataset is used is realistic as it contains real time human gestures obtained from the web camera and taking it as training dataset for the. The open palm gesture depicts “HELLO” , The thumbs up gesture depicts “THANKS” and 4 fingers on lips depicts the “ILOVEYOU” phrases respectively. Figure 3.2(c) shows the above explaination.

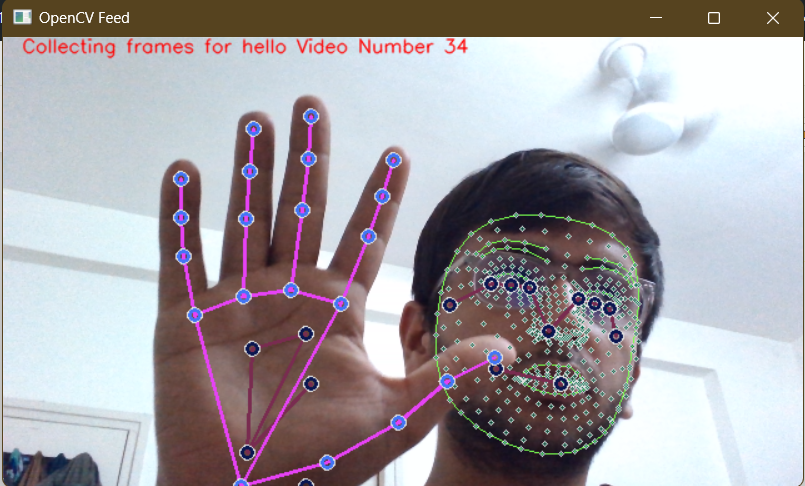


Figure 3.2 (c) Collecting data using web camera using one hand gesture.

A collage of hands making different gestures

Description automatically generated

Figure 3.2(d) Sample dataset from our source taking images for letters & converting them to phrases

1. **RESULT ANALYSIS**

.

* 1. MODEL SUMMARY

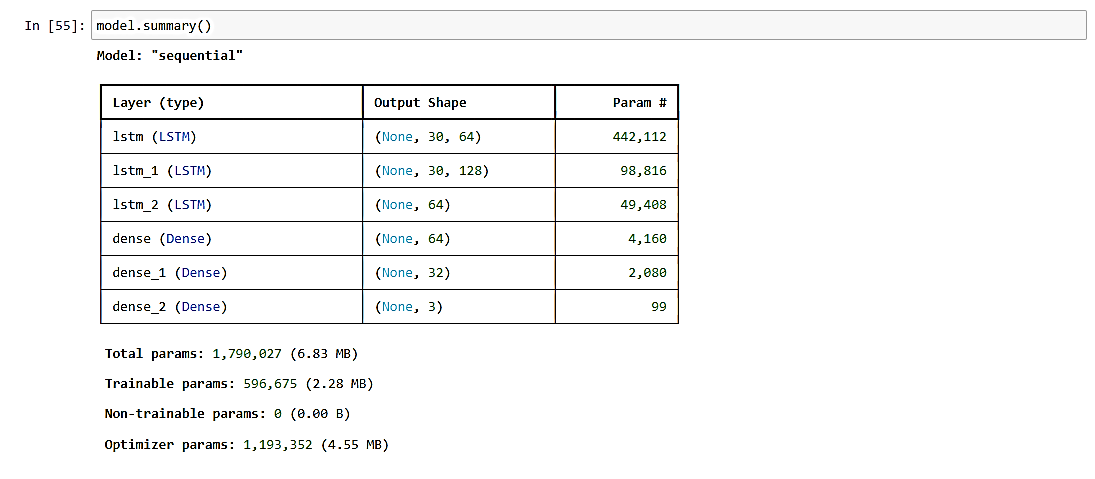


Figure 4.1(a) Summarization of LSTM MODEL

* 1. GRAPHICAL REPRESENTATIONS

The tensorboard has reperesented the model’s accuracy , learning rate and loss in a grid-line format to represent the model’s overall working. The following figures 4.2(a),(b),(c)

Shows them respectively

A graph with orange lines

Description automatically generated

Figure 4.2(a) Categorical acurracy of the model in real time

A graph with a red line

Description automatically generated

Figure 4.2(b) Learning Rate of the model

A graph with red lines

Description automatically generated

Figure 4.2(c) Categorical loss of the model in real time

Please note that epoch here means batch size which is taken 2000 for training the model.

Real time accuracy : 99.921 %

Training time accuracy : 100.000%

Loss (in %) for real time : 100-99.921 =00.079%

1. CONCLUSION AND FUTURE WORK
   1. CONCLUSION

We hereby conclude that our model has achieved real time gesture and hand recognition by using simultaneous machine learning and deep learning techniques. The major part of our project is to collect real time data and implement it create a real life scenario. Important Result : using LSTM technique a major part of accuracy can be achieved and instead of converting gestures to letters/alphabets to translation , our model has been doing this conversion in the form of gestures to translation without any usage of the letters/alphabets converting it Indo-English Format

* 1. FUTURE WORK

The Major shortcomings of our model is that we do not know about the large dataset which can convert it to phrases. Furthermore , Different countries have different languages , so our model can only work for Indo-English language and not for the other languages. Also our model cannot differentiate Fastly between the simultaneous switching of one gesture to the other as it takes some time. Background and other channel noises can also make differences in the result. We also have an idea of translation of the gestures to voice channel instead of Text translation so that deaf-dumb people can also perform generic public actions such as booking the services, work related to banks etc, in a more convincing way.

BIBLOGRAPHY

<https://www.researchgate.net/profile/Yogeshwar-Rokade/publication/318656956_Indian_Sign_Language_Recognition_System/links/597af169a6fdcc61bb3fb2bb/Indian-Sign-Language-Recognition-System.pdf>

<https://ieeexplore.ieee.org/abstract/document/6157974>

<https://www.youtube.com/watch?v=doDUihpj6ro&t=1387s>