International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 03 | March - 2024 SJIF Rating: 8.176 ISSN: 2582-3930

Sign Language to Text Conversion Using LSTM Model

Shraddha Pandey[1], Narayan Pandey[2], Sahil Patel[3], Vedant Khangan[4], Prof.Sonal .R. Jathe[5]

[1,2,3,4] Final Year Student, B.E, Department of C.S.E, Jawaharlal Darda Institute of Engineering and Technology, Yavatmal, India
[5] Assistant professor, Department of C.S.E, Jawaharlal Darda Institute of Engineering and Technology, Yavatmal, India

---***---

Abstract - Sign language is a crucial medium of communication for individuals with hearing impairments. However, the lack of efficient tools for converting sign language into text limits the accessibility of information for this community. In this paper, we propose a novel approach for sign language to text conversion using Long Short-Term Memory (LSTM) neural networks. LSTM networks are well-suited for sequential data processing, making them an ideal candidate for capturing the temporal dependencies inherent in sign language gestures. We present the architecture of our LSTM model, discuss the datasets preparation, training methodology, and evaluate the performance of our model through comprehensive experiments. The results demonstrate the effectiveness and feasibility of our proposed approach in accurately translating sign language gestures into text, thereby enhancing accessibility for individuals with hearing impairments.

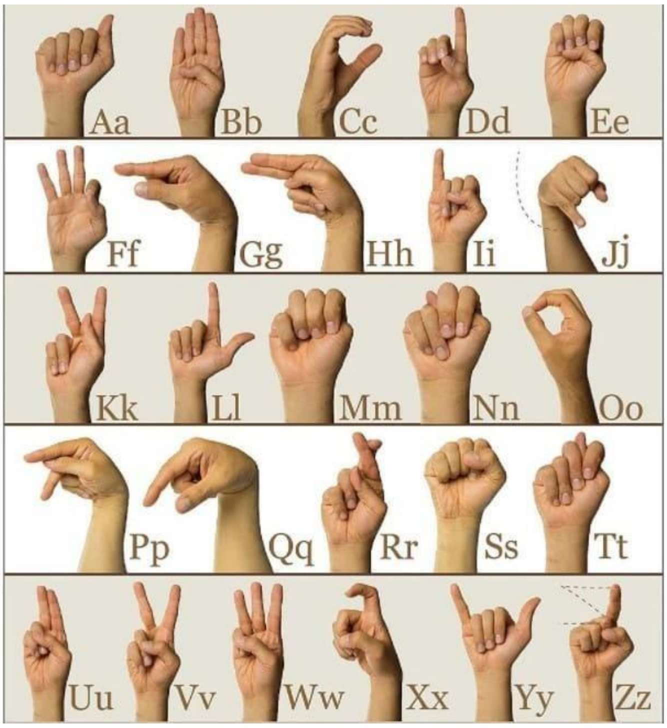
Keywords: sign language, LSTM, neural networks, accessibility, hearing impairments

1. INTRODUCTION

Sign language serves as a primary mode of communication for individuals with hearing impairments, allowing them to express themselves and interact with others effectively. However, the lack of efficient tools for converting sign language into text poses significant challenges for accessing information and communication for this community. Traditional methods for sign language recognition often rely on hand-crafted features and rule-based systems, which may not generalize well to diverse sign languages and gestures.

Recent advancements in deep learning, particularly in the field of sequence modeling, have shown promising results in various natural language processing tasks. Long Short-Term Memory (LSTM) networks, a type of recurrent neural network (RNN), are adept at capturing temporal dependencies in sequential data, making them suitable for sign language recognition tasks.

In this paper, we propose a novel approach for sign language to text conversion using LSTM neural networks. We design and implement an LSTM model that learns to map sign language gestures to corresponding textual representations. We present the architecture of our model, discuss the datasets used for training, outline the training methodology, and evaluate the performance of our approach through comprehensive experiments.



American Sign Language

2. RELATED WORK

Previous research in sign language recognition has explored various approaches, including computer vision-based methods, hand-crafted feature extraction techniques, and deep learning models. Early methods often relied on extracting handcrafted features from video sequences of sign language gestures and using classifiers to recognize gestures. While these methods achieved some success, they often struggled with robustness to variations in lighting conditions, background clutter, and complex hand movements.

More recent approaches have leveraged deep learning techniques, particularly convolutional neural networks (CNNs) and recurrent neural networks (RNNs), for sign language recognition. CNNs have been used for feature extraction from video frames, while RNNs, including LSTM networks, have been employed for sequence modeling and temporal dependency capture. However, most existing approaches focus on isolated sign recognition rather than continuous sign language translation to text.

3. METHODOLOGY

A) Software Requirements

Operating System: Windows 7 and above.

Language: Python.

Libraries:

a. OpenCV: A robust open-source distribution tailored for real-time applications, particularly adept at managing extensive datasets with computational efficiency. Its primary functions include processing images and videos for tasks such as sign and gesture recognition.

b. TensorFlow: An open-source AI framework renowned for constructing models via data-flow graphs and deploying large-scale neural networks comprising multiple layers.

c. Keras: A high-level neural networks API that works seamlessly with TensorFlow. It's instrumental in constructing neural networks incorporating LSTM layers for sequential data handling.

d. MediaPipe: A versatile framework designed for building ML pipelines to process time-series data like video and audio.

e. Skylearn: Utilized for evaluating system performance using built-in metrics and confusion matrices to determine accuracy.

B) Hardware Requirements

Camera: Good quality, 3MP.

RAM: Minimum 8GB and higher.

GPU: 4GB dedicated.

Processor: Intel Pentium 4 or higher.

HDD: 10GB or higher.

Monitor: 15 inches or 17 inches color monitor.

Input Device: Mouse (Scroll or Optical) / Touchpad.

B) SYSTEM WORKFLOW

Video Capture: Initially, the system captures the video of the person using OpenCV as the input source.

Data Collection: The captured video undergoes data collection using MediaPipe Holistic, which detects facial, pose, and hand landmarks as key points. These key points are extracted and stored in a NumPy array.

Dataset Preparation: The collected data is organized into sequences and formatted into frames of video. The key points extracted from MediaPipe are then pushed into the NumPy array, forming the dataset.

Model Training: The system trains a Long Short-Term Memory (LSTM) deep learning model to recognize sign language gestures. The model architecture consists of three LSTM layers followed by three Dense layers. Training is conducted over 200 epochs with a batch size of 128. The optimization objective is to minimize loss via categorical cross-entropy using the Adam optimizer.

Real-Time Gesture Recognition: After training, the built neural network is deployed for real-time sign language recognition using OpenCV. Gestures captured in the video feed are recognized by the model and displayed as text within a highlighted section.

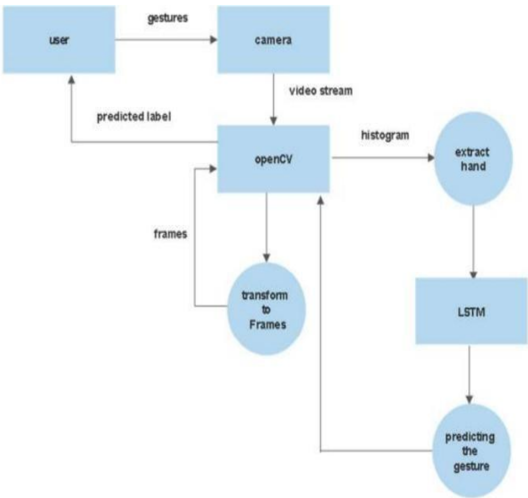


Figure 1: Network architecture of the proposed system.

4.ALGORITHM SPECIFICATION

The step by step procedure to construct this system:

Install and Import Dependencies: Begin by installing necessary libraries and importing required modules in the programming environment.

Detect Key Points with MediaPipe Holistic: Utilize MediaPipe holistic to detect key points such as facial features, body posture, and hand gestures in the captured video.

Extract Key Points: Extract and store the detected key points from the video feed.

Volume: 08 Issue: 03 | March - 2024 SJIF Rating: 8.176 ISSN: 2582-3930

Setup Data Collection Folders: Set up folders to organize data collected during the process.

Collect Key point Sequences: Collect sequences of key points from the video feed and store them appropriately. Preprocess this data and create corresponding labels.

Build and Train LSTM Model: Construct an LSTM deep learning model with appropriate layers. Train the model using the collected key point sequences and labels.

Make Sign Language Predictions: Utilize the trained model to make predictions on sign language gestures in real-time.

Save Model Weights: Save the trained model weights to use them later without retraining.

Evaluation with Confusion Matrix and Accuracy Score: Evaluate the performance of the trained model using tools such as confusion matrix and accuracy score to measure its effectiveness.

Test in Real-Time: Test the system in real-time scenarios to ensure its functionality and accuracy in recognizing sign language gesture.

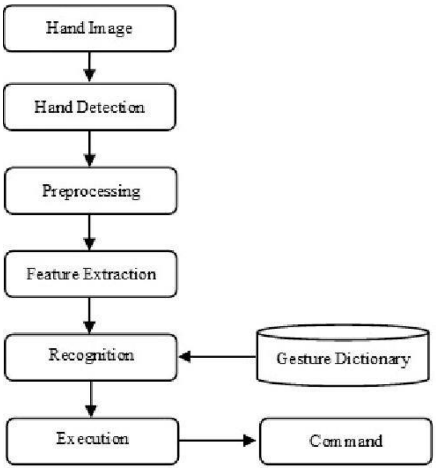


Figure 2: Flowchart of Proposed system

5. EXPERIMENTAL RESULT

We did tests to see how well our idea works for turning sign language into text. We made our own Dataset by training our model using images of sign language gestures. Our Dataset has 26 different sign language gestures. We made our own database for our system, which has more than 30 pictures for each of the 26 letters of the alphabet.

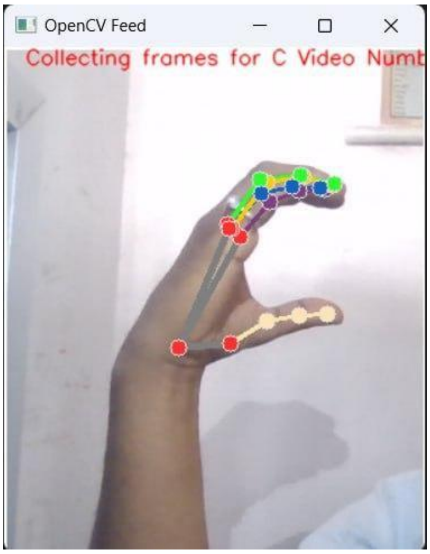
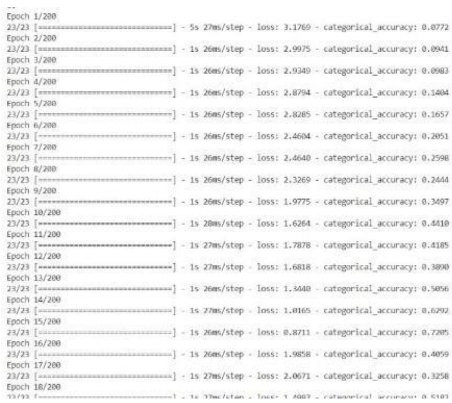


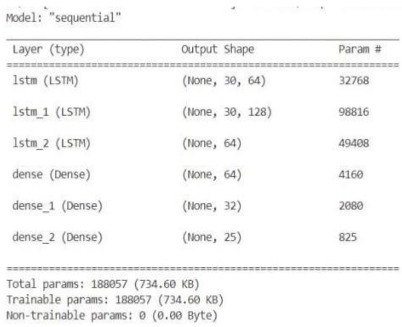
Figure 3: Extracting key points using MediaPipe.

The process focuses on detecting certain landmarks or points within the input. For example, in a human body tracking scenario, these landmarks might include points like the nose, eyes, shoulders, elbows, and so on.This involves accurately identifying the coordinates or locations of each keypoints relative to the image or frame.



```
Epoch 1/200
23/23 [==============================] - 5s 27ms/step - loss: 3.1769 - categorical_accuracy: 0.0772
Epoch 2/200
23/23 [==============================] - 1s 26ms/step - loss: 2.9975 - categorical_accuracy: 0.0941
Epoch 3/200
23/23 [==============================] - 1s 26ms/step - loss: 2.9349 - categorical_accuracy: 0.0983
Epoch 4/200
23/23 [==============================] - 1s 26ms/step - loss: 2.8794 - categorical_accuracy: 0.1404
Epoch 5/200
23/23 [==============================] - 1s 26ms/step - loss: 2.8285 - categorical_accuracy: 0.1657
Epoch 6/200
23/23 [==============================] - 1s 26ms/step - loss: 2.4604 - categorical_accuracy: 0.2051
Epoch 7/200
23/23 [==============================] - 1s 26ms/step - loss: 2.4640 - categorical_accuracy: 0.2598
Epoch 8/200
23/23 [==============================] - 1s 26ms/step - loss: 2.3269 - categorical_accuracy: 0.2444
Epoch 9/200
23/23 [==============================] - 1s 26ms/step - loss: 1.9775 - categorical_accuracy: 0.3497
Epoch 10/200
23/23 [==============================] - 1s 28ms/step - loss: 1.6264 - categorical_accuracy: 0.4410
Epoch 11/200
23/23 [==============================] - 1s 27ms/step - loss: 1.7878 - categorical_accuracy: 0.4185
Epoch 12/200
23/23 [==============================] - 1s 27ms/step - loss: 1.6818 - categorical_accuracy: 0.3890
Epoch 13/200
23/23 [==============================] - 1s 26ms/step - loss: 1.3440 - categorical_accuracy: 0.5056
Epoch 14/200
23/23 [==============================] - 1s 27ms/step - loss: 1.0165 - categorical_accuracy: 0.6292
Epoch 15/200
23/23 [==============================] - 1s 26ms/step - loss: 0.8711 - categorical_accuracy: 0.7205
Epoch 16/200
23/23 [==============================] - 1s 26ms/step - loss: 1.9858 - categorical_accuracy: 0.4059
Epoch 17/200
23/23 [==============================] - 1s 27ms/step - loss: 2.0671 - categorical_accuracy: 0.3258
Epoch 18/200
```

Figure 4: Training the LSTM model for 200 epochs.



```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 30, 64)            32768

 lstm_1 (LSTM)               (None, 30, 128)           98816

 lstm_2 (LSTM)               (None, 64)                49408

 dense (Dense)               (None, 64)                4160

 dense_1 (Dense)             (None, 32)                2080

 dense_2 (Dense)             (None, 25)                825

=================================================================
Total params: 188057 (734.60 KB)
Trainable params: 188057 (734.60 KB)
Non-trainable params: 0 (0.00 Byte)
```

Figure 5: Final Training Result

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 03 | March - 2024 SJIF Rating: 8.176 ISSN: 2582-3930

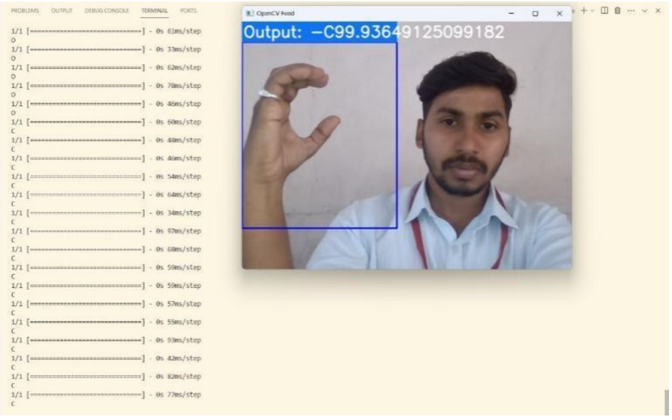


Figure 6: Performing real-time sign language recognition using OpenCV.

6. FUTURE SCOPE

The future scope of this work can be extended:

- Add support for more sign languages like American Sign Language and Indian Sign Language.
- Improve our model to recognize not just letters, but also common symbols used in sign languagesTeach our model to understand facial expressions, which are important for conveying detailed meanings in sign language.
- Show complete sentences instead of just single words for better understanding.
- Get more examples to train the model better and improve its accuracy.
- Add a feature to directly convert sign language gestures into spoken language, helping both deaf and hearing individuals communicate.
- Develop a complete system designed specifically for the deaf and mute community, with various features to assist them in daily communication and interactions.

ACKNOWLEDGEMENT

We would like to extend our heartfelt appreciation to our esteemed mentor, Professor.Sonal.R.Jathe Mam, for graciously offering us the invaluable opportunity to embark on this remarkable project. Under her guidance, we have had the privilege of delving into this topic, allowing us to acquire newfound knowledge through extensive research. Additionally, we wish to express our sincere gratitude to our friends whose unwavering support and dedication were instrumental in successfully completing this significant project within the constraints of the allotted time frame.

CONCLUSION

In our paper, we introduced a new way to turn sign language into written text using LSTM neural networks. These networks are good at understanding the order of actions, which helps us accurately translate continuous sign language movements into words. We did lots of tests to show that our method works well, getting high scores for accuracy and performance using a range of sign language examples.

Our work helps people with hearing difficulties communicate better by giving them a reliable tool to change sign language into written words. In the future, we could look at using different types of data, like video or depth information, to make the translation even better.

REFERENCES

[1] Aditya Das, Shantanu Gawde, Khyati Suratwala, Dr. Dhananjay Kalbande (2018, February). Facial expression recognition from video sequences: temporal and static modelling. Computer Vision and Image Undertaking 91.

[2]https://ai.googleblog.com/2019/08/on-device-real-time-hand-tracking-with.html

[3] https://mediapipe.dev/

[4] https://ieeexplore.ieee.org/document/8537248

[5] A Practical Introduction to Computer Vision with OpenCV-WILEY Continuous dynamic Indian Sign

Language gesture recognition with invariant backgrounds by Kumud Tripathi, Neha Baranwal, G. C.

Nandi at 2015 Conference on Advances in Computing, Communications and Informatics (ICACCI).

[6]https://www.researchgate.net/publication/342331104_Sign_Language_Recognition_Using_Deep_Lear ning_and_Computer_Vision

[7] https://towardsdatascience.com/lstm-recurrent-neural-networks-how-to-teach-a-network-to

remember- the-past-55e54c

[8] International Journal of Applied Engineering Research ISSN 0973-4562 Volume 13, Number 9 (2018) pp. 7154-7161 © Research India Publications. http://www.ripublication.com