Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 03, March-2025

Connecting Worlds: A Deep Learning Approach to Real-Time Sign Language Translation

Manasa Krishna BA, Nithya B, Madhu MM, Yashwanth Kumar K, Dr. Jyothi DG
Department of Artificial Intelligence and Machine Learning
Bangalore Institute of Technology
Bangalore, India

***Abstract*—Connecting worlds: A Deep Learning Approach to a Real-Time Sign Language Translation project focused on developing a real-time Sign Language conversion system using MediaPipe for efficient gesture detection and Long Short-term Memory (LSTM) networks for accurate sequence translation. MediaPipe is the backbone for capturing and preprocessing hand landmarks, offering high-speed, lightweight, and precise gesture recognition. The results are fed into an LSTM model, which captures temporal dependencies in sequential data, making it perfect for interpreting dynamic sign language hand gestures. The system is designed to convert sign language gestures into textual output. The key roles of the project are dataset preparation using various sign language gestures, preprocessing pipelines to normalize gestures, and real-time integration to provide a seamless interaction experience. With the efficiency of MediaPipe and the ability to perform sequential learning with LSTM, this solution is expected to provide a friendly, user-accessible, and scalable translation tool that improves the accessibility and engagement of the deaf and hard-of-hearing community.**

***Keywords*—Real-time Sign Language Interpretation, MediaPipe, Hand Landmark Detection, LSTM Model**

I. INTRODUCTION

Barriers to communication between the deaf or hard-of-hearing community and those who lack knowledge of sign language present significant challenges in both social and professional contexts. This project's goal is to design and develop a system that converts various sign language hand gestures into text, making communication more accessible and inclusive.

Sign language gestures are inherently time-series in nature, requiring models that effectively address temporal dependencies. LSTM networks excel in this domain, enabling the system to process dynamic gesture sequences and translate them into meaningful text or speech. Trained on diverse datasets, the model is robust and adaptive, accommodating variations in hand shapes, movements, and even different sign languages, thereby broadening its applicability.

The system is designed for real-time integration, making it practical for everyday use. Its preprocessing pipeline standardizes gestures, enhancing accuracy across a wide range of users and environmental conditions. By combining MediaPipe's real-time gesture detection capabilities with the deep learning power of LSTM networks, the system served an accessible and user-friendly tool that bridges the communication gap between signers and non-signers.

Beyond its immediate practical applications, this research contributes to advancements in gesture recognition, sequential modeling, and assistive technologies, fostering a more inclusive and accessible world.

II. RELATED WORK

[1] This proposed work involves Indian Sign Language (ISL) generated from text or live audio, taking inputs through CNN and RNNs, for understanding in Tamil-speaking persons. Proposed systems target a live audio-to-sign translation with heavy dependency on gloss annotations to derive a correct translation.

[2] Designed an audio-based sign language translator using MFCC for feature extraction. Although the system is performing exceptionally well in the audio processing, it is sensitive to the surrounding environment and requires high quality audio.

[3] This research proposed a bilingual ISL recognition system, which will convert the gesture into text and audio. The model allows several languages but faces regional dialect challenges and inconsistencies in the datasets of the gestures.

[4] Presented a real-time visual-audio translator for disabled individuals using human-computer interfaces. The system cannot process gesture and voice inputs simultaneously, employing techniques like template matching, and dialect recognition is difficult with this system.

[5] Developed a system that translates gestures into voice for the deaf and mute, which is inefficient although good in translation, difficult in handling complex gestures, and needs a bigger dataset to train.

[6] This paper presents a system that converts sign language into text and speech using deep learning. The solution is innovative but requires large computational power and resources.

[7] This paper presents a gesture prediction system that is used as the approach for sign-to-text translation. The system performed pretty well in static gesture recognition, but it failed at recognizing dynamic gestures and doing real-time translation.

[8] This study developed an application that converts sign language to audio using image processing. It is resource efficient but limited to basic gestures and fails to handle complex or overlapping gestures.

Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 03, March-2025

III. METHODOLOGY

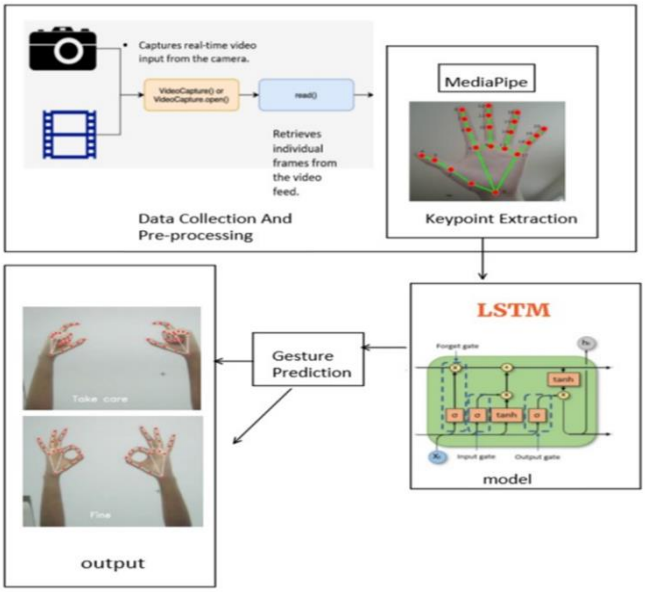

Fig. 1. Methodology

A. Data Collection

For the sake of digital communication of deaf-to-hearing persons, raw data of sign language was collected by camera frames that captured the hand gestures and finger movements.

Input Data: The input data includes the source as images or video frames. The images are processed sequentially for real time recognition and also landmark detection and tracking of the hand or face.

Device Used: The webcam or camera device captures the video stream more quickly and accurately.

B. Data Preprocessing

The gesture images are subjected to a few preprocessing steps before processing them in which their quality is ensured:

Resizing: This step refers to the process of making all images in identical format by reducing the sizes which normally is done to a smaller resolution. At the same time, the model from Mediapipe is standardized by it, meaning less computational load.

Conversion to RGB: The image captured by OpenCV in BGR type needs to be converted to RGB before it can be processed by the Mediapipe model which is done since Mediapipe is only compatible with RGB.

C. Landmark Detection

Mediapipe Hand Landmark Detection: The model is the image processed to find the specific points that belong to the landmarks in hand. For example, the top fingers or center of the palm. The hand model of Mediapipe will return all landmarks of left and right hands if those are visible in the image. Landmarks Extraction: Once the landmarks are detected, the key points for both hands (left and right) are extracted, which are the 3D coordinates (x, y, z). Moreover, if the landmarks are not detected (e.g., the hands are absent in the frame), the associated key points will become zeros, making the system not fail in such cases. Keypoint Concatenation: After the hand-detected key points marked and extracted from both hands are concatenated into a single array, which is used in the final feature vector, which is used for classification or gesture recognition tasks.

IV. MODEL DEVELOPMENT

A. Model Architecture

The LSTM model is a type of RNN that is particularly effective for sequential data. This model is designed to capture long-term dependencies in data by using memory cells.

Working of LSTM

- It works by remembering important information from earlier in the sequence and forgetting irrelevant data using special components called gates.
- Forget gate: Decide what to discard
- Input gate: Determines what new information to store
- Output gate: Decides what to send as output.

$$i_t = \sigma(w_i[h_{t-1}, x_t] + b_i)$$
$$f_t = \sigma(w_f[h_{t-1}, x_t] + b_f)$$
$$o_t = \sigma(w_o[h_{t-1}, x_t] + b_o)$$

Fig. 2. Equation of gates

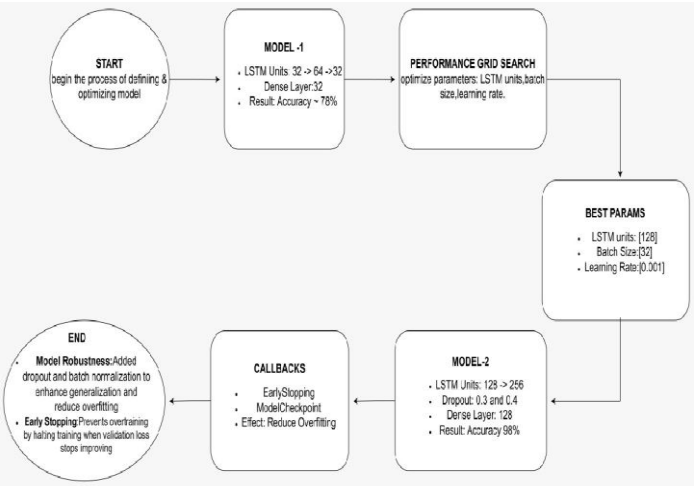

Fig. 3. Grid Search for Hyperparameter-Optimization

Goal: Determine the best set of hyperparameters for the model by doing a Grid Search.

Hyperparameters Tuned: The number of LSTM units is set to 32 and 64, the batch size is set to 16 and 32, and the learning rate is set to 0.001 and 0.01.

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 03, March-2025

Procedure:

1. For each combination of the above hyperparameters, iteratively train models.
2. Evaluate all the models using the test dataset with accuracy as the evaluation criterion.
3. Monitor the best-performing combination of hyperparameters along with the corresponding model.

Outcome: During the processing, the combination yielding the highest accuracy was identified and saved. The corresponding model architecture, hyperparameters, and weights were retained for use further.
The final model is as follows: Input Layer: It takes sequences of 10 frames, each containing 126 features (from both hands).

LSTM Layers:

- The first LSTM layer consists of 128 units to capture temporal dependencies.
- Second LSTM layer of 256 units for high-level sequential patterns.
- Activation function: tanh
- Dropout: After LSTM and Dense layers, it has been used to prevent overfitting, with rates of 0.3, 0.4, and 0.3, respectively.
- Batch Normalization: Normalization used after the first LSTM layer to stabilize learning.
- The output layer uses the softmax activation for multi-class classification.
- This architecture aims to learn temporal patterns from gesture sequences.

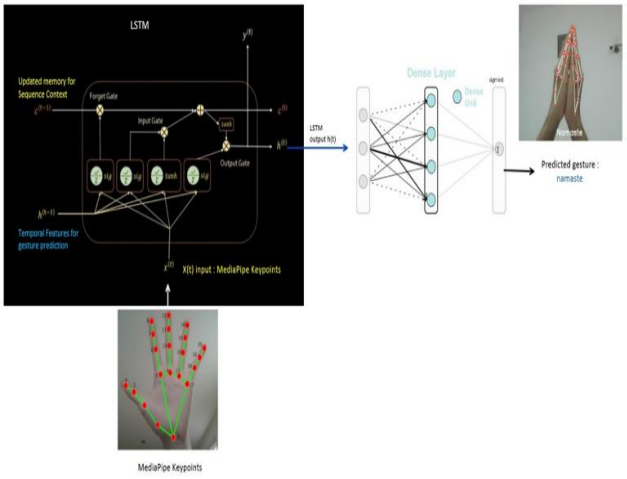

Fig. 4. LSTM-Model Working

B. Model Compilation

The model was implemented with the Adam-optimizer and categorical cross-entropy loss to optimize the multi-class classification.

C. Model Training

The model was trained using a training dataset of batch size of 32 for up to 100 epochs and early stopping to prevent overfitting. Validation was performed on the test dataset, and callbacks ensured the best-performing model was saved.

D. Model Deployment

The trained model is integrated with OpenCV for real-time video input processing. During deployment, each frame from the video feed is processed to extract hand landmarks, which are fed into the model. The model's output, representing the recognized gesture, is mapped to the corresponding text and displayed as subtitles over the video using OpenCV's `putText` function. This enables seamless real-time gesture-to-text conversion.

E. Model Evaluation

Evaluation of the model involves assessing the model's performance on the test data using evaluation metrics such as accuracy, precision, and recall. The results are obtained by comparing the predicted labels with the true labels, providing an understanding of the model's effectiveness in recognizing gestures.

V. RESULTS

The developed model demonstrated strong performance and recognition of sign language gestures, achieving an overall accuracy of 92.5% on the test-dataset. The classification report revealed precision, recall, and F1-scores across most gestures, with values consistently above 0.90 for a majority of the classes. Certain challenges were observed, as a few gestures, such as 'yes' and 'no', were occasionally misclassified, which is reflected in the confusion matrix. Despite these minor errors, the model effectively distinguished between most gestures. Additionally, the model's loss and accuracy improved steadily during training, indicating a well-converged model.

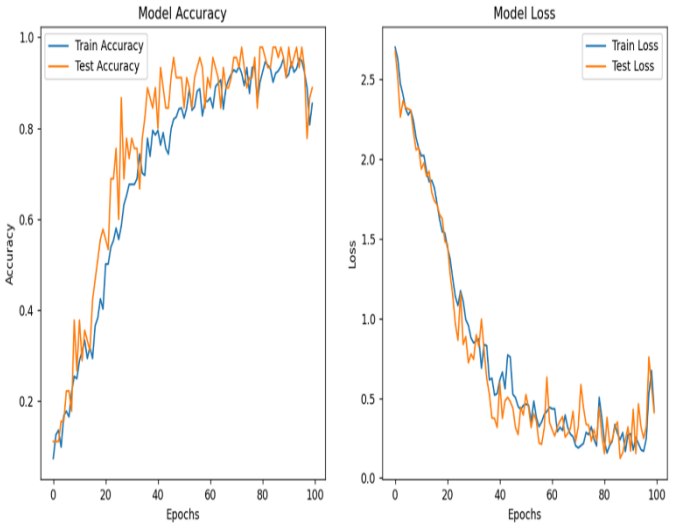

Fig. 5. Accuracy and loss

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 14 Issue 03, March-2025

Fig. 6. Output

Fig. 7. Realtime sign gesture translation

VI. CONCLUSION

The real-time sign-to-sentence system, using Mediapipe and LSTM, efficiently identifies gestures without any interruptions in the communication process for hearing impaired and speech-impaired patients. Optimized LSTM addresses the complexity of input, thus ensuring inclusivity in dialect, and a robust hand-tracking system is provided by Mediapipe, which reduces the number of gloss annotations and makes this system reliable, versatile, and effective for real world applications and fosters inclusiveness and empowerment of communication processes.

REFERENCES

[1] B. R. Reddy, D. C. Rup, M. Rohith, and M. Belwal, "Indian Sign Language Generation from Live Audio or Text for Tamil," 2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2023, pp. 1507-1513. doi: 10.1109/ICACCS57279.2023.10112880.

[2] J. Vijaya, C. Mittal, C. Singh, and M. A. Lekhana, "An Efficient System for Audio-Based Sign Language Translator Through MFCC Feature Extraction," 2023 International Conference on Sustainable Communication Networks and Applications (ICSCNA), Theni, India, 2023, pp. 1157-1164. 10.1109/ICSCNA58489.2023.10370086.

[3] N. Chandarana, S. Manjucha, P. Chogale, N. Chhajed, M. G. Tolani, and M. R. M. Edinburgh, "Indian Sign Language Recognition with Conversion to Bilingual Text and Audio," 2023 International Conference on Advanced Computing Technologies and Applications (ICACTA), Mumbai, India, 2023, pp. 10.1109/ICACTA58201.2023.10393571.

[4] B. Patil and G. D. Nagoshe, "A Real-Time Visual-Audio Translator for Disabled People to Communicate Using Human-Computer Interface System," International Research Journal of Engineering and Technology (IRJET), vol. 08, no. 04, pp. 2395-0056, April 2021.

[5] J. E. Job, N. E. Aniyan, Meera J., Mia Dhas, and Dandeep Chandran, "Hand Gesture Recognition and Voice Conversion for Deaf and Dumb," International Journal of Computer Science Trends and Technology, vol. 12, no. 2, 2024, ISSN: 2347-8578.

[6] P. Duraiswamy, A. Abhinayasrijanam, M. A. Candida, and P. Dinesh Babu, "Transforming Sign Language into Text and Speech Through Deep Learning Technologies," Indian Journal of Science and Technology, vol. 16, no. 45, pp. 4177-4185, 2023. doi: 10.17485/IJST/v16i45.2583.

[7] T. Kemkar, V. Rai, and B. Verma, "Sign Language to Text Conversion Using Hand Gesture Recognition," 2023 8th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2023, pp. 1580-1587. 10.1109/ICCES57224.2023.10192820.

[8] Subashini, V., et al. "Sign Language Translation Using Image Processing to Audio Conversion." 2024 Third International Conference on Intelligent Techniques in Control, Optimization and Signal Processing (INCOS). IEEE, 2024.