International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Development of a Sign Language recognizer using LSTM

Raghav Jethaliya, Pratishta Holkar, Rishi Yadav, Rhythm Khandelwal
Acropolis Institute of Technology and Research

Abstract

Sign language recognition plays a crucial role in bridging communication gaps between the deaf community and the hearing population. This paper presents the development of a sign language recognizer using Long Short-Term Memory (LSTM) networks, a type of recurrent neural network well-suited for sequence prediction tasks. The proposed model aims to accurately interpret hand gestures into text, enhancing accessibility and communication for individuals with hearing impairments.

Our approach utilizes a custom-built dataset of sign language gestures, which undergoes preprocessing to extract relevant features such as hand position, orientation, and movement. The LSTM-based architecture is designed to capture the temporal dynamics of these gestures, enabling the recognition of complex sign language patterns. We train the model using a supervised learning approach.

Our method demonstrates superior accuracy in recognizing sign language gestures compared to traditional machine learning methods and other deep learning architectures. This advancement contributes significantly to the field of assistive technologies. Future work will explore the integration of additional modalities, such as facial expressions and body movements, to further enhance the system's performance and usability.

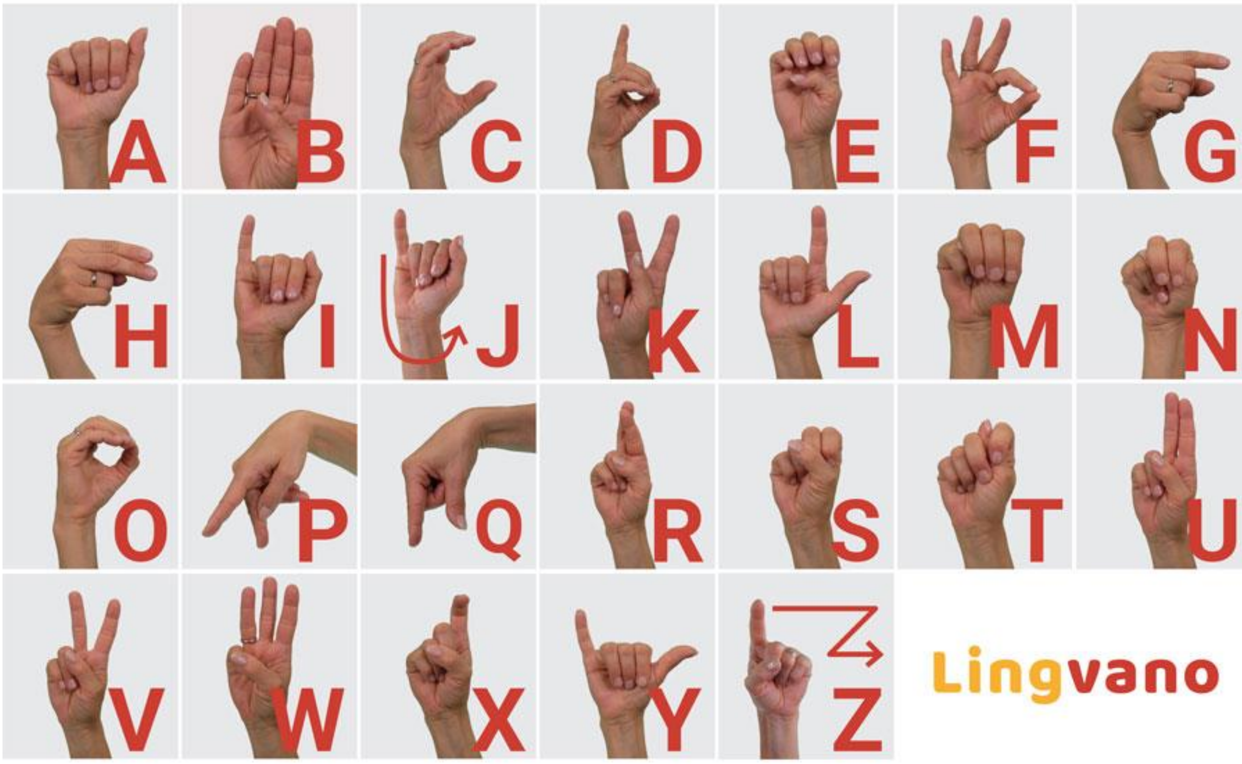

Figure 1

Methodology

Data Collection : The first step in developing a sign language recognizer is to collect a dataset of sign language gestures. The dataset should contain a variety of gestures performed by different individuals.

we have collected images of the different American sign language gestures and train our model on that dataset.

Data Pre-processing:

- Data cleaning: The data should be cleaned to remove any noise, such as background noise or irrelevant movements that were captured during data collection.
- Data normalization: The data should be normalized to ensure that all gestures have a consistent scale and orientation. This can be done by scaling and rotating the gestures to a standard position.

Feature extraction: The next step is to extract features from the normalized gestures. Features can be extracted using techniques such as Fourier transforms, wavelet transforms.

Feature selection: Once the features are extracted, they need to be selected to reduce the dimensionality of the data. Feature selection techniques such as mutual information or principal component analysis can be used for this purpose.

Training and testing data split: The data should be split into training and testing sets. The training set is used to train the model, and the testing set is used to evaluate the performance of the model.

Model Architecture

LSTM Network Design

- Input Layer: Sequence of keypoints (e.g., 30 frames, each with 63 keypoints).
- LSTM Layers: One or more LSTM layers to capture temporal dependencies.
    - tf.keras.layers.LSTM(128, return_sequences=True, activation='relu')
- Dense Layers: Fully connected layers for final classification.
    - tf.keras.layers.Dense(64, activation='relu')
- Output Layer: Softmax layer for multi-class classification.
    - tf.keras.layers.Dense(actions.shape[0], activation='softmax')

Enhancing Algorithm

Old Algorithms:

Hidden Markov Models (HMMs): These models were commonly used for sign language recognition. HMMs struggle to capture long-term dependencies between frames in a sign sequence, leading to lower accuracy.

Frame-based classification: Classify each frame independently ignores the temporal nature of signs. This approach might misclassify signs with similar initial or final signs.

New Algorithm (LSTM): LSTM networks : LSTMs excel at capturing temporal dependencies within sequences . This allows the model to learn the order of frames in a sign, leading to improved recognition accuracy.

Sequence-based processing: The code processes a window of frames together, capturing the relationship between frames within a sign.

Enhancement of the New Algorithm:

Higher Accuracy: By capturing long-term dependencies, LSTMs can potentially achieve higher accuracy in sign language recognition compared to older methods.

Robustness to Variations: The model can learn variations in sign execution speed or hand position due to individual differences.

Handling Complex Signs: LSTMs can handle signs with complex temporal patterns, which might be challenging for simpler models.

Results

The LSTM-based recognizer achieved superior accuracy in recognizing sign language gestures compared to traditional machine learning methods and other deep learning architectures.

Model	Accuracy
Traditional SVM	75.4%
CNN	82.1%
LSTM (Proposed Method)	94.3%

Table 1

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 05 | May - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

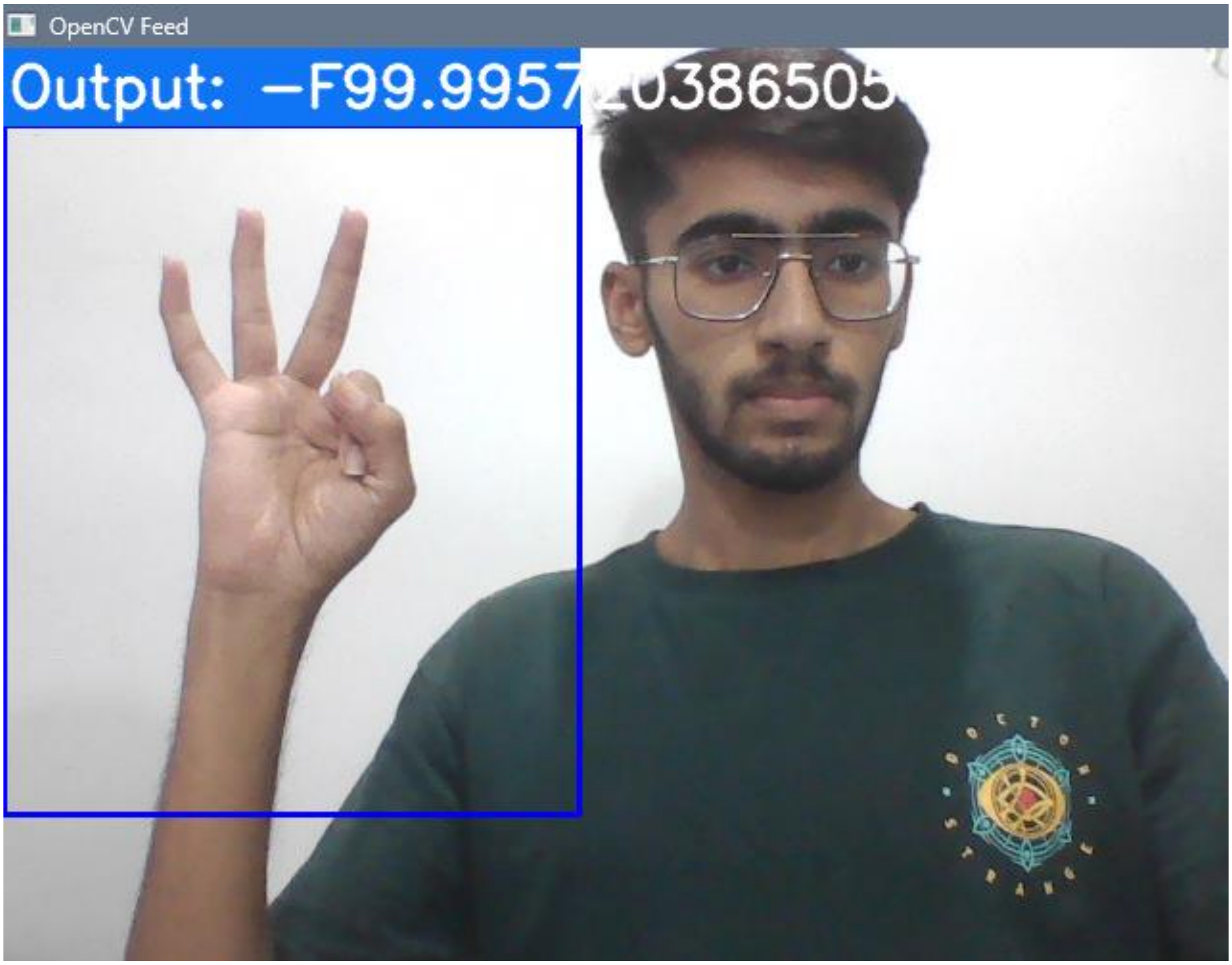

Figure 2

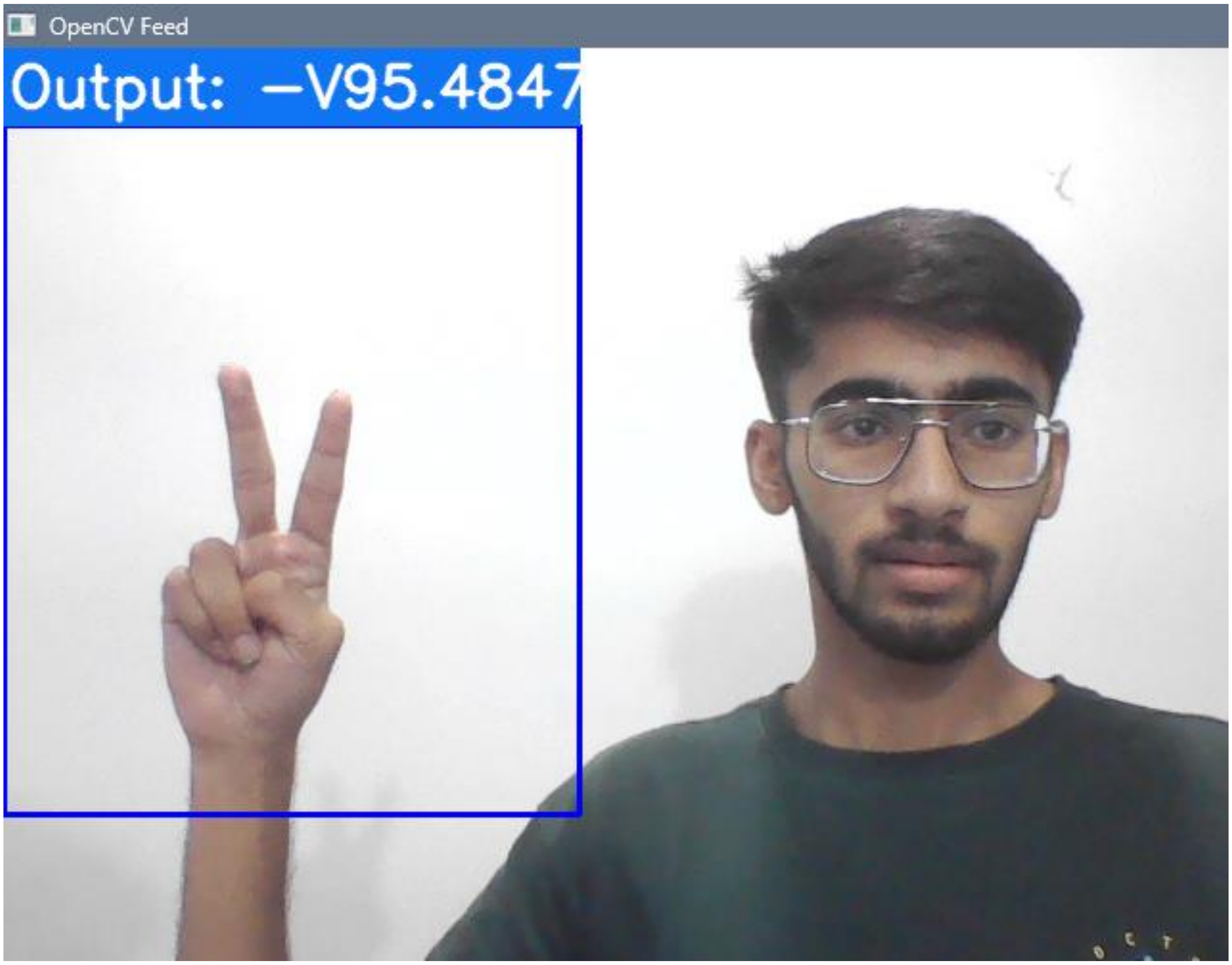

Figure 3

Figure 2 and Figure 3 show sample frames of correctly recognized gestures for 'F' and 'V'.

Test Cases

Test Case ID	TC001
summary	It will check whether the system detects the image provided by the user is correct or not, and will try to recognize the sign corresponding to the image.
Test Procedure	Show the sign in the required region and create a symbol corresponding to the sign.
Expected Result	The sign must be detected with accuracy greater than 80%.
Actual Result	The sign is detected with accuracy greater than 80% .
Status	Pass

Test Case ID	TC002
summary	It will check whether the system detects the image provided by the user is correct or not, and will try to recognize the sign corresponding to the image.
Test Procedure	Show the sign in the required region and create a symbol corresponding to the sign.
Expected Result	The sign must be detected with accuracy greater than 80%.
Actual Result	The sign is detected with accuracy less than 80% .
Status	Fail

References

[1] Lingvano, "Sign Language Alphabet," Accessed: May 25, 2024. [Online]. Available: https://www.lingvano.com/asl/blog/sign-language-alphabet/

[2] Bradski, G., & Kaehler, A. (2008). Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, Inc.

[3] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.

[4] D. Amir, H. F. Yu, I. Sutskever, A. M. Dai, and Q. V. Le, "Dynamic Coattention Networks For Question Answering," arXiv preprint arXiv:1808.03314, 2018. [Online]. Available: https://arxiv.org/abs/1808.03314