

| | | |
|--|--|------------------------------------|
| ITC 3/53 Information Technology and Control Vol. 53 / No. 3 / 2024 pp. 888-898 DOI 10.5755/j01.itc.53.3.33935 | Real-time Interpreter for Short Sentences in Indian Sign Language Using MediaPipe and Deep Learning | |
| | Received 2023/04/25 | Accepted after revision 2024/05/17 |
| | HOW TO CITE: Mariappan, S., Murugesan, P., Selvan, H. M. (2024). Real-time Interpreter for Short Sentences in Indian Sign Language Using MediaPipe and Deep Learning. <i>Information Technology and Control</i> , 53(3), 888-898. https://doi.org/10.5755/j01.itc.53.3.33935 | |

Real-time Interpreter for Short Sentences in Indian Sign Language Using MediaPipe and Deep Learning

Suguna Mariappan, Ponmalar Murugesan, Hemapriya Muthamil Selvan

Department of Computer Science, Thiagarajar College of Engineering, Madurai, India

Corresponding author: mscse@tce.edu

The expression of thoughts and feelings through communication plays a major part of human life in building relationship among others. Most of the population with hearing ability expresses their thoughts in their own or known language through voice-oriented communication. The people belonging to deaf-mute community uses hand movement gestures and expressions of face for communication which is called sign language. There exists a difficulty in building a conversation between the hearing community and non-hearing community. To make easy conversation of deaf-mute people with the external world and to connect the gap for communication between the hearing people and non-hearing people, we developed an interpreter that translates sign language to text. Most system developed for the recognition of Indian Sign Language is built for alphabets and numbers. We attempted in building a model for 15 meaningful short sentences of Indian sign gestures using, custom built video datasets captured using OpenCV, keypoints of hands, pose and face extracted using MediaPipe. The model is trained using LSTM and achieved training and testing accuracy of 99.17% and 97.78% respectively.

KEYWORDS: Keypoints, LSTM, MediaPipe, OpenCV, Sign language.

1. Introduction

Communication plays a major part of human life. It paves a way in building a strong relationship among people by sharing their thoughts, feelings and knowledge with others. The people with hearing and speaking ability use voice-oriented communication. The

majority of population in the world uses this type of communication by delivering their speech via mouth. The people belonging to hearing impaired community uses sign language for communication. The hearing-impaired people express their thoughts by using

body gestures, mainly hands, arms and facial expressions. Sign language is the communicating way for the people belonging to hearing disabled community. Sign language has been naturally developed by the group of people in different parts of the world based on their habits, tradition, culture, native speaking language and habitats. There exists no universal sign language around the world. The sign languages that are most popular are American Sign Language (ASL), German Sign Language (GSL) and British Sign Language (BSL). Each sign language has their own gestures based on the country's spoken language, culture, tradition and habits. Sign language recognition is an important area that needs to be focused for developing the translation system. The major population with hearing ability is unaware of this sign language gestures. There exists the gap for communication between the hearing-impaired community and the hearing community. The advancement in Computer vision can be incorporated in the translation of Sign language to bridge the gap for communication of deaf-mute people with the external world. Most of the research for recognition system has been done in American Sign Language as it uses mostly single hand gestures. Indian Sign Language recognition system is still in the developing phase as it uses double hand static gestures for alphabets and digits; double hand continuous gestures for words and short sentences. Most of the research in ISL is done in the recognition of alphabets and digits. Only few of the research has been attempted in developing the recognition model for words. This paper focuses on developing an Indian Sign Language recognition system for meaningful short sentences that takes Indian sign gestures as videos. The videos are recorded using the open-source library OpenCV (Open-Source Computer Vision Library) which provides the facility for webcam access. The landmarks of face, pose and hands are extracted from the recorded video using another open-source framework called MediaPipe [16,26,4,6]. The extracted keypoints are fed as input to a neural network LSTM model for training. Using the trained model, the matching short sentence for the given sign gesture is predicted.

2. Related Works

Many recognition systems for the translation of sign language have been developed using Computer vision, Sensor gloves and other technologies. Anderson et al.

[1] explained the steps in developing the interpreter model which includes data acquisition for obtaining input sign gestures, preprocessing for feature extraction and training the data with suitable algorithm to display the output in the form of text or speech.

2.1. Glove Based Recognition System

The existing sensor-based model developed by Heera et al. [12] uses flex sensor which determines voltage of each finger based on the degree of bend with accelerometer and gyroscope to determine the hand position. The determined values are then passed and mapped to the application in Android which is accompanied with the database of ISL words using Bluetooth module. The output is produced in the form of speech when the reading matches. The model built using sensors are more expensive than compared to computer vision system. The improvements for the VirtualSign platform is introduced by Oliveira et al. [18] for bidirectional sign language translation that converts sign to text as well as text to sign using set of gloves and Kinect. This model works well in sign to text but inaccurate predictions are shown in text to sign translation. The model for recognition of ASL alphabets developed by Lee et al. [15] uses Leap motion controller for hand detection which detects hand palm data, hand palm sphere radius, angles between finger and distance between finger position. The features are extracted and the data are trained using LSTM model. The position angle and number of users will affect the accuracy of the model.

2.2. ML Based Recognition System

The focus on building a recognition model for 4 Indian gestures A, 'B', 'C', 'Hello' is executed in [19] by Raheja et al. The captured input video frame is first converted into binary image using thresholding and the noises are removed. Later feature extractions are performed and finally classified using SVM. It is developed only for four gestures and requires long pre-processing for videos. The research paper by Ekbote et al. [11] is based on building a model for Indian sign numerals (0-9). They created a custom dataset for numbers and preprocessed for feature extraction using Shape descriptors, SIFT and HOG. The combination of three feature extractors trained with SVM and HOG trained with ANN provided better results and accuracy. In 2017 the research work proposed by Dehankar et al. [10] discusses about the processing of images

captured in variable background and blurred images in identifying the hand gestures using number of end points and branch points. Architecture for continuous sign language recognition for video input is proposed by Wei et al. [25]. It uses semantic boundary-based reinforcement learning for predicting the output. The development of GUI in [13] is done by taking video input of Indian signs (alphabets and digits). Bag of words is used in finding histogram frequency for image features, extracted using SURF. The extracted features are trained separately with SVM and CNN. The output is displayed in text format as well as speech format. The reverse recognition model is developed by matching the input video with the labels. The model works well only in plain background environment. Tamiruet al. [24] deals in developing the model for Amharic Sign language recognition, where Amharic is a language spoken in Ethiopia. The videos are split into frames using MATLAB in-built function. The segmentation is done using adaptive threshold algorithm and features are extracted using shape feature descriptors and motion feature descriptors and finally trained using two different algorithms ANN and SVM. Das et al. [9] proposed a model for Bangla signs using a combination of four pre-trained CNN models and a random forest classifier. The model is developed using smaller datasets which is not sufficient.

2.3. DL Based Recognition System

Sajanraj et al. [20] developed CNN based model that extracts Regions of Interest from the video frames and displays the output in text format. This real-time recognition is developed only for static numerals. Sruthi et al. [22] proposed a deep learning based approach for alphabet recognition based on static signs. This architecture of static input suits only for digits and alphabets and does not work with words as it takes continuous actions. Bhagat et al. [5] developed the translation model for Indian sign alphabets, numbers and 10 dynamic word gestures. The RGB and depth based image dataset of static alphabets and numbers are trained using CNN. The RGB and depth based video dataset of dynamic words are trained using LSTM. The research paper proposed by Barbhuiya et al. [3] focuses on building the model for American Sign Language (alphabets and digits) using Transfer learning based pre-trained AlexNet and pre-trained VGG16 for feature extraction and classified using SVM. The drawback

of this model is signs with similar gestures are incorrectly predicted. The work proposed by Breland et al. [7] focuses on building a model using embedded system Raspberry Pi connected to thermal camera. The thermal images are trained using CNN. The developed model is bit expensive and works only with numbers. The research paper by Atitallah et al. [2] deals with the recognition of sign language using CNN classification, where the hand signs are identified using EIT imaging process and monitored with Gauss-Newton image reconstruction algorithm with low complexity. This model construction is highly expensive in terms of computation. In 2021, Sharma et al. [21] focused on building an American Sign Language recognition model for 100 words using a Lexicon video dataset. The extracted video frames are reduced to a fixed frame size of 25 and trained using 3DCNN model. A Deepsign model developed by Kothadiya et al. [14] for Indian sign gesture words take video input and uses InceptionResNetV2 for feature extraction. The training model is built using LSTM and GRU layers. This research work focuses on building a translation desktop application for ASL alphabets. In the work of Obi et al. [17], the image dataset collected from Kaggle are processed by the CNN model. The GUI for desktop application is created for displaying the predicted output. It provides better results under plain background and good lighting conditions.

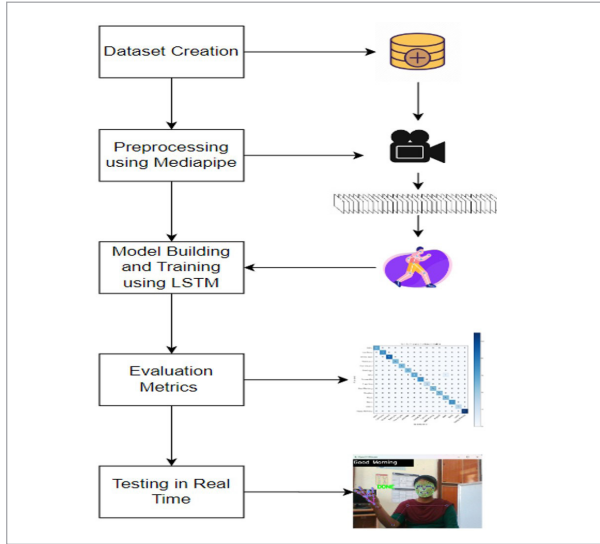
In the existing literature, the interpreter model for translation of sign gestures to text are mostly developed for alphabets and numbers which uses static sign gestures. Only few researchers attempted for dynamic gestures. Our proposed work focuses on developing interpreter model for Indian sign gestures of short sentences which uses continuous dynamic gestures with efficient storage.

3. Proposed Methodology

The custom dataset for Indian sign gestures is taken in form of video using OpenCV, which helps us to access the webcam. The videos are split into frames and keypoints are extracted from each frame using MediaPipe. Each frame stores the hand, pose and facial keypoints. The extracted keypoints are then fed into the LSTM model for training. The model after training is tested and validated for accuracy in real-time. This is the method by which Indian Sign Language Interpreter is

built in real-time. The overall methodology used for building the proposed model is shown in Figure 1.

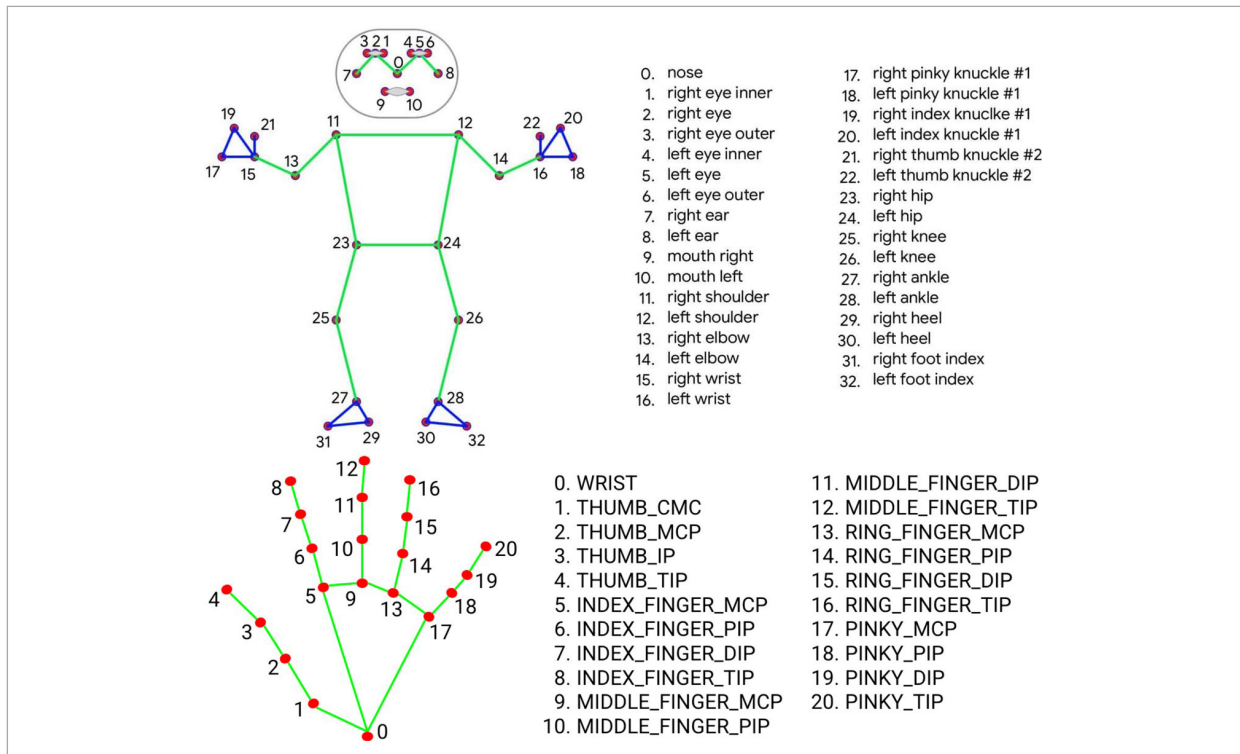
Figure 1
System Design



3.1. Preliminaries

- 1 OpenCV – It is an open-source Python library. It is used for image processing, machine learning, and computer vision related applications. It is used to recognise objects, people, and even human handwriting when processing photos and videos [8].
- 2 MediaPipe – MediaPipe is an open-source framework that performs functions for building pipelines to perform certain tasks of object identification, landmark detection for face, hand and pose. The landmark keypoints for the hand and pose are shown in Figure 2. The major applications of MediaPipe are Object detection, Face landmark detection and segmentation, real-time hand and body pose tracking [16, 26, 4, 6].
- 3 Long Short Term Memory (LSTM) – LSTMs, or Long Short Term Memory networks, are a particular type of RNN that can learn long-term dependencies. They are recurrent units that attempt to “remember” all of the prior knowledge that the network has seen and to “forget” irrelevant material [23].

Figure 2
Pose and Hand Landmarks



3.2. Dataset Creation





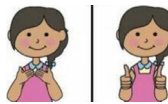


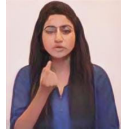
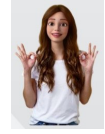





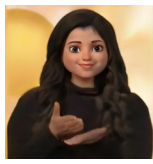
The datasets of Indian Sign Language are available only for alphabets and numbers. The letters and digits of Indian signs use double hand static gestures, so the datasets are mostly available in image format. The proposed work focuses on building a recognition system for short sentences which uses double hand dynamic gestures and facial expressions. So, the dataset must be given in the format of video and it must be created. The custom-built video datasets are recorded using OpenCV, a framework that provides algorithms and infrastructure for Computer vision applications. It also provides support for reading, writing and capturing video files. The videos are captured by OpenCV using 'Cv2.VideoCapture' class in python either in real-time or recorded videos can also be used by specifying the path of storage. We have captured the video in real-time and the folder for each action contains 30 videos. Each action video is split into 30 frames. So, our dataset contains 30 videos for every 15 actions, totally 450 videos. Each of the video is split into 30 frames. The recognition model developed for the short sentences are listed in Table 1.

3.3. Preprocessing

The major step in pre-processing is the extraction of keypoints. In this work, MediaPipe is used for the extraction of hand, face and pose landmarks. It recognizes the face, pose and hand landmarks using in-built MediaPipe function `mp.solutions.holistics`. This pipeline contains framework that recognizes face, left hand, right hand and pose with 468, 21, 21 and 33 landmarks respectively. MediaPipe works only on RGB (Red, Green, Blue) format images. The video captured from OpenCV is in the format of BGR (Blue, Green, Red). So, color conversion is necessary to work with MediaPipe, the image frames extracted from the videos are converted from BGR to RGB using the function `cv2.cvtColor` and writable status is set to false for saving memory. The converted images are then passed for landmark extraction, each landmark is represented in x, y, z format, where x denoting the position of x-axis, y denoting the position of y-axis and z denoting the relative distance of camera. The land marks of face, right hand, left hand and pose are extracted with x, y, z values in addition pose landmark has an extra value denoting the visibility. Thus, totally for each frame 1662 landmark values are ex-

Table 1

Words / Actions

| | | |
|--|--|---|
|  Hello |  I am Sorry |  All the best |
|  Thank you |  How are you |  Greetings |
|  Help |  Excuse Me |  I am fine |
|  Good morning |  Hospital |  Book |
|  Sleep |  Listen |  Happy Birthday |

tracted, 468 keypoints for face landmarks and 21 keypoints for right hand landmarks, 21 keypoints for left hand landmarks with x, y, z values and 33 keypoints for face landmarks with x, y, z and visibility values ($468*3+21*3+21*3+33*4 = 1662$) and stored in numpy arrays. As an error handling mechanism, the non-detected landmark points are replaced with zero. The labels for the action are converted into binary format for easy representation. The training and testing data are split as 80% and 20%.

3.4. Training

We have trained the model with LSTM (Long Short-Term Memory) type of Recurrent Neural Network, as it works well with video dataset with the feature of

learning long-term dependency. A Sequential model is built with 3 LSTM and 3 dense layers. The first LSTM layer consists of 64 neurons with Relu activation function, second LSTM layer consists of 128 neurons with Relu activation function and third LSTM layer consists of 64 neurons with the same Relu activation function. The first two dense layers contain 64 and 32 neurons with Relu activation function and last dense layer with Softmax activation function. The optimizer and loss function used are Adam and Categorical cross entropy. The model is trained with 30 videos for each action having 30 frames for each video storing 1662 landmarks in each frame for 200 epochs. The epoch size of 200 to achieve better categorical accuracy is fixed by training the model at different epoch sizes. The comparison of different epochs with the categorical accuracy and loss are shown in Table 2.

Table 2

Comparison of evaluation metrics at different epochs

| Epochs | loss | categorical accuracy |
|--------|--------|----------------------|
| 50 | 0.4550 | 0.7639 |
| 100 | 0.2378 | 0.9083 |
| 150 | 0.1461 | 0.9389 |
| 200 | 0.0315 | 0.9972 |

3.5. Environmental Setup

We used HP Elite Desk 800 G4, 3.00 GHZ Processor, 16 GB RAM 1TB HDD, NVIDIA GeForce GTX 1660 for running the recognition model which imports the libraries like MediaPipe, OpenCV, Tensorflow, Numpy, Matplotlib and Scikit-learn.

4. Results and Discussions

4.1. Model Training

We have trained the model using keypoints obtained from MediaPipe which consist of landmarks for hands, pose and face. Each frame consists of 1662 keypoints. So, for each video 30×1662 keypoints and for every action $30 \times 30 \times 1662$ keypoints are trained. The epoch size of 200 for training the array of extracted keypoints was fixed by comparing the accuracy at various epoch levels of 50, 100, 150 and 200. The 200 epoch size mod-

el produced more accuracy when compared to others. The overfitting can be avoided by using the dropout layer. We also trained the model using a LSTM dropout layer of 0.3. The accuracy achieved by the model with and without dropout was nearly the same. The achieved accuracy is purely based on performance. For the larger dataset to avoid overfitting, the LSTM layer with dropout can be added to achieve better performance of the model. We achieved the categorical training accuracy of 99.72% for 15 words by training the model with extracted keypoints of $30 \times 30 \times 1662$ for each word. Figure 3 shows the graph of categorical accuracy, where x-axis indicates the epoch size and y-axis indicates the accuracy values and Figure 4 shows the loss graph where x-axis indicates the epoch size and y-axis indicates the loss values. In accuracy graph, with increase in epoch size there occurs the increase of accuracy values, whereas in lose graph the loss value decreases with increase in epoch size.

Figure 3

Categorical Accuracy Graph

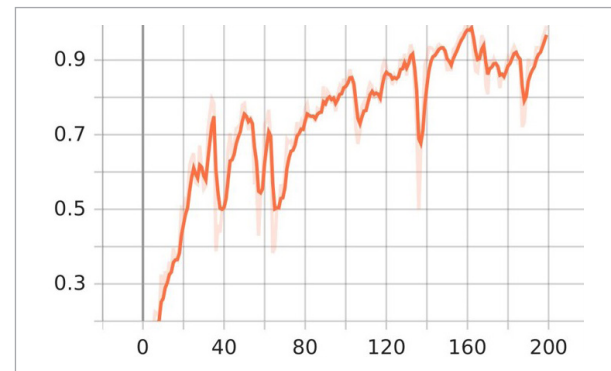
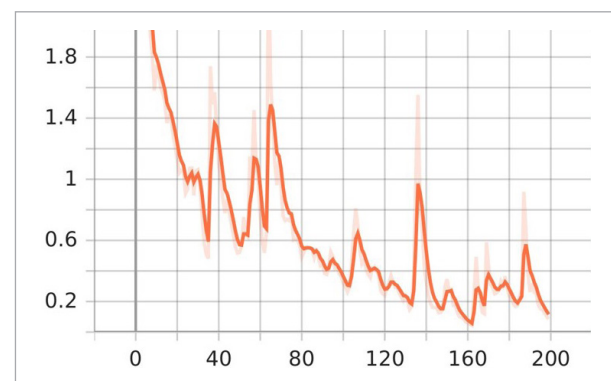


Figure 4

Loss Graph



Happy Birthday. The individual class precision, recall and F1-Score values are calculated and shown in Table 3. The evaluation of accuracy by varying different parameters results in overall training and testing ac-

curacy of 99.17% and 97.78%. The 100% accuracy is achieved for 12 classes (Hello, I am sorry, All the best, Thank you, How are you, Excuse me, I am fine, Good morning, Hospital, Book, Sleep, Listen and Happy Birthday) and 80% is achieved for 2 classes (Greetings and Help). The sign gesture of Greetings and Help contains flat surfaces in vertical and horizontal manner. The MediaPipe was not able to extract keypoints clearly from flat surface of hand as major portion of hand is not visible to the camera. So the output for the gesture help is sometimes incorrectly predicted as book and Greetings is predicted as Hospital. The problem of incorrect prediction is produced, as MediaPipe detects only few keypoints on flat surface of the hand and contains more hidden points.

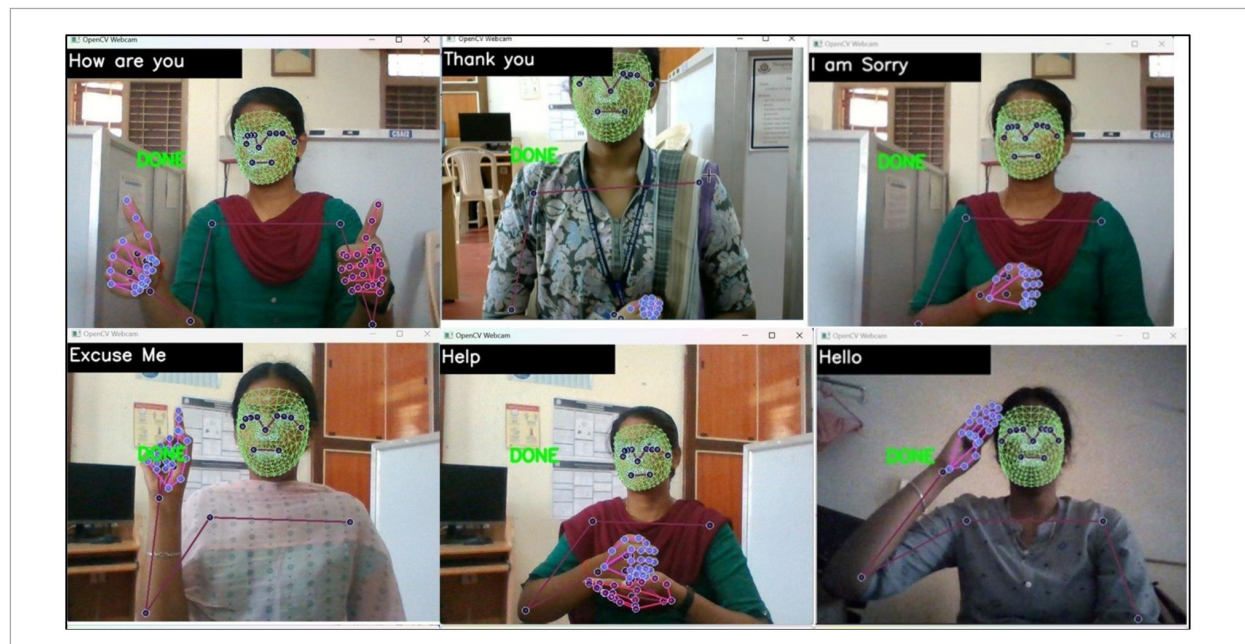
Table 3
Individual Class Precision, Recall and F1-Score values

| label | class | precision | recall | f1-score |
|-------|----------------|-----------|--------|----------|
| 0 | Hello | 1.00 | 1.00 | 1.00 |
| 1 | I am Sorry | 1.00 | 1.00 | 1.00 |
| 2 | All the best | 1.00 | 1.00 | 1.00 |
| 3 | Thank you | 1.00 | 1.00 | 1.00 |
| 4 | How are you | 1.00 | 1.00 | 1.00 |
| 5 | Greetings | 1.00 | 0.80 | 0.89 |
| 6 | Help | 1.00 | 0.83 | 0.91 |
| 7 | Excuse Me | 1.00 | 1.00 | 1.00 |
| 8 | I am fine | 1.00 | 1.00 | 1.00 |
| 9 | Good morning | 1.00 | 1.00 | 1.00 |
| 10 | Hospital | 0.83 | 1.00 | 0.91 |
| 11 | Book | 0.75 | 1.00 | 0.86 |
| 12 | Sleep | 1.00 | 1.00 | 1.00 |
| 13 | Listen | 1.00 | 1.00 | 1.00 |
| 14 | Happy Birthday | 1.00 | 1.00 | 1.00 |

4.3. Prediction Results

The output results for the actions ‘Hello’, ‘I am Sorry’, ‘Thank you’, ‘All the best’, ‘How are you’, ‘Greetings’, ‘Help’, ‘Excuse Me’, ‘I am fine’, ‘Good morning’, ‘Hospital’, ‘Book’, ‘Sleep’, ‘Listen’, ‘Happy Birthday’ are taken under varying background conditions and achieved good results. The model is also tested with different people other than dataset creators, it also achieved good results and the output are almost correctly predicted. The sample output results for the short sentences with continuous gestures are shown in Figure 6.

Figure 6
Sample results screenshot for short sentences with continuous gestures



4.4. Comparison

4.4.1. Comparison with Similar Algorithms

The keypoint array of face, pose and hands extracted from the created video dataset are trained using different algorithms like KNN, LSTM and Bidirectional LSTM. These algorithms are suitable for training the array of keypoints.

The Table 4 shows the comparison of different algorithms trained using the same dataset. The LSTM model achieved better accuracy of 97.78% compared to KNN and Bidirectional model. The KNN and Bidirectional model achieved an accuracy nearer to 92%. From the Table 4, it is inferred that LSTM model achieved more accuracy for the training of keypoints.

Table 4

Comparison of our proposed method with similar algorithms using the same dataset

| algorithm | precision | recall | f1-score | Accuracy (%) |
|--------------------|-----------|--------|----------|--------------|
| KNN | 0.93 | 0.90 | 0.91 | 92.22 |
| LSTM | 0.97 | 0.98 | 0.97 | 97.78 |
| Bidirectional LSTM | 0.92 | 0.94 | 0.92 | 92.22 |

4.4.2. Comparison with Existing Methods

In most of the literature, the recognition models are built for American Sign Language. In Indian Sign Language recognition, development of models is mostly done for alphabets and numerals, few models are created for words. The model proposed in [5,14] are based on recognition of Indian sign gestures for words.

The research work by Bhagat et al. [5] focuses on building a recognition model for ISL alphabets and words. Two separate models are built, one for translation of al-

phabets and numbers using static image gesture dataset and other one is built for recognition of words in Indian Sign Language using video dataset. The 10 words chosen are Lock, Wi-fi, Aeroplane, License, Local, Low, machine, Mall, maths, Win. The dataset is self-created by capturing dynamic videos from Microsoft Kinect RGBD camera and trained using convolutional LSTM. The dynamic depth videos, RGB videos and both depth + RGB videos shows an accuracy of 97.52%, 98.11%, 99.08% with training and 76.4%, 77.6%, 78.3% with testing. The model developed for the recognition of continuous signs in ISL using LSTM produced lesser accuracy compared to our model.

The recognition model developed by Kothadiya et al. [14] is based on building a translation model for words in Indian Sign language. The video dataset is created for 11 words namely Hello, Bye, Morning, Good, Nice, House, Thank you, Welcome, Yes, No, Work. The features are extracted from the frames using Inception-ResNetV2 and stored in numpy arrays. The feature vectors are trained with LSTM model and achieved an overall accuracy of 95% for continuous Indian signs which is less when compared to our proposed model.

The proposed model is developed for the recognition of 15 words in ISL trained with LSTM and achieved a training and testing accuracy of 99.17% and 97.78%. We used the same algorithm for training the model, the different thing that is used in our work is extraction of keypoints using MediaPipe. The MediaPipe usage for keypoint extraction has achieved better accuracy and also for training the model, those keypoints are stored in numpy arrays. This achieves efficient storage and increases the computational speed.

Table 5 shows the comparison of our proposed method with the existing methods [5, 14]. The table clearly specifies that the proposed model outperforms the existing works in terms of accuracy with better storage and computational efficiency.

Table 5

Comparison of our proposed method with the existing methods

| Model | Dataset | Pre-processing | Training | Accuracy (in %) |
|---------------------|----------|--|----------|-----------------|
| Proposed Model | 15 words | MediaPipe for keypoints extraction | LSTM | 97.78 |
| Existing Model [5] | 10 words | One to one mapping of the pixels of depth and RGB pixels using 3D construction and affine transformation | LSTM | 77 |
| Existing Model [14] | 11 words | InceptionResNetV2 for feature extraction | LSTM | 95 |

5. Conclusion and Future Works

In this work, we proposed a translation model for the interpretation of Indian Sign Language for short sentences 'Hello, Thank you, All the best, I am Sorry, How are you, I am fine, Good morning, Help, Excuse me, Greetings, Hospital, Listen, Sleep, Book and Happy Birthday' using custom-built video dataset. The gestures are captured using OpenCV and keypoints for hand, pose and face landmarks are extracted from each image using MediaPipe. The keypoints are trained using LSTM and achieved better results and accuracy of 97.78% for the prediction of continuous dynamic Indian sign gestures compared to the existing works. The work can be extended for other short sentences and long sentences with output displayed

in both text and speech format. MediaPipe was not able to detect keypoints from the flat surface which contains hidden points. Thus, the gestures with flat horizontal surfaces cannot be accurately predicted. The hidden points for flat surfaces must be handled in future.

Funding

This work did not receive any funding.

Data availability

The datasets will be shared upon request to the corresponding author.

Conflict of Interest

None.

References

- Anderson, R., Wiryana, F., Ariesta, M. C., Kusuma, G. P. Sign Language Recognition Application Systems for Deaf-Mute People: A Review Based on Input-Process-Output. *Procedia Computer Science*, 2017, 116, 441-448. <https://doi.org/10.1016/j.procs.2017.10.028>
- Atitallah, B. B., Hu, Z., Bouchaala, D., Hussain, M. A., Ismail, A., Derbel, N., Kanoun, O. Hand Sign Recognition System Based on EIT Imaging and Robust CNN Classification. *IEEE Sensors Journal*, 2022, 22(2), 1729-1737. <https://doi.org/10.1109/JSEN.2021.3130982>
- Barbhuiya, A. A., Karsh, R. K., Jain, R. CNN Based Feature Extraction and Classification for Sign Language. *Multimedia Tools and Applications*, 2021, 80(2), 3051-3069. <https://doi.org/10.1007/s11042-020-09829-y>
- Bazarevsky, V., Grishchenko, I., Raveendran, K., Zhu, T., Zhang, F., Grundmann, M. Blazepose: On-Device Real-Time Body Pose Tracking. *arXiv Preprint arXiv:2006.10204*, 2020.
- Bhagat, N. K., Vishnusai, Y., Rathna, G. N. Indian Sign Language Gesture Recognition Using Image Processing and Deep Learning. *Proceedings of the 2019 Digital Image Computing: Techniques and Applications (DICTA)*, December 2019, 1-8. <https://doi.org/10.1109/DICTA47822.2019.8945850>
- Boesch, G. MediaPipe: Google's Open-Source Framework for ML Solutions (2023 Guide). January 2023, Accessed 2023, January 31. <https://viso.ai/computer-vision/mediapipe/>
- Breland, D. S., Skriubakken, S. B., Dayal, A., Jha, A., Yalavarthy, P. K., Cenkeramaddi, L. R. Deep Learning-Based Sign Language Digits Recognition from Thermal Images with Edge Computing System. *IEEE Sensors Journal*, 2021, 21(9), 10445-10453. <https://doi.org/10.1109/JSEN.2021.3061608>
- Culjak, I., Abram, D., Pribanic, T., Dzapov, H., Cifrek, M. A Brief Introduction to OpenCV. *2012 Proceedings of the 35th International Convention MIPRO*, May 2012, 1725-1730.
- Das, S., Imtiaz, M. S., Neom, N. H., Siddique, N., Wang, H. A Hybrid Approach for Bangla Sign Language Recognition Using Deep Transfer Learning Model with Random Forest Classifier. *Expert Systems with Applications*, 2023, 213, 118914. <https://doi.org/10.1016/j.eswa.2022.118914>
- Dehankar, A. V., Jain, S., Thakare, V. M. Using AEPI Method for Hand Gesture Recognition in Varying Background and Blurred Images. *Proceedings of the 2017 International Conference of Electronics, Communication and Aerospace Technology (ICECA)*, April 2017, 1, 404-409. <https://doi.org/10.1109/ICECA.2017.8203715>
- Ekbote, J., Joshi, M. Indian Sign Language Recognition Using ANN and SVM Classifiers. *Proceedings of the 2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, March 2017, 1-5. <https://doi.org/10.1109/ICIIECS.2017.8276111>
- Heera, S. Y., Murthy, M. K., Sravanti, V. S., Salvi, S. Talking Hands-An Indian Sign Language to Speech Translating Gloves. *Proceedings of the 2017 Interna-*

- tional Conference on Innovative Mechanisms for Industry Applications (ICIMIA), February 2017, 746-751. <https://doi.org/10.1109/ICIMIA.2017.7975564>
13. Katoch, S., Singh, V., Tiwary, U. S. Indian Sign Language Recognition System Using SURF with SVM and CNN. *Array*, 2022, 14, 100141. <https://doi.org/10.1016/j.array.2022.100141>
 14. Kothadiya, D., Bhatt, C., Sapariya, K., Patel, K., Gil-González, A. B., Corchado, J. M. Deepsign: Sign Language Detection and Recognition Using Deep Learning. *Electronics*, 2022, 11(11), 1780. <https://doi.org/10.3390/electronics11111780>
 15. Lee, C. K., Ng, K. K., Chen, C. H., Lau, H. C., Chung, S. Y., Tsoi, T. American Sign Language Recognition and Training Method with Recurrent Neural Network. *Expert Systems with Applications*, 2021, 167, 114403. <https://doi.org/10.1016/j.eswa.2020.114403>
 16. Lugaresi, C., Tang, J., Nash, H., McClanahan, C., Ubowaja, E., Hays, M., Zhang, F., Chang, C. L., Yong, M. G., Lee, J., Chang, W. T., Grundmann, M. MediaPipe: A Framework for Building Perception Pipelines. *arXiv Preprint arXiv:1906.08172*, 2019.
 17. Obi, Y., Claudio, K. S., Budiman, V. M., Achmad, S., Kurniawan, A. Sign Language Recognition System for Communicating to People with Disabilities. *Procedia Computer Science*, 2023, 216, 13-20. <https://doi.org/10.1016/j.procs.2022.12.106>
 18. Oliveira, T., Escudeiro, N., Escudeiro, P., Rocha, E., Barbosa, F. M. The Virtualsign Channel for the Communication Between Deaf and Hearing Users. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, 2019, 14(4), 188-195. <https://doi.org/10.1109/RITA.2019.2952270>
 19. Raheja, J. L., Mishra, A., Chaudhary, A. Indian Sign Language Recognition Using SVM. *Pattern Recognition and Image Analysis*, 2016, 26(2), 434-441. <https://doi.org/10.1134/S1054661816020164>
 20. Sajanraj, T. D., Beena, M. V. Indian Sign Language Numeral Recognition Using Region of Interest Convolutional Neural Network. *Proceedings of the 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT)*, April 2018, 636-640. <https://doi.org/10.1109/ICICCT.2018.8473141>
 21. Sharma, S., Kumar, K. ASL-3DCNN: American Sign Language Recognition Technique Using 3-D Convolutional Neural Networks. *Multimedia Tools and Applications*, 2021, 80(17), 26319-26331. <https://doi.org/10.1007/s11042-021-10768-5>
 22. Sruthi, C. J., Lijiya, A. Signet: A Deep Learning Based Indian Sign Language Recognition System. *Proceedings of the 2019 International Conference on Communication and Signal Processing (ICCSP)*, April 2019, 0596-0600. <https://doi.org/10.1109/ICCSP.2019.8698006>
 23. Staudemeyer, R. C., Morris, E. R. Understanding LSTM-A Tutorial into Long Short-Term Memory Recurrent Neural Networks. *arXiv Preprint arXiv:1909.09586*, 2019.
 24. Tamiru, N. K., Tekeba, M., Salau, A. O. Recognition of Amharic Sign Language with Amharic Alphabet Signs Using ANN and SVM. *The Visual Computer*, 2022, 38(5), 1703-1718. <https://doi.org/10.1007/s00371-021-02099-1>
 25. Wei, C., Zhao, J., Zhou, W., Li, H. Semantic Boundary Detection with Reinforcement Learning for Continuous Sign Language Recognition. *IEEE Transactions on Circuits and Systems for Video Technology*, 2021, 31(3), 1138-1149. <https://doi.org/10.1109/TCSVT.2020.2999384>
 26. Zhang, F., Bazarevsky, V., Vakunov, A., Tkachenka, A., Sung, G., Chang, C. L., Grundmann, M. MediaPipe Hands: On-Device Real-Time Hand Tracking. *arXiv Preprint arXiv:2006.10214*, 2020.

