Journal of Scientific & Industrial Research
Vol. 82, July 2023, pp. 745-755
DOI: 10.56042/jsir.v82i07.2375

Sign Language Recognition using Deep CNN with Normalised Keyframe Extraction and Prediction using LSTM

Jayanthi P, Ponsy R K Sathia Bhama* & B Madhubalasri
Department of Computer Technology, MIT, Anna University, Chennai 600 044, Tamilnadu, India

Received 01 April 2021; revised 18 May 2023; accepted 08 June 2023

Sign Language Recognition (SLR) targets interpreting the signs so as to facilitate communication between hearing or speaking disabled people and normal people. This makes communication between normal people and signers effective and seamless. The scarcely available key information regarding the gestures is the key to recognise the signs. To implement continuous sign language gesture recognition, gestures are identified from the video using Deep Convolutional Neural Network. Recurrent Neural Network- Long Short-Term Memory verifies the semantics of the gesture sequence, which eventually will be converted into speech. The problem of constructing meaningful sentences from continuous gestures inspired the proposed system to develop a model based on it. The model is designed to increase the effectiveness of the classification by processing only the principal elements. The keyframes are identified and processed for classification. Validation of sentences can be done O(N). The sentences are converted into voiceover to have elegant communication between impaired and normal people. The model obtained an accuracy of 89.24% while training over Convolutional Neural Network to detect gestures and performed better than other pre-trained models and an accuracy of 89.99% while training over Recurrent Neural Network- Long Short-Term Memory to predict the next word using grammar phrases. This keyframe-to-voice conversion, forming proper sentences, enthrals people to have harmonious communication.

Keywords: Deaf-mute people, Gesture recognition, Indian sign language, Relationship signs, Signer

Introduction

People with hearing impairments communicate with the help of sign language. Their linguistic properties are the same as those of spoken languages, performed using hand movements and facial expressions.[1,2] Conversations can only be made if there is a mutual understanding of the language between the parties involved. Signers use hand gestures to convey their thoughts, which is difficult for normal human beings to understand. Sign Language Recognition (SLR) is very effective in converting signs into a form that is recognizable by others. There are many techniques incorporated in deep learning to carry out gesture detection[3,4], and they are focused on extracting complex features from the raw input. Deep Learning (DL) can be thought of as a way of bringing out predictive analysis[5] into picture. Neural networks are generally used to train huge amounts of data in an efficient manner. All layers are interdependent on each other. The model can be trained to recognize classes of gestures. The gestures in the frames are classified according to the features extracted. But gesture detection and classification processes take a lot of time when working with a normal system. Graphics Processing Unit (GPU) can be used, which reduces the processing time and also provides the facility to handle information in larger memory sizes. Leap Motion is a popular instrument to capture hand movements[6] in a real-time environment. The device, which is very similar to a mouse, helps in getting hand motions as input without any contact with the device. The controller can cover an area of about one metre by using certain wavelengths of infrared LEDs and IR cameras. The signals are then transmitted to the personal desktop through a USB, where they are further processed. Another variant for sequential modelling will be Hidden Markov Model[7] (HMM), which comprises a system with unobservable hidden states whose behaviour is analyzed by the study of the model that is dependent on it. HMM mainly aims at deriving the output sequences with the maximum likelihood from the given set of sequences. The HMM is accompanied by the Viterbi algorithm in decoding the probabilities of the test sequences[8] generated. One of the toughest aspects of image processing is constructing the boundary box around the gesture.

*Author for Correspondence
E-mail: ponsy@mitindia.edu

Computing the rectangular boundary coordinates[9] representing the performed gesture is strenuous compared to classifying the gestures. The hand gesture is extracted using the correct dimensionality filter matrix and evaluated further. The segmentation of the image helps refine the detection process.[10–12] It is mainly used to locate a particular object in an image. The sign language recognition system uses a segmented probabilistic approach[10] to recognise the continuous sign sequence. This works by computing the edges and points that represent hands and further process them. Other techniques for sign language recognition include SVM classification[13] where the gestures are identified into a bag of words and classified with a SVM model whereas ANN uses parallel computational nodes to identify and classify gestures[14] and Self Organizing Feature Maps employ HMM to classify the gestures.[15] Eigenvectors of gestures can be processed through PCA to reduce the dimension and further classify them.[16] Commonly, for all approaches, the gestures are reduced to useful information, which is finally classified using a model. Processing all frames from the recording is 1) Time inefficient, 2) Memory consuming and 3) Misleading with duplicate data. Extracting only a reasonable number of frames and evaluating them to identify gestures is preferable. The recorder is expected to make nugatory hand movements between gestures. The use of Root Mean Square Error (RMSE) eliminates unnecessary frames using image comparison. Further, keyframes are extracted by normalizing the frames with respect to time period and withdrawing frames closer to the mean. These frames are processed to detect gestures in a transitive manner. The frames can be passed through Deep CNN to detect the gesture. The comprehensive content of the sentence might not be performed or detected. A faultless version of the continuous gestures can be assembled by evaluating them using RNN-LSTM. This sentence is read out to start a conversation.

Convolutional Neural Network

Gesture detection and classification are important aspects to be carried out in sign language recognition. The extracted frames were processed by CNN to detect 3D motion oriented sign language recognition.[17–20] The spatio-temporal features were detected as Joint Angular Displacement Maps (JADMs)[16] which are further processed and classified. JADMs record the distance between the angular joints in hands to depict individual gestures. Bao *et al*.,[17] discussed how the detection of a particular gesture can be carried out without any segmentation or bounding boxes. Abstract features can be derived from the image by increasing the number of convolutional layers.[15–19] The convolutional layers are followed by a set of max pooling layers and fully connected layers interlaced with the ReLU activation unit. Overfitting is prevented by adding a set of dropout layers. The CNN architecture prefers low window size[15,16], initially to increase the accuracy of the model. From the final pooling layers, the output is passed on to the fully connected layers where flattening of the data takes place. The softmax layer is added to assign probability to the gesture classes, and the gesture with the maximum probability is chosen. The learning rate has been set to a constant value until the accuracy becomes stable.[21]

Hidden Markov Model

Several Sign Language Recognition systems have been built on the Hidden Markov Model approach and its variants.[15,22–24] The HMM describes a static mode which is a probability distribution. Glove based Sign Language Recognition using HMM is used to identify continuous gestures.[22] Multidimensional input vector input processed through the HMM and is classified using their probability.[23] Modified HMM can also be used to detect and classify gestures.[24]

Leap Motion Sensor

Leap Motion Sensor is an accurate finger point-detecting sensor that can be used to monitor the movements of hands depicting gestures. This can very well be used to detect the gesture.[25,26] This controller consists of an API with which the direction and length of fingers in hands are being recorded. The variation in their measures is used to evaluate and identify the gesture. Bayesian classification predicts the gestures by comparing the hand movements using leap motion sensors[27,28] and leap motion can record continuous signs[29] that can be trained over the CNN network and include the LSTM network to predict the sentence.

Multimodal Classification

Multimodal solutions are also common among Sign Language Recognition systems. The modals either work sequentially or in parallel, depending on the processing technique. This system is beneficial in the way that more than one technology is involved in the recognition process. Convolution Neural Network (CNN) along with Data Motion Maps (DMM) and

Trajectory methods run in parallel to evaluate the output[30] and the best predicted outcome is processed through the LSTM network to predict for a long term. On the other hand, the processing can be sequential, where the data can be processed by sending it through a 3-D CNN and then through a convolutional LSTM network.[31] The networks being used are trained to obtain maximum accuracy. From the results of the study as a whole, it can be seen that gestures can be recorded in a variety of ways, but cameras are the best option because they don't require any additional hardware. Since the keyframe extraction improves prediction, proposed system suggests a normalised keyframe extraction-based system for classifying sign language gestures. The proposed method is aimed to predict sentences based on the gesture prediction, although there is relatively little research on continuous word prediction after gesture recognition.

Materials and Methods

Sign Language Convertor

The recordings of Indian Sign Language depicting various relationships performed by the students of MIT-Anna University, India are shown in Fig. 1, keyframes representing gesture information are extracted and compressed to be processed by the CNN layers. The architecture of the proposed system is depicted in Fig. 2.

The embedded patterns of hand gestures present in the keyframes are recognized using filter matrices moving across the image. Multi-class classification of gestures is done to form sentences, and the RNN-LSTM network is modelled to train over grammar phrases created with gesture classes.

Keyframe Extraction

Processing any video involves complicated issues. A 3-second video may comprise more than 100 frames, which is both memory and power consuming. Resource utilization is one of the major factors in solving a problem. Also, the frames contain a lot of redundant information. It will be impractical to process every frame and wait to predict one gesture. In order to combat these challenges, only a selected set of frames is extracted and processed in such a way that there is no loss of information.[32] Images extracted from the video have two dimensions, with RGB pixels representing the height and width. Background de-noising is an important part of image processing. The extracted frames not only contain useful information that can be used for classification but also contain unwanted information. Unnecessary background information pertaining to the recorder is eliminated for further processing. De-noising does not lead to any kind of misclassification. On the other hand, it enhances the gesture recognition process. Video

Fig. 1 — Hand gestures representing relationships in Indian sign Language system recorded by three performers for 15 signs each

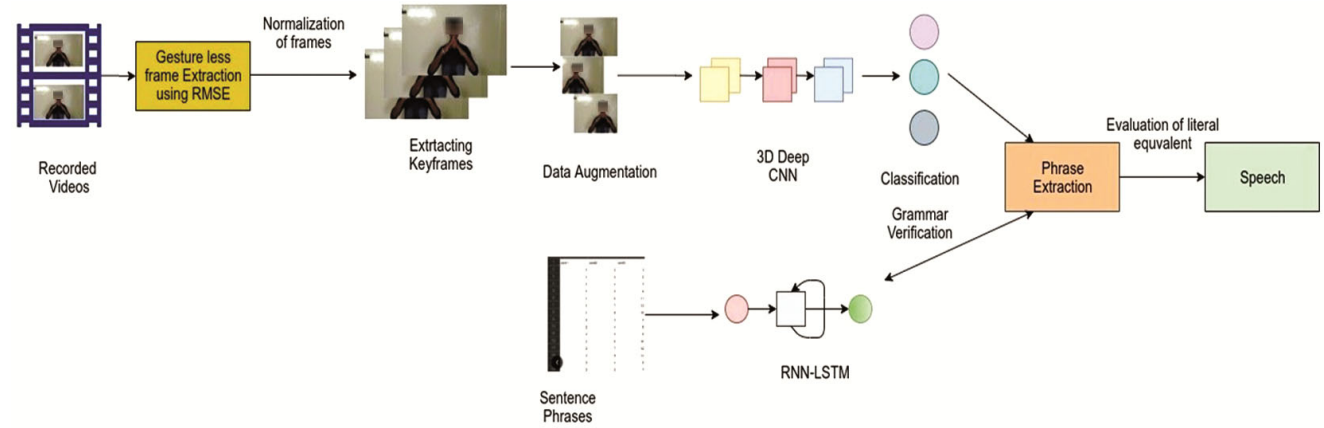

Fig. 2 — Architecture of Sign Language Recognition System processing the videos independently through Deep CNN and verifying the phrases from the trained model of RNN with sentence phrases and converting to audio

consists of certain frames that represent the rest position of the recorder between the gestures or on completion. These frames are of no use in further modelling. On the other hand, the presence of such unwanted frames can increase ambiguity during gesture detection and classification. In order to avoid such uncertainties, frames representing the rest position of the recorder are eliminated and frames representing a gesture are separated. The elimination of frames representing the rest position of the recorder is done using Root Mean Square Error (RMSE) by similarity comparison method computed as follows:

$$RMS_{Value} = \sqrt{\frac{\sum_{i=1}^{n}(y_i - x_i)^2}{n}} \quad \ldots (1)$$

For every pixel in the frame, the corresponding pixel from the static position image is chosen, and their difference is computed. This difference in value is squared to give more importance to even smaller differences. It is divided by n, representing the total number of pixels in the image, and the square root of this value represents error as in Eq. 1. In order to perform the comparison, unique frames representing the static positions of the recorders are stored. Frames in the video are compared with the unique frame, and the RMSE value is calculated for each of those frames. If the error value is low, then those frames represent gesture less frames. This can be used to separate individual gestures in a continuous video. The videos are normalized over the time period into frames. This process is done in order to pick the minimum frames to be processed in order to identify the gesture. Only those frames that remain close to the mean value are selected as keyframes. The frames on the extreme ends of the curve are eliminated because of their clumsiness. Those frames are not capable of producing any significant results because they contain incomplete gesture data. The frames on the left and right extremes indicate the starting and finishing of a gesture, respectively, and including those frames will not contribute much. Thus, only frames that are of high importance and contain meaningful information are picked for further processing. The frames are extracted at a constant five-frame distance on either side of the mean. The keyframes extracted are augmented in a translative manner to increase the training data without actually recording it. This process helps in getting an adaptive model. The efficiency of the model depends on the amount of data with which it is trained, and having versatility can improve its performance. The technique involves moving the image along an X or Y axis. This modification can ease the detection of gestures even if the performer performing the gesture is not present in the centre of the frame. The frames are pushed one-eighth of the width pixels to the left and also to the right. In this way, two augmented images are obtained from a single frame. When a frame is moved left or right, the empty space on the opposite side is compensated using the filling technique for the loss of pixels. This filling can be done using a colour that matches the background pixels in the case of a simple background. Gaussian blurring is done to increase the fineness of the sign by removing the noises recorded along with the video. The noise happens when there is a change in the intensity of light, a variation in the temperature, etc. The videos are recorded on an average-quality lens with noise. The noiseless pixel value acts as an epicentre and affects the surrounding pixels with its value. In this way, the noisy pixels that are present in the image will be affected by their neighbouring noiseless pixels, and noise will be eliminated.

Sentence Prediction

The frames extracted from the videos are shrunk for easier computation. These are passed through a series of convolutional layers for training. The filters help determine the presence of certain features or patterns in the input image. It is usually expressed as a matrix of dimensions smaller than those of the original image through a series of convolutional layers for training. The filters help determine the presence of certain features or patterns in the input image. It is usually expressed as a matrix of dimensions smaller than those of the original image. The model is built layer by layer sequentially, i.e., the output of a layer is passed as input to the next layer in order. The filters help in transitively determining the features across the image without any bounding boxes. The first layer starts with stride value three. The stride value represents the number of pixels a convolutional filter moves, and when the stride value is 1, the filter moves by one step, and so on. A larger stride value is chosen in the initial layers to reduce the computation and improve efficiency. Later, stride value is decreased to note all details in order to not miss any detail. The layer's input data are expected to widen their range over layers. An internal covariate shift occurs when weight values are modified after each epoch. This range keeps on expanding, making

training a forever process. In order to eliminate this hitch, batch normalization is done. Batch Normalization allows a layer to learn about the other layers, especially the hidden layers, and normalise the input values. The activation unit used for the nodes is ReLU. It does not activate all the neurons at the same time; instead, it converts all the negative input to zero, and the corresponding neurons do not pass on to the next layer. Only those positive values move as input to consecutive layers. Images are expected to contain non-linear features in the case of hand gestures. To make these features linear, the negative output values are eliminated and made linear.

The max pooling layer appended to the convolutional layer reduces the number of features and computations in the network by reducing the spatial size of the network. The max pooling layer takes out only the maximum from the pool. This is done with the help of filters sliding through the pixel values. This will actually down sample the network, thereby compressing the features for further computation. Batch normalization is also used with pooling layers to normalize and reduce the range of input for the next layer. This helps in drastically decreasing the number of training epochs and also helps to quicken the training process by deep CNN. Usually, the use of higher learning rates increases the stability of the neural networks, but higher learning rates can also lead to explosions because of the change in the gradients to a level that cannot be handled. Batch normalization can be used in order to avoid such explosions. Convolutional and Max-Pooling layers are placed alternately while developing the model. When convolutional layers are placed next to each other, the computational complexity increases, and the time taken for computation is very high. Alternating with the max pooling layer will reduce the size gradually, making it easy for computation. The complete architecture of the CNN model is shown in Fig. 3, where the features are extracted and will be available in the last max-pooling layer. The classification process begins by flattening the output from the convolutional layers. The three-dimensional array of features is converted into a vector that is to be fed into the fully connected layers well depicted in Fig. 3, which starts at the max pooling layer. Overfitting is prevented by picking up the nodes randomly and setting their output to zero using the Dropout mechanism, which is a regularization technique that reduces the complexity of the model. It reduces the interdependent learning among the neurons and helps the neural network learn about more robust features for computation.

The fully connected layers, along with the dropout mechanism, are repeated until the values are reduced to output class values. The softmax function, as a last layer, assigns a certain probability to each of the

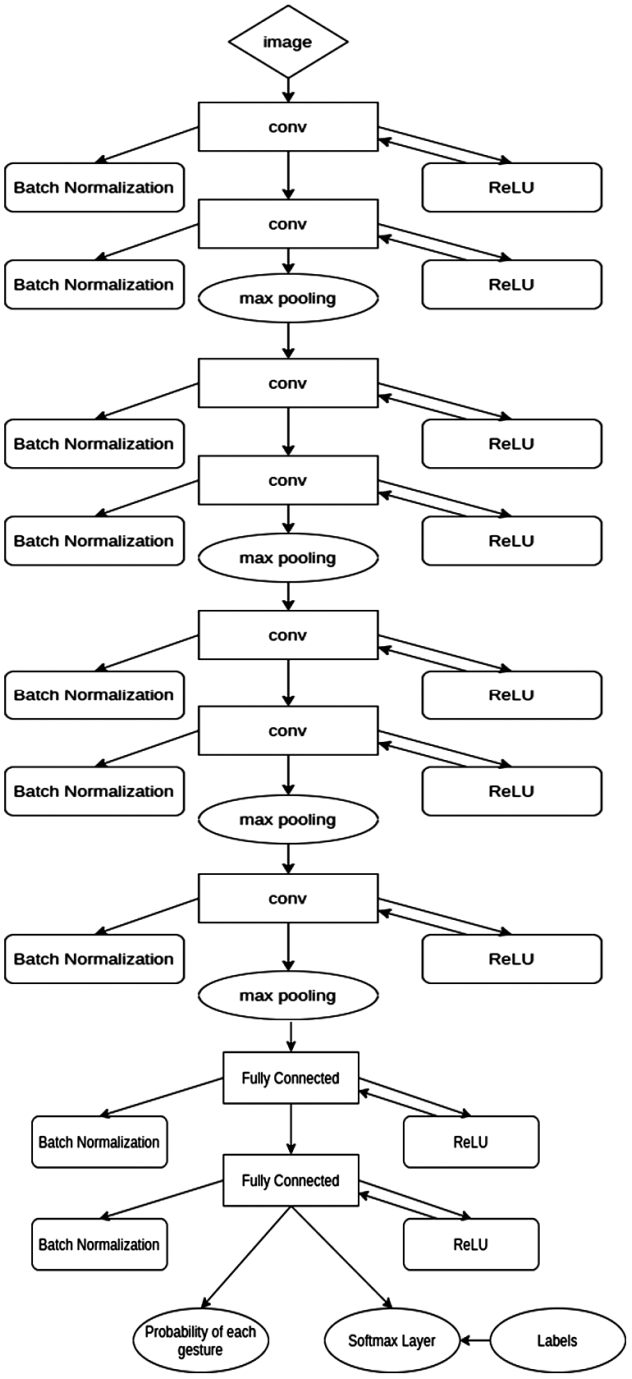

Fig. 3 — CNN & Classification model

gesture classes, and the class with the maximum probability is chosen as the output. The classes are assigned probabilities, which all add up to one. The signer's video, which contains continuous gestures performed in order, is evaluated using the RMSE value to separate individual gestures. The keyframes for each of those gestures are sent through the trained convolutional layers in order to predict their class. The classes are appended to a list, forming a sentence in order. The sequence of gesture classes in the form of sentences is passed through a window mechanism that extracts every grammar chunk from the sentence. A grammar chunk comprises three individual classes that are supposed to be in a specified order. In order to form such grammar chunks, the complete sentence is processed under the sliding window mechanism. According to this principle, the sentence is processed from the beginning until the end. The first grammar chunk from the sentence is extracted and will be processed to verify its grammatical quality. In this way, all chunks are processed by moving the window with stride value 1, as depicted in Fig. 4(a).

Implementation of Sentence Prediction

A few gestures may go unnoticed or misclassified, which may lead to inaccurate predictions. Also, the performer might not perform all the words in the sentence. If the words are not recognized and detected by the CNN model or if the performer fails to perform all gestures, the complete meaning of the sentence may not get conveyed to the listener. To tackle these issues, predicted words must be verified to ensure pleasant communication. The next word prediction can be efficiently done using the RNN-LSTM network. Long Short Term Memory has been very effective in next word prediction for a sequence and can retain the information for a very long period of time, and has a chain structure containing four neural networks and memory blocks called cells. The computations in LSTM are carried out by gates. The input, along with the previous cell state value, is passed to the forget gate, which is evaluated by the sigmoid function represented in Eq. (2).

$$f_t = \sigma(W_f\ [h_{t-1}, x_t] + \ b_t) \qquad \ldots (2)$$

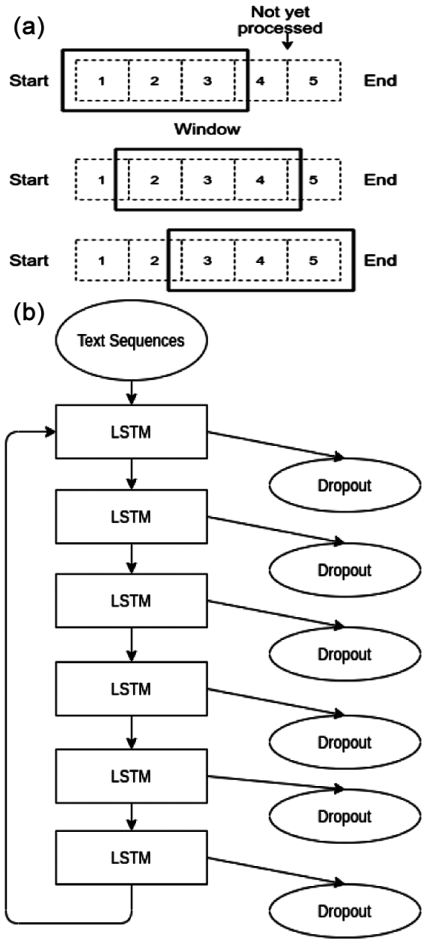

Fig. 4 — (a) Extraction of Grammar phrases from sentence for verification, (b) RNN network layer

Algorithm: Sentence Prediction

Procedure : SENTENCE PREDICTION (sentence Sequence, k, n)
Input: Sentence sequence, Window Size(k), Length_of_sequence(n)
Output: Grammar phrases for the sentence

```
1:      end Window Limit = n-k
2:      nitialize sentence Grammar as null
4:      for i = 0, 1, 2, ...(end Window Limit + 1) do
5:              Initialize individualGrammar as null
6:              for i = 0, 1, 2, ...k do
7:              individual Grammar.append (sentence Sequence [i + j])
8:                      Sentence Grammar. append (individual Grammar)
```

where, $(x) = 1/1 + e^x$, W_f is the weighted average of input x at time period t and the hidden state value at time period t-1. Sigmoid evaluation of weighted average and bias restricts the output range between 0 and 1, where the value closest to 0 explains those values that will have to be forgotten and those closest to 1 will be forwarded for every node input. The input gate multiplies the sigmoid and tanh values of the previous hidden state and input values as in Eqs. (3–5). The Tanh function regulates the network mentioned in Eq. (5), and the Sigmoid function decides those values that will have to be retained and those that can be lost.

$$i_t = \sigma(W_f\,[h_{t-1}, x_t] + b_t) \quad \ldots (3)$$

$$c = \sigma(W_c\,[h_{t-1}, x_t] + b_c) \quad \ldots (4)$$

$$\tan h(x) = \frac{\sinh(x)}{\cosh(x)} = \frac{e^{2x}-1}{e^{2x}+1} \quad \ldots (5)$$

The weighted average of the previous hidden state value and input along with bias under the sigmoid function is multiplied with the tanh function of the same weighted average along with its own bias value as the output of the cell state. The previous hidden state and input are evaluated with a sigmoid function multiplied by the tanh of the summation of outputs from the forget and input gates to produce the output of the layer. The model is trained by predicting the third word from the linguistic expressions of the previous two words using the model, as shown in Fig. 4(b). The prediction is done for sentence phrases. This prediction is very useful when a sequence after processing from the CNN model undergoes a mismatch with its grammatical representation. The phrases extracted from the detected set of gestures using the sliding window mechanism are assessed by passing them through RNN-LSTM trained models. The sentence misclassification is compromised by predicting the next-word from the two previous words. The prediction is verified by the next gesture detected by the CNN classification. The missed gesture, if any, is included in the sequence of gesture classifications according to the grammar rules. This chronological order of gesture sequences is mapped to the Indian Sign Language system to represent meaningful words. The consequent gestures, interpreted into classes verified with defined grammar, are assembled into one sentence, which is further processed to create its speech equivalent. Text-to-speech helps in reading the text out loud. This gTTS API provides a multisensory facility that allows users to read and hear the same content in unison. The sentence sequence is first converted into a speech module stored in MP3 format. The application can handle many languages, like English. The text-to-speech system is composed of two parts: the front end, which performs normalization of text by converting the numerals, abbreviations, etc., into a specially designed hash to be further processed, and the back end, which converts the linguistic text into sound. The audio can be processed at varied speed levels. The text-to-speech system aimed to convert the complete text to speech rather than translating only the linguistic expressions. A powerful neural network is trained to produce text converted to a high-fidelity voice equivalent.

Results and Discussion

Dataset

The dataset consists of recorded videos representing Indian Sign Language that denotes various relationships among human beings. They are complete words unlike[31] and have the same linguistic properties as spoken languages and are expressed either by hand movements or hand movements along with facial expressions. These languages are generally used by signers to communicate their thoughts. The gesture videos were performed by 10 different people. There are fifteen different signs performed and each was recorded 10 times by each of the 10 performers. The dataset consists of words that denote relationships like baby, family, engagement, etc. The videos are labelled according to their classes. Recordings were done using both hands. Performers wore dark attire in a light back ground. This reduces the complexity in recognizing the gestures. The performers were made to practice repeatedly before recording to ensure continuity in the video without discrepancy. Another dataset comprises gesture class sequences. These sequences represent the grammar phrases that are valid for the sign language. Fifteen classes are rearranged to form nearly 47 patterns containing three instances each. These sequences represent the sequential order in which the signs are expected to appear in grammar following the same order. The length of the phrase is fixed to three where the first two represents the training input and the third one to be the predicted next-word of the sentence.

Keyframe Extraction

Each of the input recorded videos is being processed to obtain keyframes in a novel method against uniform keyframe extraction by pulling out frames at equal distance from the video, as shown in Fig. 5(a). In this

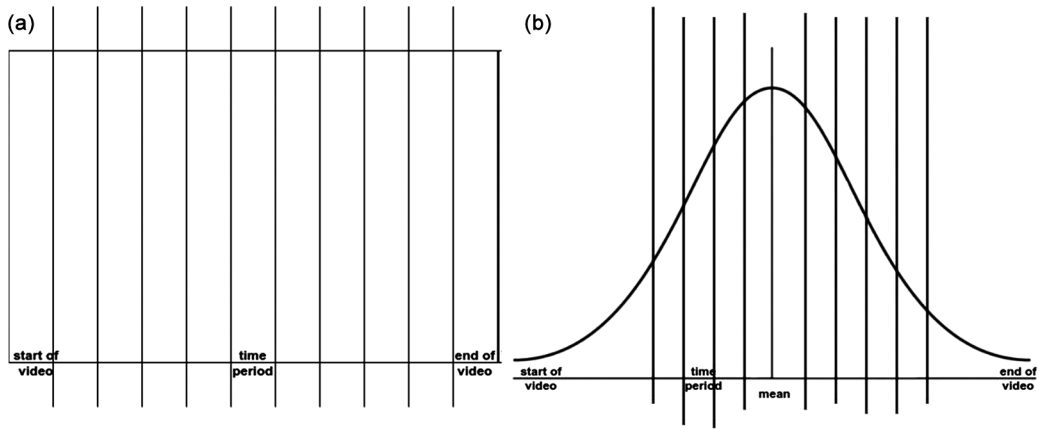

Fig. 5 — Keyframe extraction: (a) Uniform, (b) Normalized

method, the video frames are normalized with respect to the time period. The frames that are at the beginning and end of the video have low intensity in the normalized curve as they contain incomplete information regarding the gesture class, as shown in Fig. 5(b). The mean obtained is used for computing the keyframes. The frames are taken at a constant distance from one another around the mean. Four frames are taken to the left of the mean time period and five frames to the right of the meantime period, along with the mean time period frame.

These keyframes are augmented translatively to improve the dataset count. The videos are reduced to $10 \times 32 \times 32 \times 3$ size for easier computation through the Deep CNN layers. Here, 10 represent the number of frames for the video and. $32 \times 32 \times 3$ represents the size of each frame.

Gesture Recognition

Training took place for a maximum of 15 minutes and included a set of complex operations performed using the GPU. Three performers recording five videos for fifteen gestures each, making up to 225 videos, which are augmented and used as a dataset for training the CNN model, where several optimizers are utilized to have the maximum accuracy and minimum test loss. The Adam optimizer is more efficient than other as results depicted in the Table 1. The frames of the videos were split into train and test with a random state. The videos were running over the model for about 250 epochs until the change in parameters became negligible. Every epoch was evaluated within 4 seconds. For every epoch, there was an improvement in accuracy along with a reduction in loss, which demonstrates that the dataset is trained in the right manner.

Table 1 — Comparison of test accuracies and loss over various optimizers

Optimizers/Parameters	Test accuracy	Test loss
SGD	0.77847113884	1.66007373262
RMSProp	0.67831513260	0.81501329488
Adamax	0.71903276131	1.17280815172
Adam	0.89903276131	0.93798099793

Table 2 — Grammar phrases represented as classes forming patterns to predict word 3 from word1 and word 2

S. No	Word 1	Word 2	Word 3
1	1	2	3
2	1	3	5
3	1	4	7
4	1	5	9
5	1	6	11
6	1	7	13
7	1	14	15

Sentence Validation

The sentence sequences consist of three classes comprising a grammar sequence. The three classes are separated as two training inputs and one next-word for the sequence in the same order. These grammar sequences are trained over an RNN- LSTM network with soft max classification to predict classes. There are 47 records of such grammar sequences that are being formed in patterns. These sequences represent the sequential order in which the sign language words are expected to appear in grammar, as depicted in Table 2. The dataset is being split into 37 training records and 10 testing records. The training period was ~10 minutes, with around 4 seconds for each epoch. Since there are only a few records available for training, it is being split with a batch size of 1. This data was trained for about 200 epochs, and the accuracy of the training model increases over epoch.

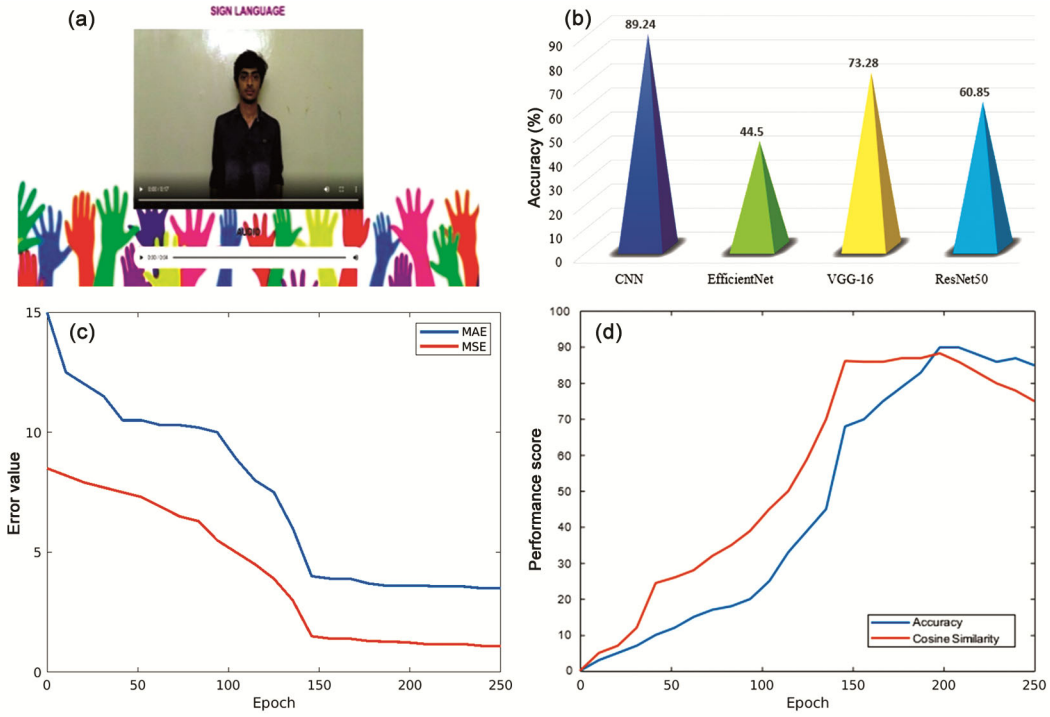

Fig. 6 — Performance analysis: (a) UID of sign language convertor, (b) Performance analysis of designed CNN model with pre-trained CNN models, (c) Error value of next-word prediction over epoch, (d) Accuracy and cosine similarity score over epoch

Table 3 — Confusion matrix sign language gesture recognition

S. No.	Labels	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	House	0.875	0.0	0.0	0.0	0.0	0.0	0.0	0.125	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	Neighbour	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	Address	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	Family	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
5	Relative	0.0	0.0	0.0	0.0	0.941	0.059	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6	Children	0.0	0.0	0.0	0.0	0.0	0.875	0.0	0.125	0.0	0.0	0.0	0.0	0.0	0.0	0.0
7	People	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
8	Person	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
9	Engagement	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.923	0.07	0.0	0.0	0.0	0.0	0.0
10	Baby	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.077	0.923	0.0	0.0	0.0	0.0	0.0
11	Marriage	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.923	0.077	0.0	0.0	0.0
12	Divorce	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
13	Enemy	0.125	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.875	0.0	0.0
14	Birth	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.091	0.0	0.0	0.0	0.0	0.091	0.818	0.0
15	Funeral	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0

The model, after reaching stability gives a test accuracy of 89.99%. The pre-recorded videos of signs are trained over Deep CNN to detect and classify the gestures. The set of sentence sequences is trained over the RNN-LSTM network to find the next-word aiding sentence formation and adhering to the grammar. An algorithm to evaluate every sentence sequence and modify the sentence in the case of wrong grammar has also been implemented. The video is converted to a list of gestures, and the list is modified to have proper grammar and is converted to its voice equivalent.

The User Interface (UI) of the proposed work with ease of interaction is shown in Fig. 6(a). The confusion matrix of 15 gestures being classified for the test data after 250 epochs is illustrated in Table 3. From this matrix, it is clear that the misclassification of data among the trained gestures is very low. Thus, a sequence of keyframes extracted from the recording after normalization is classified as having accuracy,

Table 4 — Comparative study of normalized and uniform keyframe extraction

Performance Metrics	Uniform	Normalized
Accuracy	0.797	0.892
Precision	0.823	0.894
Recall	0.797	0.807
F1 Score	0.787	0.770

precision, recall, and f1 score values of 89.2%, 89.4%, 80.7%, and 77% respectively. The metrics of the normalized keyframe extraction methodology in comparison to the uniform keyframe extraction mechanism are depicted in the Table 4. To compare the performance of the designed CNN with existing pre-trained models like EfficientNet, VGG-16, and ResNet50, they were trained and tested with normalised keyframe extraction. During the analysis, CNN outperformed the other three networks with a maximum accuracy of 89.2%, while the other pre-trained models ended up with an accuracy of 44.5%, 73.2%, and 60%, respectively and shown in the Fig. 6(b).

There is a significant increase in accuracy when detecting gestures using the novel method. Evaluation of these performance metrics for the CNN model indicates that extracting key frames closer to the mean outperforms extracting frames uniformly across the video. It is observed in Fig. 6(c) that the accuracy of the RNN-LSTM model for predicting the next word kept increasing over epochs and reached a maximum of 89.99% after 200 epochs. Training for 200 epochs provided a cut above any other option. The cosine similarity of the predicted next-word with the actual next-word also was increasing. The mean-square and mean-absolute errors of the model had a steep decrease while training for more epochs, plotted in Fig. 6(d). Dropout after every layer helped in avoiding over fitting for the model. This proved that the model was a good fit to predict the next-word and eventually complete the whole sentence with grammar.

Conclusions

The novel normalized key frames extraction from pre-recorded videos has proved advantageous by improving accuracy with a value of 84.2% using custom CNN. Also, RNN-LSTM, which was used to validate the sentence detected by CNN, ensures the reliability of the model. The voice output for the video gives us a clear picture of what is being conveyed by the hearing- and speech-impaired people. Thus, Sign Language Recognition can be accomplished by getting video samples of gestures and evaluating them against the known Sign Language. This can be put to use in real life, enhancing communication. This also helps us break the barrier between common people and hearing-impaired ones. The proposed system is not designed to recognize specific regional language signs, like Indian Sign Language signs. As the Sign Languages are prone to being regional, the application could be made to facilitate the needs of the users. At the outset, the proposed model is capable of recognizing sign gestures and predicting the sentence with an accuracy of 89.99% with the help of custom CNN-LSTM model.

Acknowledgement

This research work is supported by TIH-IoT CHANAKYA Group (PhD, PG & UG) Fellowship Program, 2021-2022, TIH Foundation for IoT and IoE, IIT Bombay (TIH- IoT), Mumbai, India. We are immensely thankful to MIT students for their help rendered to prepare the dataset of ISL relationship signs.

References

1 Kim S, Park G, Yim S, Choi S & Choi S, Gesture-recognizing hand-held interface with vibro tactile feedback for 3D interaction, *IEEE Trans Consum Electron*, **55** (2009) 1169–1177, DOI:10.1109/TCE.2009.5277972.

2 Soumya R M, Deepthi K, Goutam S & Anirban S, A feature weighting technique on SVM for human action recognition, *J Sci Ind Res*, **79(7)** (2020) 626–630, DOI:http://nopr.niscpr.res.in/handle/123456789/54986.

3 Jayanthi P, Ponsy R K B, Swetha K & Subash S A, Real time static and dynamic sign language recognition using deep learning, *J Sci Ind Res*, **81(11)** (2022) 1186–1194, DOI:https://doi.org/10.56042/jsir.v81i11.52657.

4 Jayanthi P & Sathia P R K B, Gesture recognition based on deep convolutional neural network, *Proc Int Conf Adv (IEEE)* 2018, 367–372, DOI:10.1109/ICoAC44903.2018.8939060.

5 Palak M, Pawanesh A & Parveen K L, Scene based classification of aerial images using convolution neural networks, *J Sci Ind Res*, **79(12)** (2020) 1087–1094, DOI:http:// nopr.niscpr.res.in/handle/123456789/55729.

6 Mohandes M, Deriche M & Liu J, Image-based and sensor-based approaches to Arabic sign language recognition, *IEEE Trans Hum Mach Syst*, **44** (2014) 551–557, DOI:10.1109/THMS. 2014.2318280.

7 Kritika N & Madhu S, Automated isolated digit recognition system: an approach using HMM, *J Sci Ind Res*, **70(4)** (2011) 270–272, DOI:http://nopr.niscpr.res.in/handle/123456789/11585

8 Elmezain M, Al-Hamadi A, Appenrodt J & Michaelis B, A Hidden Markov Model-based continuous gesture recognition system for hand motion trajectory, *Proc Int Conf Pattern Recognition* (Tampa, FL) 2008, 1–4, DOI:10.1109/ICPR.2008.4761080.

9 He K, Zhang X, Ren R & Sun J, Spatial pyramid pooling in deep convolutional networks for visual recognition, *Proc*

Comput Vis ECCV (Zurich) 2014, 346–361, DOI:https:// doi.org /10. 1007/978-3-319-10578-9_23

10 Kishore P V V, Prasad M V D, Prasad C R & Rahul R, 4-Camera model for sign language recognition using elliptical Fourier descriptors and ANN, *Proc Int Conf Signal Proc Commun Eng Syst* (Guntur, India) 2015, 34–38, DOI:10.13140/RG.2.1.4220.8803.

11 Starner T & Pentland A, Real-time American sign language recognition from video using Hidden markov models in motion-based recognition, *Comput Image Vis*, **12** (1997) 227–243, DOI:10.1109/ISCV.1995.477012.

12 Pankajakshan P C & Thilagavathi B, Sign language recognition system, *Proc Int Conf on Inno in Infor Embedded and Commun Syst* (Coimbatore, India) 2015, 2–5, DOI:10.1109/IC IIECS.2015.7192910

13 Dardas N H & Georganas N D, Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques, *IEEE Trans Instrum Meas*, **60** (2011) 3592–3607, DOI:10.1109/TIM.2011.2161140.

14 Adithya V, Vinod P R & Gopalakrishnan U, Artificial neural network based method for Indian sign language recognition, *Proc Int Conf Info & Commun Tech IEEE* (Thuckalay, Tamil Nadu, India) 2013, 1080–1085, DOI:10.1109/ CICT.2013.6558259.

15 Gaolin F & Wen G, A SRN/HMM system for signer-independent continuous sign language recognition, *Proc Int Conf on Automatic Face Gesture Recognition IEEE* (Washington-DC, USA) 2002, 312–317, DOI:10.1007/ 3-540-47873-6_8.

16 Haque P, Das B & Kaspy N N, Two-handed Bangla sign language recognition using principal component analysis (PCA) and KNN algorithm, *Proc Int Conf on Electr Comput Commun Eng* (Cox's Bazar, Bangladesh) 2019, 1–4, DOI:10.1109/ECACE.2019.8679185.

17 Bao P, Maqueda A I, Del-Blanco C R & Garcıa N, Tiny hand gesture recognition without localization via a deep convolutional network, *IEEE Trans Consum Electron*, **63** (2017) 251–257, DOI:10.1109/TCE.2017.014971

18 Kumar E K, Kishore P V V, Sastry A S C S, Kumar M T K & Kumar D A, Training CNNs for 3-D sign language recognition with color texture coded joint angular displacement maps, *IEEE Signal Process Lett*, **25** (2018) 645–649, DOI:10.1109/LSP.2018.2817179

19 Bantupalli K & Xie Y, American sign language recognition using deep learning and computer vision, *Proc Int Conf on Big Data IEEE* (Seattle, WA, USA) 2018, 4896–4899, DOI:10.1109/BigData.2018.8622141.

20 Islam M R, Mitu U K, Bhuiyan R A & Shin J, Hand gesture feature extraction using deep convolutional neural network for recognizing American sign language, *Proc Int Conf on Frontiers of Signal Proc* (Poitiers, France) 2018, 115–119, DOI:10.1109/ICFSP. 2018.8552044.

21 Molchanov P, Gupta S, Kim K & Kautz J, Hand gesture recognition with 3D convolutional neural networks, *Proc Conf Comput Vis. Pattern Recognit* (Boston, MA) 2015, 1–7, DOI:10.1109/CVPRW.2015.7301342.

22 Rung-Huei L & Ming O, A real-time continuous gesture recognition system for sign language, *Proc Int Conf on Automatic Face and Gesture Recognition IEEE* (Nara) 1998, 558–567, DOI:10.1109/AFGR.1998.671007.

23 Wang H, Leu M C & Oz C, American sign language recognition using multi-dimensional hidden Markov models, *J Inf Sci Eng*, **22(5)** (2006) 1109–1123.

24 Pradeep K, Himaanshu G, Partha P R & Debi P D, Coupled HMM based multi-sensor data fusion for sign language recognition, *Pattern Recognit Lett*, **86** (2017) 1–8, DOI: 10.1016/ j.patrec.2016.12.004

25 Mittal A, Kumar P, Roy P P, Balasubramanian B & Chaudhuri B B, A modified LSTM model for continuous sign language recognition using leap motion, *IEEE Sens J*, **19** (2019) 7056–7063, DOI : 10.1109/JSEN.2019.2909837.

26 Chuan C, Regina E & Guardino C, American sign language recognition using leap motion sensor, *Proc Int Conf on Mach Learn Appl* (Detroit, MI) 2014, 541–544, DOI:10.1109/ICMLA.2014.110

27 Kumar P, Roy P P & Dogra D P, Independent Bayesian classifier combination based sign language recognition using facial expression, *J Inform Sci*, **428** (2018) 30–48, DOI:10.1016/ j.ins.2017.10.046

28 Naglot D & Kulkarni M, Real time sign language recognition using the leap motion controller, *Proc Int Conf on Inventive Comput Tech* (Coimbatore, Tamilnadu) 2016, 1–5, DOI:10.1109/INVENTIVE.2016.7830097.

29 Hisham B & Hamouda A, Arabic sign language recognition using Ada-Boosting based on a leap motion controller, *Int J Inf Tecnol*, **13** (2021) 1221–1234, DOI: https://doi.org/ 10.1007/s41870-020-00518-5

30 Huang S, Mao C, Tao J & Ye Z, A novel Chinese sign language recognition method based on keyframe-centered clips, *IEEE Signal Process Lett*, **25** (2018) 442–446, DOI:10.1109/ LSP.2018.2797228

31 Zhu G, Zhang L, Shen P & Song J, Multimodal gesture recognition using 3-d convolution and convolutional LSTM, *IEEE Access*, **5** (2017) 4517–4524, DOI:10.1109/ACCESS. 2017.2684186.

32 Man G & Sun X, Interested keyframe extraction of commodity video based on adaptive clustering annotation, *Appl Sci*, **12** (2022) 1502, DOI:https://doi.org/10. 3390/app12031502.