Received: November 29, 2021. Revised: January 4, 2022. 163

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

A Reversible Convolutional Neural Network Model for Sign Language Recognition

Arun Prasath Govindan[1]* **Annapurani Kumarappan[1]**

[1] *Department of Networking and Communications, School of Computing,*
SRM Institute of Science and Technology, Chengalpattu, Kattankulathur, Tamilnadu 603 203, India
* Corresponding author's Email: ag7678@srmist.edu.in

Abstract: Hand gestures and voice inputs have been measured as the vital communication component for the past few decades. Here, deep learning-based Reversible Convolutional Neural Network (Rev-CNN) is explicitly modelled to predict gesture-based sign language. Similarly, this work concentrates on modelling a reversible model to identify the voice gesture into sign language. It shows reversible representation by attaining superior accuracy with a lesser amount of model parameters and various CNN architecture. Here, the efficiency of the reversible model is evaluated with the prevailing G-CNN, VGG-11/16 model over the testing and training environment. Here, two diverse datasets like ROBITA Indian Sign Language Gesture Database and the standard voice-input dataset, is considered for evaluation purpose. The highest prediction accuracy of 94.38 % and 97.89 % is attained using the proposed reversible CNN model over the other approaches like GCNN, VGG-11 and VGG-16 model. The experimental outcomes and metrics like loss function, error rate and execution time are measured and compared with different methods like GCNN, VGG-11/16. Additionally, other efficiency metrics are utilized to determine the efficiency of the anticipated model. The model outperforms the existing approaches by categorizing the gestures with reduced error rate. The prediction accuracy of the reversible CNN (dataset 1) is 95.38 % and for dataset 2 is 96.69%. Similarly, the execution time is 5.5 minutes.

Keywords: Sign language, Voice input, Deep learning, Reversible CNN, Gesture model.

1. Introduction

Millions of deaf and hard of hearing persons communicate with each other through sign language. For example, in America, around 32 million people with hearing loss share using American sign language [1]. On the other hand, most people have limited sign language knowledge, making it complex to interact with the deaf and hard of hearing [2]. Therefore, sign language recognition (SLR) has attracted much interest to bridge the large communication gap. On the other hand, sign language is far more complicated and unpredictable than other activities, consisting of perfect finger and irregular arm motions, making accurate recognition [3]. Many various types of sign language recognition (SLR) systems have been developed, including vision, acoustic, radio frequency (RF), and inertial measurement unit (IMU) sensor-based model. Although, most people are unable to give SLR consistently and rather identify sign language in a confined manner. Although few vision-based techniques offer consistent identification by training on whole sentences, the signals (video) are unable to capture fine-tuned finger movements and exposed to background textures and illumination noises [4]. Practically the signals employed in current SLR systems are incapable of capturing sign gestures effectively. Acoustic-based approaches, record motions are sensitive to noises, and they do not detect finger movements [5]. RF-based approaches, record arm motions only, whereas photoplethysmography (PPG)-based approaches [6] do not influence movements. Sign speaker [7], the most recent SLR technology, identifies fingerspelling and does continuous SLR with one smartwatch. However, the gyroscope and accelerometer make sign speaker capture finger

movements correctly and detect only one-handed signals when using one smartwatch [8].

The following obstacles need to be addressed to construct an SLR system that can be employed in practical circumstances. First, how can two arms consistently capture perfect finger movements as well as irregular arm motions? It's impossible to get accurate SLR without the proper signals. Second, without signal segmentation, how can continuous SLR be achieved? Instead of performing signal segmentation, it may be preferable to identify the entire text. Third, how can the scalability of SLR be improved in terms of diverse sign signal strengths? Because different people's sign signals are different strengths, the proposed SLR should be adaptable to various persons to ensure identification accuracy [9]. Finally, can we also create an SLR system can be adopted in real-life? Various prevailing solutions are bulky (vision or sensing gloves) or requires calm environment (acoustic) causes impractical outcomes [10]. As a result, a portable and effective SLR system is immediately needed to assist the hearing impaired in communicating with common people at any time and in any location [11]. Existing study proposes and develops DeepSLR, a unique end-to-end SLR system. It continually converts sign language into audio in real-time so that individuals can realize what a deaf person is saying, even if they are unfamiliar with sign language [12]. We employ two armbands with an IMU and sEMG (Electromyography) sensors to gather sign signals on both forearms, different from existing SLR systems. Arm movements are captured by the IMU sensor, consisting of a gyroscope and an accelerometer; the sEMG sensors capture fine-grained finger motions [13]. The author uses IMU signals to extract the euler angle and quaternion to describe complicated hand motions for improved SLR. However, there are common drawbacks like computational complexity and loss error [14, 15].

- An E2E prediction system performs some preliminary pre-processing steps to avoid the redundancy and noise over the input data.
- The feature vectors are clustered using conventional k-means clustering (clusters sign and voice feature vectors separately) which is followed by feature learning process.
- The feature vector classification is done with the proposed reversible CNN which is embedded with auto-encoder and decoder to extract the input without any loss or error rate.
- The simulation is done with MATLAB 2020a simulation environment where the comparison is made among the existing G-CNN, VGG-11 and 16 models where various performance metrics like accuracy, error rate, CV are compared and evaluated.

The work is organized as: In section 2, a comprehensive analysis is done with various existing approaches where the pros and cons of the anticipated model is highlighted; in section 3, the anticipated model is deliberately explained to show the models' significance. The experimented numerical outcomes are discussed and compared with existing approaches in section 4. The summary of the research is given in section 5 with the idea for future research enhancements.

2. Related work

Koller [16, 17] suggested an approach for SLR using EMG and data glove sensors. In continuous SLR, electromyography signals from hand muscles are gathered for word. Joshi, [18] presented a moment invariant sign language recognition system for Australia. The design created a database with ten images for each sign and extracted features using the moment-consistent approach. A neural network is used to classify the data. The experimental results revealed that the proposed method successfully ranks six postures for interpretation, whereas four are not identified, and it may occasionally misclassify 5–15 % of the time [19]. Moment invariants are traditionally generated in "geometric moments analysis" using information from the interior region and shape boundary. The moments utilized to create moment invariants are depicted in a consistent manner; however, it is evaluated for the practical purpose in discrete manner. On the other hand, using cosine functions indeed of sine tasks is vital for compression because some cosine functions are needed to suitable distinctive signal [20]. However, cosines express certain boundary conditions set in differential equations. The suggested methods demonstrate how numerous characteristics depicted on hand geometry in depth-based images describe finger and hand postures to predict the difficult hand postures correctly [21].

In 2013, an eight different signers video stream proposed a dynamic hand posture identification technique. Skin colour detection techniques are used to extract features from the videos [23]. Twenty various Arabic postures were studied with this method and attained a recognition ratio of 85.67 %. Although the current approach lowered the error rate from 45 % to 28 %, distinguishing between the same postures remains a challenge. Simonyan [24] offers

Microsoft Kinect approach for hand posture identification. They took advantage of Kinect's ability to capture density, depth, and 3D objects scan. They then used a Bayes classifier to classify the postures, achieving a 100 % accuracy rate, but the algorithm calculated only for five poses. Furthermore, it does not distinguish between various rotations and orientations of hands. Finally, Shrenika [25] introduced MCNN for implicit feature extraction and hand posture detection. Based on the JTD dataset and employs video camera of an NAO robot where it merges cubic kernel to increase features for classification and multi-channel information flow for detecting images. The multi-channel architecture is used to tune the Sobel operator-based filters, but it couldn't get the best characteristics out of them. Nevertheless, they scored 91 % recognition in all images, 92 % in the smaller images, and 94 % in the original images.

Gemmeke [26] discusses SVM, ANN, DT, and RF, and CNN are some of the machine learning patterns that can be employed. The assumption that multiclass operations can prevent overfitting and be considerably more accurate on large databases is undoubtedly correct. The primary goal of SVMs is to do data correlation using non-linear mapping. Rather than computing the inner products of all pairings of data in the feature space, kernel techniques function in implicit feature space and high-dimensional without computing the data co-ordinates. This procedure is frequently less computationally expensive than explicit coordinate computation. ANN design is not chosen as it fails previously to offer satisfactory outcomes; while SVM perform linear and non-linear classification by translating inputs into high-dimensional feature spaces with kernel property [27, 28].

3. Methodology

The research flow includes four essential phases: 1) data acquisition, 2) pre-processing the input voice, and 3) classification. Here, simulation is done with MATLAB 2020a environment, and metrics like accuracy, loss function, error rate, CV and execution time are evaluated to show the model significance. Fig. 1 depicts the block representation of the anticipated model.

Table 1. comparison of various approach

Categories	References	Advantages	Disadvantages
Separating speakers	Hou, [10]	Identify the voice of speakers	Adopted only under the controlled environment
		based on facial video using filtering model	
		Predict soft mask for filtering the wild nature	---
	Gemmek, [26]	Differentiate the association among the lip movement and speech	It considers only two speakers and incredibly adopted for background noise
	Chung, [27]	Identify the problematic spectrogram mask for every speaker	It is a highly complex and weaker explanation
	Leidal, [28]	Robustness and acquire speakers' information	Enormous preparation and complex network
		Single image and more robust sub-network capacity	Lesser complex towards the applications
Localized and separated objects	Joshi, [18]	Modelling visual and auditory modalities	Localized audio source
		Adopts low rank for extracting the correlated components	It does not work effectually in a wild environment
		Provide mixed and separate audio devoid of conventional supervision	Motion information is not determined for evaluation
	Koller, [17]	Produce curriculum learning and motion trajectory	Suitable for audio and video synchronization
		Predicting unlimited videos and entertainment media	Requires added sound source localization (video and audio)

Received: November 29, 2021. Revised: January 4, 2022.

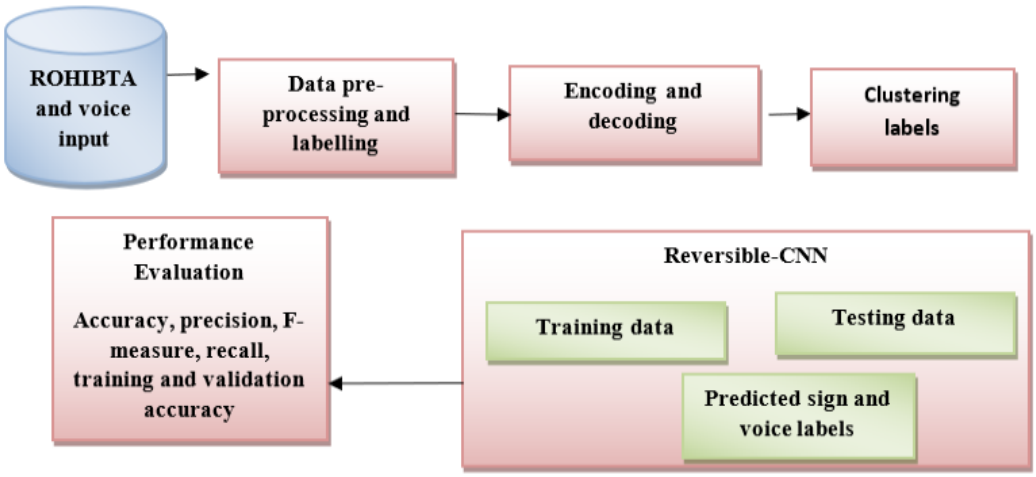

Figure. 1 Block diagram of reversible-CNN model

3.1 Dataset description (sign language)

The real-time data is collected from ROBITA Indian sign language gesture database. The dataset includes both testing and training data with labels and counts. There are three labels above_dynamic, across_dynamic, and advance_dynamic. In training set, the counts of these labels are above_dynamic = 210, across_dynamic = 112, and advance_dynamic = 34; similarly, in case of testing set, the count of these labels are above_dynamic = 34, across_dynamic = 34 and advance_dynamic = 34. The size of the dataset is smaller, which the main cause of reduced accuracy is when the number of training samples is higher. It is directly proportional to the prediction accuracy. The real-time videos are captured and transformed into image frames. The converted images (frame-by-frame) is cropped and resized for 200x200 pixels. It reduces the image quality; thereby image enhancement process is carried out to improve the visualizing nature of the image.

3.2 Dataset description (voice input)

The dataset is acquired from the real-time standards, and it is not a benchmark or standard dataset. Even though it is considered the research constraint, it is efficiently achieved for the data collected from real-time samples. It includes both pros and cons.

3.3 Pre-processing

Pre-processing is considered an essential task in gesture recognition to enhance the dataset quality. The acquired sign gesture of diverse sizes with high resolution influences the efficiency and speed. Therefore, dataset with the signing gestures is cropped for all the available images. Then, to provide the dataset in a usable format for the DL model, every image is down-sampled spatially to 256x256 size. The reduced image resolution and size reduce the computational complexity and assist in faster convergence.

3.4 Labelling

Data pre-processing is followed by a crucial part specifically for supervised learning. It is the process of dataset samples tagging with meaningful tags to offer learning bias. The collected images are categorized into various classes, and the images of various classes are provided in various folders, respectively. Therefore, data labelling is done based on the class name.

3.5 Mathematical modelling of encoding and decoding part

The auto-encoding part of NN is split into two diverse parts: encoder and decoder. It is mathematically provided as in Eqs. (1 - 3):

$$\phi = \chi \rightarrow \mathcal{F}\ (encoder) \tag{1}$$

$$\psi: \mathcal{F} \rightarrow \chi\ (decoder) \tag{2}$$

$$\phi, \psi = \arg\min_{\phi,\psi} ||X - (\psi \ \ o \ \ \phi)||^2 \tag{3}$$

The encoder part ϕ marks the provided original data χ towards the latent space $\mathcal{F}$ for dimensionality reduction. Subsequently, decoder function ψ needs to map latent and reduced output space. Here, the output is the same of input data χ where the encoder and decoder pair intends to reconstruct the data and

shape after capturing and performing certain generalized non-linear data transformation. The network's encoding part is specified with some standard NN (here CNN is considered) function

passed via bias parameter b, activation function σ and latent dimension z, and it is shown in Eq. (4):

$$z = \sigma\,(Wx + b) \tag{4}$$

It is a related way for providing the NN's decoding part, and it is represented with diverse activation functions, weight, and bias. It is expressed as in Eq. (5):

$$x' = \sigma'\,(W'z + b') \tag{5}$$

The loss function L for the provided NN is expressed using the encoding and decoding network function. It is expressed as in Eq. (6):

$$L(x, x') = \left|\left|x - x'\right|\right|^2 = \left|\left|x - \sigma'\big(W'\left(\sigma(Wx + b)\right) + b'\big)\right|\right|^2 \tag{6}$$

Based on the provided Eq. (6), the loss function L is used for training the NN via the standard backpropagation process. The objective of auto-encoder is to choose suitable encoder and decoding functions with minimal information encoded and re-generated using the decoder with a minimal loss function. This method facilitates supervised learning with the construction of cluster labels (sign and voice) using k-means clustering and the generated tags for a different purpose. The following are the step-by-step process:

1) Initially, capture the meta-data descriptive and characteristics as features and construct the feature vectors as $< f_1, f_2, \ldots, f_n)$ for all the sign data.
2) Apply the traditional k-means for feature vector clustering and predict the cluster group (sign and voice).
3) Consider the class groups and corresponding identifications (tags) as labels;
4) Fed the input data with its corresponding feature vectors and generate labels for its successive stages.

Then, construct the auto-encoder model based on NN with specific hidden neurons and layers, i.e., nodes.

1) The number of nodes over the inner layers specify the number of clusters;
2) The number of nodes over the input layer specifies the feature size and vectors;
3) The nodes over the output layer specify the probabilistic values for the provided two datasets representing the cluster labels.
4) Then, partition the constructed data to testing/training datasets.
5) Train auto-encoder based CNN with the training dataset.
6) Predict and cluster the testing dataset labels with the trained NN.

The encoding part is accountable for predicting the sign or voice data's most influencing or essential features. However, the encoder and decoder decrease the feature space, and the chosen features are used for clustering. The encoder then diminishes the total features from the most critical input data components. Subsequently, the decoder considers the diminished set of influencing features and intends to reconstruct initial values devoid of losing the information. The $< encoder\ and\ decoder >$ pair forms the mechanism for diminishing the data dimensionality for clustering the clustered data.

3.6. Reversible CNN

Here, a well-known CNN model is modelled explicitly for voice and gesture-based sign language recognition. The anticipated model is known as reversible CNN and it is composed of 4 convolutional, 3 pooling, 2 dropouts, 2 fully-connected and 1 SoftMax layers with a total of 12 layers. With the weighted layer, the filter size of 3, 2, and 1 (smaller) is used indeed of other CNN architectural model (larger). The gesture size ($256 * 256$) is fed to the convolutional layer for extracting features using sliding window. The filter weights are learned automatically for feature extraction from the input image. Here, 32 convolutional filters with reduced features $[3 * 3 * 32]$ are used. As an outcome, the higher-level features are specified by $[256 * 256 * 32]$ dimensions are extracted. The non-linear activation function is performed after convolutional layers and known as a hyperbolic tangent for learning non-linear boundaries. The anticipated R-CNN architecture model is not so comprehensive; therefore, the computation load of tanh is not influenced by efficiency. The adoption of the tanh function provides faster training process (training time). Therefore, the tanh function utilization seems to be more advantageous. The activation function tanh is provided in Eq. (7):

$$f(x) = \frac{1-\exp^{-2x}}{1+\exp^{-2x}} \tag{7}$$

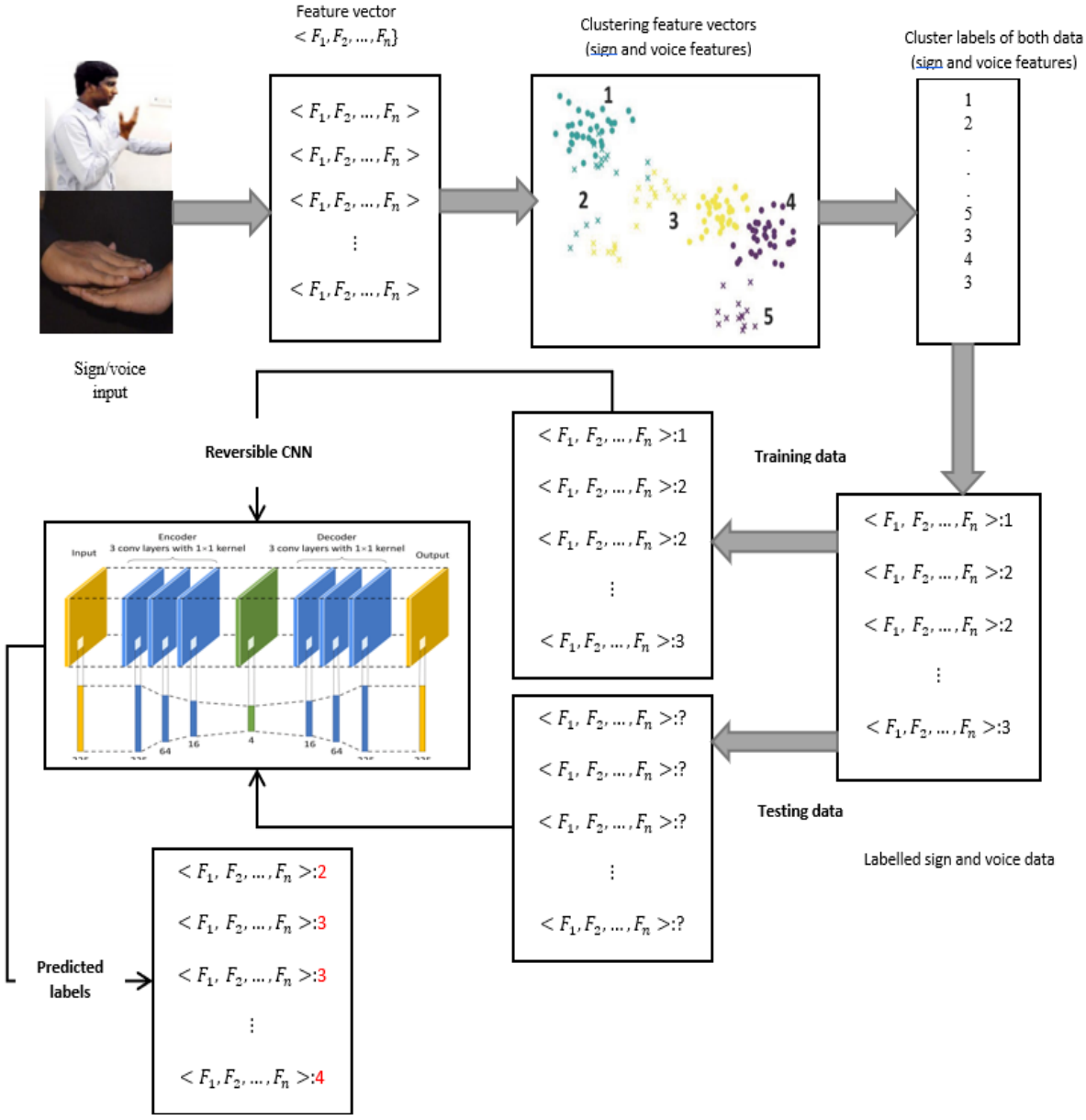

Figure. 2 Block diagram of reversible-CNN model

The size of the outcoming feature maps is scaled down by factor (2) with maximal pooling operation. Some other sets of max-pooling and convolutional layers are stacked for the generation of spatio-temporal gesture representation. Here, 4 convolutional layers with 1 stride are used, and the tanh activation functions are used. The kernel size for all convolutional layers is 3, 3, 1 and 3, with a broader depth of 32, 64, 64 and 128, placed over the model. The smaller kernel size is used to learn the smaller sign textures. During pooling operations, max-pooling is utilized to reduce the feature size with 2 filter sizes and 2 strides. Some fully connected layers are utilized to link the extracted features, and the number of hidden layers is utilized towards 2 FC layers, i.e., 84 and 512. During training process, two dropouts with the probability of inactive neuron discarding are used for eliminating the over-fitting issues. At last, the output from the final fully-connected layers are provided to the soft-max layers to identify the clustered classes with the evaluation of corresponding probability function as in Eq. (8):

$$P(y = i|x) = \frac{e^{x^T} w_i}{\sum_{k=1}^{K} e^{x^T} w_k} \quad (8)$$

Here, x^T specifies the T^{th} array element, and K sets the total element count over the array x. The

Reversible-CNN configuration is given in table 2. Similarly, the algorithm for the reversible-CNN model is shown below. Here, the model is designed explicitly for gesture prediction to handle both the sign and voice of the recognition model. The efficiency of the model is evaluated with the various existing approaches. The foremost objective of considering this reversible--CNN model is to predict the features automatically. As an outcome, the model is superior than the prevailing recognition process. The convolutional layers followed by pooling, drop out, fully connected, and SoftMax layers are provided for compact representation of the CNN model. With the less-dependent Reversible--CNN architecture, the model provides a superior recognition system with lesser training time consumption over prevailing deep learning approaches. Fig 2 depicts the overall architectural diagram.

Algorithm 1: Reversible-CNN functionality

Input: Feature extraction from convolutional layers;
Output: Reduction of negative values;

1. $int\ i = 1$;
2. Feature vector extraction from the convolutional layers;
3. **For** all $2 < i < 5$; // Reversible-CNN convolutional layres;
4. {
5. Attain feature vectors extracted from the successive layers L_{i-1};
6. Use activation function $tanh$;
7. Apply $f(x) = \frac{1-\exp^{-2x}}{1+\exp^{-2x}}$ as in Eq. (7);
8. Use feature vector refined using activation function to the successive Reversible-CNN;
9. Extracting features with successive convolutional layers C_{i+1};
10. }
11. **End process**

4. Experimental analysis

Here, two diverse architectures of the standard CNN model and tested for sign and gesture recognition. The experimental analysis is done where the comparison with other approaches is discussed in section 3. The expected model is executed on the system using the MATLAB 2020a simulator. Various metrics like classification accuracy, processing time, loss, and accuracy are considered for the performance evaluation.

4.1. Accuracy

The accuracy is a quality index to evaluate the classifier efficacy and it is properly depicted as the predicted sample ratio to the total provided/input samples. It is mathematically expressed as in Eq. (9):

$$Accuracy = \frac{TN+TP}{FN+FP+TN+TP} \quad (9)$$

Here, $TN, TP, FN, and\ FP$ are true negative, true positive, false negative and false positive, respectively. The accuracy with the provided dataset 1 using existing and proposed approaches like GCNN, VGG-11/16 and reversible CNN model is depicted in table 3. The accuracy attained with reversible CNN model is 95.38 % and 96.69 %, respectively. The accuracy attained by GCNN is 94.38 % and 97.89 % for dataset 1 and dataset 2. Similarly, the accuracy achieved by VGG-11 model

Table 2. Reversible CNN description

Layers	Filters	Feature mapping size	Kernel size	Stride
Input	--	256x256	--	--
Conv 1	32	256x256x32	3x3	1x1
Max-pool 1	1	128x128x32	2x2	2x2
Conv 2	64	128x128x64	3x3	1x1
Conv 3	64	128x128x64	1x1	1x1
Max-pool 2	1	64x64x64	2x2	2x2
Conv 4	128	64x64x128	3x3	1x1
Max-pool 3	1	32x32x128	2x2	2x2
Dropout 1	--	--	--	--
FC 1	--	512x1	--	--
FC 2	--	84x1	--	--
Dropout 2	--	--	--	--
Output	--	43x1	--	--

Table 3. Original and augmented outcomes of datasets 1 and 2

Approaches	Dataset 1		Dataset 2	
	Original outcomes	Augmented outcomes	Original outcomes	Augmented outcomes
Reversible CNN	95.38%	97.34%	96.69%	97.58%
CNN [7]	94.38%	96.34%	97.89%	96.23%
VGG-11 [16]	93.06%	94.02%	96.78%	94.73%
VGG-16 [16]	93.5%	94.86%	96%	94.24%

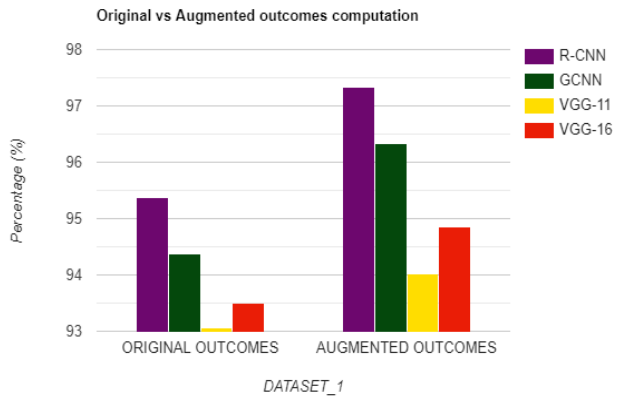

Figure. 3 original and augmented outcomes of dataset 1

is 93.06 % and 96.78 % for both datasets. The prediction accuracy of the VGG-18 models for both datasets is 93.5 % and 96 %. The outcomes reveal the superiority achieved with reversible CNN over GCNN, VGG-11 and VGG-16 (see Fig 3). The existing model like GCNN, VGG-11/16 and its performance are compared over the provided dataset. It is performed to attain the generalization ability of the training model. Generally, augmentation is done for the generation of newer samples by converting the collected initially dataset. Here, four diverse samples of signers are produced above, across and advanced. The classification outcomes of the provided dataset are shown in table 3. The predominant outcomes on the provided datasets are more convincing to foresee the generalization capability of the training model.

Table 4. Comparison of execution time and parameter considered

Approaches	Execution time (mins)	Parameters considered
Reversible CNN	5.5	65, 052, 548
CNN [11]	9.20	67, 250, 845
VGG-11 [16]	40.50	160, 298, 368
VGG-16 [16]	45.33	164, 791,580

4.2. Loss function

Here, the categorical loss function (cross-entropy) is used to evaluate the loss identified during multiple sign language gestures classification. It is mathematically expressed as in Eq. (10):

$$Loss = \sum_{i=1}^{n} O_i \log \hat{O}_i \tag{10}$$

Here, $\hat{O}_i$ is the output model value, O_i specifies the targeted value, and $'n'$ specifies the number of scalar values over the output model. The loss value observed for all these four models for both datasets is evaluated to highlight the training accuracy. The evaluated loss for all the four diverse approaches of CNN variants constantly drops with the iteratively increasing time, and over the successive iterations, it reaches the fixed values. For the provided dataset, the loss of the reversible CNN model is 0.2897, GCNN is 0.3565, VGG-11 is 0.465, and VGG-18 is 0.450, respectively. For the following dataset, the loss function of reversible CNN is dropped to 0.012, GCNN is 0.0135, VGG-11 is 0.0615, and VGG-16 drops to 0.178. Reversible CNN converges faster than the prevailing GCNN, VGG-11 and VGG-16, respectively. Table 4 depicts the execution time of the reversible CNN over other approaches.

4.3. Prediction result

Another performance metric known as the confusion matrix is also evaluated as it summarizes the appropriately and non-appropriately predicted samples of every class. Therefore, the recognition accuracy of these classes is extracted. Fig. 4 shows the broader way of analysing the prediction accuracy for every class of given dataset attained by all these variants of the CNN model. It is observed that the anticipated CNN model provides promising outcomes for all the available courses.

4.4. Other prediction parameters

The computational time is the crucial parameter for hand gesture or sign language prediction for some real-time applications. Table 4 depicts the training time taken by all these CNN variants. The parameter details considered by these models are provided in this table for determining the model complexity. The total amount of trainable parameters is evaluated using some expressions. The parameters for every convolutional layer are computed with Eq. (11):

$$P_{convolution} = \left(\left(width_{filter} * height_{filter} * no.of\ the\ previous\ layer\ filters + 1\right) * no.of\ filters\right) \quad (11)$$

Here, total parameters considered for the fully connected layers are computed with Eq. (12):

$$P_{fc} = \left(\left(previous\ layer\ (p) * current\ layer\ (c)\right) + 1 * c\right) \quad (12)$$

It is proven that the predictions with the four diverse variants of the CNN model take lesser computational time and some fewer parameters than the prevailing variants of the CNN model.

4.5. Cross-validation (CV)

Generally, k-fold CV is used for stabilizing the model performance. Here, a 10-fold CV is considered to measure the performance of the complete data range, and it is adopted over the reversible CNN model.

Table 5. Overall prediction accuracy comparison

Total gestures	Total signers	Dataset	Prediction accuracy (%)
16 signs	18	NA	63.87%
16 signs	18	NA	63.93%
10 signs	72	Single	90%
26 signs	90	Single	93.5%
23 signs	10	Single	91.4%
18 signs	10	NA	91.2%
24 signs	7	Multiple	90.2%
26 signs	12	Single	80.67%
Category 1	50	Single	93.38%
Category 2	50	Single	94.69%
Real-time	3	Multiple	95.38
Real-time	3	Multiple	96.69

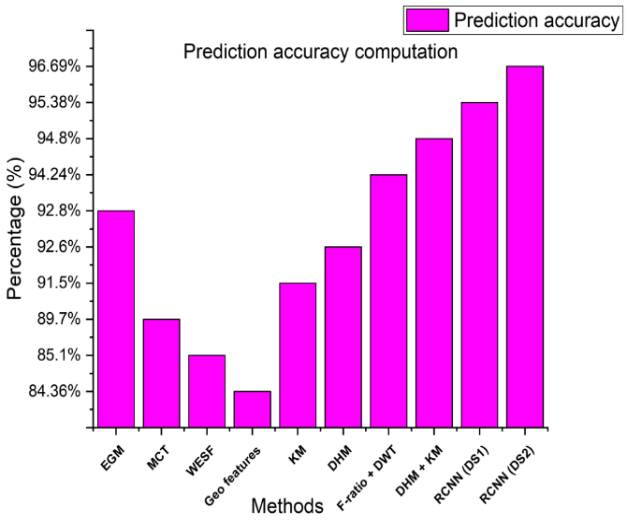

Figure. 4 Comparison of various prediction accuracy

Table 6. Comparison of reversible CNN with existing approach

Method	Prediction accuracy (%)
EGM [12]	92.8%
MCT [16]	89.7%
WEST [29]	85.1%
Geometric features [18]	84.36%
KM [6]	91.5%
DHM [3]	92.6%
F-ratio + DWT [1]	94.24%
DHM + KM [2]	94.8%
Reversible CNN Dataset1	95.38%
Reversible CNN Dataset2	96.69%

4.6 Comparison of prediction outcomes

Here, the performance outcomes are evaluated with the various prevailing approaches of similar classifier problems of sign-to-voice and voice-to-sign language prediction. The broader analysis of this evaluation is provided in table 5 and table 6. The assessment is done to attain the prediction accuracy, and it is observed as a widely adopted performance metric of all the prevailing approaches. Table 6 shows the evaluation of the reversible CNN model is done with the provided real-time and voice dataset. From this table, it is proven that the existing models have simulated with the constraint number of signs and attained better prediction accuracy 64.87 %, 64.93 %, 91 %, 94.35 %, 92.4 %, 91 %, 91.2 %, 81.67 %, 95.38 % and 96.69 %, respectively. The classification accuracy attained by GCNN is 94.38 % and 97.89 %, respectively (see Fig. 4). It is proven that the prediction with the reversible CNN model exceeds all these prevailing approaches as it acquires the superior prediction accuracy of 94.38 %

and 97.89 % for the provided real-time ROBITA Indian sign language gesture database (converted to image frames) and real-time voice standards, respectively. Table 6 compares the prevailing proposed work with the various overall result for the online publicly available dataset. The prediction of these reversible CNN models seems to be robust for all these datasets. The efficiency of the model completely relies on the model design in a reversible manner. Generally, all the existing works concentrate only on either encoding or decoding part. However, this work intends to predict both the sign-to-voice and voice –to-sign language prediction. There are only limited studies that attempts to concentrate on both. Also, the efficiency is achieved with reduced execution time. The computational time is lesser which is directly proportional to reduced computational complexity.

5. Conclusion

This research concentrates on modelling efficient sign language and voice prediction for gesture-based and vision-based recognition models. Here, a novel deep learning-based reversible CNN model with proper representation. Additionally, three diverse variants of CNN like GCNN, VGG11 and VGG16 are also evaluated and modified to predict sign language and voice input. The anticipated vision-based approach avoids the user dependency; thus, it is suitable for practical adoption. The research contribution is its competency to predict the difficult sign and voice input for the provided standard dataset with superior prediction outcomes over the prevailing approaches. The reversible model performance is tested under various gestures and voice input with the ROBITA Indian sign language gesture database (converted to image frames) and real-time voice standards. With the complete experimental evaluation, it is evident that the three diverse categories (above, advance, and across) of signs are used in this work. The anticipated reversible CNN model attains superior classification accuracy of 94.38 % and 97.89 %. Along with the prediction accuracy, some other efficiency metrics are used to establish the model efficiency. It is experimented with the augmented data and known as an invariant towards various transformations. It is proven to be entirely robust towards the classification process with the voice input and gestures with lesser error rate. However, the anticipated architectural deep learning approaches are further optimized for hand and voice input in the future, and a detailed comparison is made. They provided architectural model is explored to reduce the error rate over the real-time sign language recognition. In the future, the work is extended with the adoption of meta-heuristic optimization approach to attain global solution.

Conflicts of interest

"The authors declare no conflict of interest".

Author contributions

"Conceptualization, Arun Prasath Govindan and Annapurani kumarappan; methodology, software, validation, formal analysis, investigation, resources, data curation, writing—original draft preparation Arun Prasath Govindan; review and editing, Arun Prasath Govindan and Annapurani kumarappan".

References

[1] R. Naoum, H. H. Owaied, and S. Joudeh, "Development of a new Arabic sign language recognition using the k-nearest neighbour algorithm", *Journal of Emerging Trends in Computing and Information Sciences.*, Vol. 3, No. 8, 2012.

[2] P. V. V. K. Mieee, M. V. D. Prasad, C. R. Prasad, and R. Rahul, "4-Camera model for sign language recognition using elliptical fourier descriptors and ANN", In: *Proc. of International Conf. On Signal Processing and Communication Engineering Systems.*, pp. 34-38, 2015.

[3] N. Baranwal, N. Singh, and G. C. Nandi, ''Indian Sign Language Gesture Recognition Using Discrete Wavelet Packet Transform", In: *Proc. of 2014 International Conf. on Signal Propagation and Computer Technology.*, p. 573577, 2014.

[4] S. K. Gharghan, R. Nordin, M. Ismail, and J. A. Ali, "Accurate Wireless Sensor Localization Technique based on Hybrid PSO-ANN Algorithm for Indoor and Outdoor Track Cycling", *IEEE Sensors Journal.*, Vol. 16, No. 2, pp. 529–541, 2016.

[5] J. Zhang, W. Zhou, and H. Li, "A new system for Chinese sign language recognition", In: *Proc. of IEEE China Summit and International Conf. on Signal and Information Processing.*, pp. 534-538, 2015.

[6] O. Koller, J. Forster, and H. Ney, "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers", *Computer Vision and Image Understanding.*, Vol. 141, pp. 108–125, 2015.

[7] R. Cui, H. Liu, and C. Zhang, "Recurrent convolutional neural networks for continuous sign language recognition by staged optimization", In: *Proc. of IEEE CVPR.*, pp. 7361–7369, 2017.

[8] J. Huang, W. Zhou, Q. Zhang, H. Li, and W. Li, "Video-based sign language recognition without temporal segmentation", In: *Proc. of Thirty-Second AAAI Conference on Artificial Intelligence, arXiv:1801.10111.*, 2018.

[9] Q. Pu, S. Gupta, S. Gollakota, and S. Patel, "Whole-home gesture recognition using wireless signals", In: *Proc. of ACM MobiCom.*, pp. 27–38, 2013.

[10] J. Hou, X. Y. Li, P. Zhu, Z. Wang, Y. Wang, J. Qian, and P. Yang, "Signspeaker: A real-time, high-precision smartwatch-based sign language translator", In: *Proc. of ACM MobiCom.*, 2019.

[11] P. Asadzadeh, L. Kulik, and E. Tanin, "Gesture Recognition Using RIFD Technology", *Personal and Ubiquitous Computing*, Vol. 16, No. 3, pp. 225–234, 2012.

[12] J. Wu, Z. Tian, L. Sun, L. Estevez, and R. Jafari, "Real-time american sign language recognition using wrist-worn motion and surface emg sensors", In: *Proc. of IEEE BSN*, 2015.

[13] K. Ren, Q. Wang, C. Wang, Z. Qin, and X. Lin, "The security of autonomous driving: Threats, defenses, and future directions", In: *Proc. of the IEEE*, Vol. 8, No. 2, pp. 357–372, 2020.

[14] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting", *The Journal of Machine Learning Research*, Vol. 15, No. 1, pp. 1929–1958, 2014.

[15] N. C. Camgoz, S. Hadfield, O. Koller, H. Ney, and R. Bowden, "Neural sign language translation", In: *Proc. of Conf. on Computer Vision and Pattern Recognition*, pp ,7784–7793, 2018.

[16] O. Koller, J. Forster, and H. Ney, "Continuous sign language recognition: Towards large vocabulary statistical recognition systems handling multiple signers", *Computer Vision and Image Understanding*, pp.108–125, 2015.

[17] O. Koller, H. Ney, and R. Bowden, "Deep learning of mouth shapes for sign language", In: *Proc. of the IEEE International Conf. on Computer Vision Workshops*, pp. 85–91, 2015.

[18] G. Joshi, R. Vig, and S. Singh, "CFS-InfoGain based Combined Shape-based Feature Vector for Signer Independent ISL Database", In: *Proc. of International Conf. on Pattern Recognition Applications and Methods*, pp. 541-548, 2017.

[19] J. L. Raheja, A. Mishra, and A. Chaudhary, "Indian sign language recognition using SVM", *Pattern Recognition and Image Analysis*, Vol. 26, No. 2, pp. 434–441, 2016.

[20] B. Xie, X. He, and Y. Li, "RGB-D Static gesture recognition based on convolutional neural network", In: *Proc. of the Second 2018 Asian Conference on Artificial Intelligence Technology.*, pp. 1515–1520, 2018.

[21] http://www.who.int/mediacentre/, World health organization (WHO), Deafness and hearing loss Key Facts, 2015.

[22] S. Stoll, N. C. Camgoz, S. Hadfield, and R. Bowden, "Text2Sign: Towards Sign Language Production Using Neural Machine Translation and Generative Adversarial Networks", *International Journal of Computer Vison*, pp. 891–908, 2020.

[23] A. Kulshreshth, K. Pfeil, and J. L. Viola Jr, "Enhancing the gaming experience using 3D spatial user interface technologies", *IEEE Computer Graphics and Applications.*, Vol. 37(3), pp. 16–23, 2017.

[24] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition", *arXiv preprint arXiv:1409.1556*, 2014.

[25] S. Shrenika and M. M. Bala, "Sign Language Recognition Using Template Matching Technique", In: *Proc. of International Conf. on Computer Science Engineering and Applications.*, pp. 1–5, 2020.

[26] J. F. Gemmeke, D. P. W. Ellis, D. Freedman, A. Jansen, W. Lawrence, R. C. Moore, M. Plakal, and M. Ritter, "Audio set: An ontology and human-labelled dataset for audio events", In: *Proc. of IEEE International Conf. on Acoustics, Speech and Signal Processing, New Orleans, USA.*, pp. 776−780, 2017.

[27] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition", *arXiv preprint arXiv:1806.05622.*, 2018.

[28] C. Sanderson and B. C. Lovell, "Multi-region probabilistic histograms for robust and scalable identity inference", In: *Proc. of the 3rd International Conference on Advances in Biometrics, Alghero, Italy*, pp. 199−208, 2009.

[29] B. Li, X. Liu, K. Dinesh, Z. Duan, and G. Sharma, "Creating a multitrack classical music performance dataset for multimodal music analysis: Challenges, insights, and applications", *IEEE Transactions on Multimedia*, Vol. 21, No. 2, pp. 522–535, 2018.

[30] K. Leidal, D. Horwath, and J. Glass, "Learning modality-invariant representations for speech

and images", In: *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop, IEEE, Okinawa, Japan.*, pp. 424−429, 2017.