



# A hybrid feature-extracted deep CNN with reduced parameters substitutes an End-to-End CNN for the recognition of spoken Bengali digits

Bachchu Paul<sup>1</sup> · Santanu Phadikar<sup>2</sup>

Received: 3 August 2022 / Revised: 16 February 2023 / Accepted: 21 April 2023 /

Published online: 9 May 2023

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2023

## Abstract

Speech Recognition (SR) is an emerging field in the native language nowadays. Recognizing isolated words in the local language helps people use smartphones and electronic gadgets without technical or educational knowledge. This paper proposes a novel deep Convolutional Neural Network (CNN) architecture to classify ten spoken Bengali numerals. The proposed model generates almost similar prediction accuracy as compared to an end-to-end CNN with nine times fewer parameters has been trained. Here, the raw audio samples are pre-processed, and then a unique hybrid feature of Mel Frequency Cepstral Coefficients (MFCC), Spectral Sub-band Energy (SSE), and Log Spectral Sub-band Energy (LSSE) have been extracted frame-wise and engendered into a vector. Finally, these vectors are fed to the proposed architecture of a one-dimensional CNN and achieve the highest test accuracy of 98.52%. The model has been trained for our created speech corpus of 14000 spoken Bengali digits and 30000 spoken English digits from the audio-MNIST dataset. The proposed neural model generates high prediction accuracy with a few times fewer parameters to be trained, generating low computational costs. The outcome of the proposed model is compared with several pre-trained deep learning models; the result shows the model's superiority. Source Code: <https://github.com/BachchuPaul/Bengali-Isolated-Spoken-Digit>.

**Keywords** Hybrid feature · Mel frequency cepstral coefficients · Sub-band energy · Isolated word · One-dimensional convolution neural network · Hyper-parameter tuning · Max pooling

---

✉ Bachchu Paul  
ableb.paul@gmail.com  
Santanu Phadikar  
sphadikar@yahoo.com

<sup>1</sup> Department of Computer Science, Vidyasagar University, Midnapore, West Bengal 721102, India

<sup>2</sup> Department of Computer Science & Engineering, Maulana Abul Kalam Azad University of Technology, West Bengal, BF-142, Sector-I, Salt Lake, Kolkata 700064, India

## 1 Introduction

People are interested in communicating with smart devices and current electronic gadgets through voice. It saves time and requires less technical knowledge to use these devices. Speech recognition in native languages gains popularity among researchers as there is a scope to develop and improve the models to obtain high recognition accuracy. Bengali is a language, and almost one-sixth of the world speaks Bengali [24]. Voiced-based interaction is the easiest way to pass commands to smart devices like mobile, ticket vending machines, and many other electronic gadgets. But most of these devices operate in English, Spanish, or Chinese like languages. Still, in the world, most people only know a single language. So, these people have faced problems with using these smart devices. In a country, many languages use for communication purposes. Several deep learning-based speech recognizer models are available in English. However, the complexity of these models is large enough to run on low-resource devices. So, we aim to develop a speech recognition model in the native language of isolated spoken Bengali words. The model requires less computational cost compared to existing models. For this purpose, 14000 Bengali spoken numerals have been recorded and used for the experiment. Table 1 shows the list of Bengali numerals and their transcript.

Isolated words of spoken digit recognition have several applications like word-to-text, voiced-based call forwarding, card-less ATM use, lift without pressing floor number, command interpretation to smart devices, use of a smart wheelchair, etc. To justify the robustness of a model, the dataset should be large enough to recognize an unknown pattern. For this purpose, the dataset should be unbiased to gender, data recording devices, people from a different geographical regions, and other related factors [22, 23]. The Deep Learning models are the generalized model which handles these biased factors of a large dataset. One of the challenges of Deep Learning (DL) models is the number of trainable parameters. The model complexity and size grow according to the number of parameters. That's why deep learning models need GPU and huge memory (RAM) for computation. To overcome this, a feature extractor model always takes less memory and fewer trainable parameters, but the performance degrades. The proper features, architecture, and hyper-parameter tuning play a vital role in getting satisfactory prediction accuracy. The advantage of these feature extractor models is that they can run on the CPU with less memory.

**Table 1** List of Bengali digits and their corresponding transcript in Bengali and English

| Bengali Digits | English Equivalent | Bengali Transcript | English Transcript |
|----------------|--------------------|--------------------|--------------------|
| ০              | 0                  | শূন্য (Sunno)      | Zero               |
| ১              | 1                  | এক (Ek)            | One                |
| ২              | 2                  | দুই (Dui)          | Two                |
| ৩              | 3                  | তিন (Tin)          | Three              |
| ৪              | 4                  | চার (Char)         | Four               |
| ৫              | 5                  | পাঁচ (Panch)       | Five               |
| ৬              | 6                  | ছয় (Chhoy)        | Six                |
| ৭              | 7                  | সাত (Saat)         | Seven              |
| ৮              | 8                  | আট (Aat)           | Eight              |
| ৯              | 9                  | নয় (Noy)          | Nine               |

Though CNN is such a DL model that performs outstanding for image classification, it is being used for SR nowadays. In end-to-end models of SR, the one-dimensional CNN has been used [1, 27]. In this type of CNN, the input shape is large, and at the feature extractor level, only the input data size is reduced before feeding to the fully connected layer. The filters and the kernels are all one-dimensional vectors. On the other hand, in the feature-extracted CNN, the input shape is small, and at the feature-extractor level of CNN, the input shape is more condensed in size, and the fully connected network needs fewer neurons in each layer to train.

The highlights of the proposed work are-

1. A unique one-dimensional CNN model has been proposed to classify the isolated words that generate almost similar prediction accuracy compared to an end-to-end model with fewer parameters that need to be trained.
2. The outcome of the proposed technique has been compared with some popular pre-trained DL models to validate the technique's robustness.
3. A significant contribution to the proposed experiment, we have created a speech corpus of 14000 audio clips containing ten spoken Bengali numerals taken from 28 persons with 1400 clips in each class.
4. A different combination of frame-wise features (20 MFCC, 5 SSE, and 5 LSSE) have been extracted by a standard mathematical formula and stored in a vector of length 1860 (30x62).
5. The proposed model is irrelevant to the language and the frequency of the samples. The model's complexity (time and space cost) does not grow on different frequency levels and languages.

This paper is structured as Section 2 discusses the literature survey; Section 3 illustrates the proposed model's methodology and architecture. The result and analysis are explained in Section 4, and the conclusion and future work are discussed in Section 5.

## 2 Literature review

Researchers have used several approaches to isolated word recognition in the past decade based on various features and classifiers (based on DNN). As our proposed work is based on CNN, we have also figured out the CNN-based research works for this literature. Following that, strategy research work related to speech features is briefly reviewed first in 2.1, classification techniques are discussed in Section 2.2, and finally, the CNN-based literature is discussed in Section 2.3.

### 2.1 Based on features

Speech features are highly effective for recognizing isolated words. Lots of features are taken by researchers and used to find the significant feature vectors for proper identification. In 2015, LPC, ZCR, energy, and MFCC were used [6] to identify the isolated words effectively. Among different sets, the combination of MFCC, LPC, and neural network gives an average of 86.66% recognition accuracy in this literature. In 2016, MFCC was used here [2] on their own created dataset for identifying the Bengali speech properly. The network is trained using a back propagation neural network learning technique, and after

testing three different setups, a recognition accuracy of 98.46 percent has been reached. In 2016, MFCC, Teager-Energy Cepstral Coefficients (TECC), and Teager-based Mel-Frequency Cepstral Coefficients (TEMFCC) were used [8] for detecting the whispered speech on the WhiSpe speech database by using the Deep De-noising Auto-encoder (DDAE). This speech's cepstral feature vectors are transformed into clean cepstral feature vectors of neutral speech using the Deep De-noising Auto-encoder (DDAE). At last, with the help of TECC features, 92.81% accuracy is achieved on this whispered speech. In 2017, along with MFCC, features such as LPC, PLP, and RASTA were used [12] to identify the speakers and speech with 97.05% accuracy using a deep neural network. Lisa et al. [15] proposed an MFCC and HMM-based Bengali isolated word recognition and achieved 89.47% accuracy. Sumon et al. [34] suggested three different CNN-based Bengali speech command recognition systems. MFCC was used to extract the audio files. For a handful of short Bangla verbal instructions, they must create a data set on their own and obtain an accuracy of 74.01%. In 2020, A Bengali voice recognition system was proposed by Sharmin et al. [29]. Several parameters, such as gender and dialects, are taken into account. CNN was utilized as a classifier to classify the spoken digits using 1230 data fed on this model with 98.37% accuracy.

## 2.2 Based on classifiers

According to classification, some pieces of literature are also focused on here. Research on enhancing the accuracy of isolated word recognitions continues with several machine and deep learning approaches. In 2012, DNN was used here [32] to boost the recognition accuracy of phonetic attributes and phonemes. DNN and shallow multi-layer perceptron (MLP) are compared within the Automatic Speech Attribute Transcription (ASAT) framework. Several DNN architectures will be presented and evaluated, ranging from five to seven hidden layers and up to 2048 hidden units per hidden layer. Similarly, on phoneme classification, 86.6% recognition accuracy is achieved by using DNN. In 2014, HMM toolkit called HTK was used [26] for developing an isolated digit recognition system on the self-recorded corpus, CUAVE dataset. Finally, it achieved 95% recognition accuracy on this English language-based dataset. In 2016, DNN was used [5] to compare different speech-language recognition approaches. Among three different approaches, it is concluded that bottleneck features by following the i-vector perform better than the other two approaches. DNNs trained with additional languages enhance performance on par with single systems in fusing systems. Finally, a strategy for adapting DNNs trained with additional languages with fewer data is proposed here. In 2018, DNN based approach [11] was proposed and tested against the GMM-based system that utilizes several hidden layers. The overfitting issue is successfully solved for DNN on a large dataset for training purposes before its worse performance becomes. DNN implementation with MFCC and GFCC approach performs well in the Punjabi speech recognition system. The hybrid classifiers GMM-HMM and DNN-HMM are utilized to increase performance on both connected and continuous Punjabi speech corpora. In 2018, layer-wise relevance propagation (LRP) [3] was used to explore the interpretability of neural networks in audio. LRP is also used for identifying the relevant features for two neural network architectures to process either waveform or spectrogram representations of the data. In 2018, isolated Bengali numerals [10] were recognized in both noiseless and noisy conditions in speaker-independent mode. Support Vector Machines, Multi-Layer Perceptron, and Random Forest are used as a classifier, and the performance of these used classifiers are compared in this system. The features are extracted using MFCC, and the feature vector is built using Principal Component Analysis

as a feature summarizer. Finally, an average of 89.51% accuracy for Multi-Layered Perceptron, 91.67% for SVM, and 84.06 for Random Forest are achieved here. In 2019, NF Classifier, KNN, and SVM classifiers were used [13] to recognize the speech using the Tonal Frequency Cepstral Coefficients (TFCC). A mathematical model for a cochlear frequency map is proposed here. A hybrid classifier using a neural network and fuzzification is developed. Among different algorithms, the average accuracy of 99.06% is achieved here. In 2019, Using Multiscale Scattering of Audio Signals, LSTM is deployed here [16] to identify speech despite superfluous background noise and external disturbances. They have focused on validating the efficiency of wavelet scattering technique and LSTM networks for uttered digit recognition. In 2019, various deep learning algorithms [37], such as LSTM, BLSTM, DLSTM, DBLSTM, and DBLSTM-DNN, were used to recognize speech in the Persian language. The combination of deep belief network (DBN) is used for extracting features of speech signals, and finally, with the help of DBLSTM-DNN, 83.3% standard accuracy is achieved here. In 2020, LSTM, SVM, and ANN were used [36] to interpret fuzzy frame analysis schemes for the word-level noisy system. In MFCC, the framing step has been tweaked to reduce the number of noisy frames by employing thresholding at two levels of local maxima techniques. After investigating, it clearly shows that using LSTM, the recognition accuracy of 98 to 99% is achieved here. In 2021, a novel stochastic deep resilient network (SDRN) [30] was used to recognize speech effectively. DNN is used as a classifier for predicting the input speech signals. With the help of the neural-based opposition whale optimization algorithm (NOWOA), the hidden layers and neurons are optimized to reduce computation time. Using this approach, the standard accuracy of 99.1% of real-time accuracy of 98.1% is achieved for isolated words and 98.7% for continuous words, respectively.

### 2.3 Based on CNN

Traditional speech recognition methods have a shallow learning structure and drawbacks. DNNs have made significant progress in speech recognition in recent years but may cause overfitting and poor generalization in low-resource scenarios. Convolutional Neural Networks (CNNs) is a type of neural network that seeks to address some of these problems.

In 2014, CNNs were used [1] for speech recognition in this study, where the structure of the CNN directly accommodates specific sorts of speech variability. Their hybrid CNN-HMM method delegated temporal variability to the HMM while convolving along the frequency axis created invariance to minor frequency shifts, which are common in real-world speech signals due to speaker variances.

In 2014, the scalability of an ASR strategy was studied [20] based on CNN, which takes the raw speech signal as input to a considerable vocabulary challenge in this study. Their experiments on the Wall Street Journal corpus revealed that the CNN-based system outperforms the ANN-based system, which uses standard cepstral characteristics as input. In 2018, a CNN-based model was proposed [18] whose key aspect of weight connectivity, local connection, and polling result is that the system has been thoroughly trained, resulting in superior testing results. Experiments on wideband speech signals show that the system's performance is much improved compared to the previous technique. In 2020, a one-dimensional CNN was proposed [27] using a multilayer perceptron to extract and classify learned features. On a specified dataset of 119 speakers speaking Kurdish digits (0–9), the proposed models are tested and got average accuracy of 98.5% on speaker-dependent

and 97.3% for speaker-independent. The discussion of the related work is summarized in Table 2, stating similarities and differences among them.

The literature shows that ML-based classifiers [2, 10, 11, 13, 26] do not have a satisfactory level of accuracy. Moreover, these models need clean data, and performance degrades when the size of the dataset grows. On the other hand, the DL-based models [1, 5, 11, 16, 18, 20, 27, 30, 32, 37] didn't focus on the number of parameters to be trained. Though some literature claims a high prediction on DL-based isolated word recognition in the native language Bengali [2, 10, 15, 29, 34], the dataset size is insufficient to feed it into CNN because the DL models are a high chance to memorize the data. So, the robustness of the approach raised a question. In the proposed technique, 14000 isolated Bengali words of spoken digits have been used, which is large enough to feed into CNN with less chance to memorize the data. Also, we have focused on minimizing the number of trainable parameters to get satisfactory accuracy. A standard speech corpus audio-MNIST [3] (<https://www.kaggle.com/datasets/sripaadsrinivasan/audio-mnist>) has also been used in the experiment to justify the proposed technique.

### 3 Methodology

The phases of the proposed method are shown in schematic diagram 1. Classifying the spoken Bengali digit using a one-dimensional CNN has five significant stages shown in Fig. 1. In the first step, the individual audio sample from different speech corpus is read for pre-processing. For this experiment, two speech corpora have been used. The first speech corpus contains 14000 spoken Bengali digits, and the second contains audio clips of English spoken numerals taken from the standard dataset audio-MNIST [3] (<https://www.kaggle.com/datasets/sripaadsrinivasan/audio-mnist>) [35]. The details of each speech corpus are explained in Section 3.1. In the second phase, as each clip is of a different duration, the audio signals are split into one second to extract equal dimensions of features from each clip. In this preprocessing stage, the clips are broken down into one second by adding zero amplitude signals or by removing the silence portion by finding the average zero crossing and energy. In the next step, each sound clip has extracted the frame-wise 20 MFCC, 5 SSE, and 5 LSSE features. Then the extracted feature is transformed into a vector of length 1860. Finally, the 1860 feature is fed to the proposed CNN model architecture for training given in Section 3.4. To justify the superiority of the proposed technique, we have provided only 20 MFCC features to the proposed model. It is observed that the best accuracy obtained by feeding the frequency (20 MFCC) and the time domain features (5 SSE, 5 LSSE) is similar to an end-to-end CNN with nine [9] and twenty-seven [25] times fewer parameters trained. The model captures and predicts a real-time audio clip and is shown at the end. While designing the proposed methodology, a unique combination of features has been fused to enhance the classification rate. The MFCC distinguishes the words, whereas the SSE and LSSE determine the vocal characteristics of inter-speaker variation. The second focus has been addressed on designing the neural network. Here, a unique architecture has been proposed by CNN to reduce the computational cost compared to several pre-trained models.

The overall process of the whole work is given in Algorithm 1. The individual speech signal from the corpus is taken as input audio. Then the audio is pre-processed for the conversion of a uniform audio length of one second. This is done by taking threshold zero-crossing and energy. The MFCC, SSE, and LSSE are extracted frame-wise in the next step.

**Table 2** Summary of the following related works

| Author & Year                | Title   | Proposed Work  | Classifier                             | Result | Conclusion  | Limitation   |
|------------------------------|---|--|--|--------|---|--|
| Siniscalchi et al. 2013 [32] | Exploiting deep neural networks for detection-based speech recognition        | DNN was used here to boost up the recognition accuracy of phonetic attributes and phonemes.                            | DNN                                    | 88.3%  | DNNs and shallow MLPs were compared in this paper using the ASAT framework to categorize phonetic characteristics and phonemes. Many DNN designs were presented and analyzed with five to seven hidden layers and up to 2048 hidden units per hidden layer.                             | The absence of temporal overlapping (i.e., asynchrony) characteristics in the attributes across different dimensions was a glaring drawback of the existing detection-based speech recognition system. |
| Pawar et al. 2014 [26]       | Isolated English Language Digit Recognition Using Hidden Markov Model Toolkit | A speech recognition system was developed for recognizing the isolated digits in the English language by using the HTK | HMM                                    | 95%    | The results of the experiments and tests indicated that if the language model was created for a small selection of vocabulary and trained using a substantial amount of user speech data, a better degree of accuracy might be attained.  | However, the work was not entirely definitive because it only considered isolated digit speech data. Only a limited speech corpus was relevant.  |
| Gamit et al. 2015 [6]        | ISOLATED WORDS RECOGNITION USING MFCC, LPC AND NEURAL NETWORK                 | In this work, it was investigated how ANN was used to recognize the isolated word based on the features of MFCC & LPC  | Back Propagation Neural Network (BPNN) | 86.66% | According to the experimental findings, utilizing the suggested MFCC and combining both MFCC and LPC feature extraction techniques yields better results than using MFCC alone.   | Recognition accuracy was comparatively low and was trained for a large number of epochs.   |
| Ahammad et al. 2016 [2]      | Connected Bangla Speech Recognition using Artificial Neural Network           | A Bengali digit recognition system was developed based on the MFCC features by using the neural network                | Back Propagation Neural Network (BPNN) | 98.46% | It was concluded that connected Bengali digits are recognized using the BPNN and MFCC features. Optimizing the network based on network parameters, including the number of hidden layers, learning rate, error threshold, and epochs, was one of the main aims of this study endeavor. | The output may alter significantly with minor network architecture changes. The results also show that due to phonetic differences between the digits, recognition accuracy differs between them.      |

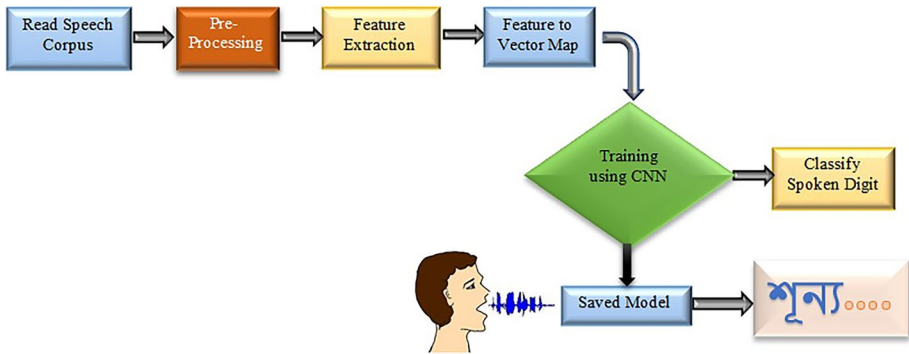
Table 2 (continued)

| Author & Year          | Title  | Proposed Work  | Classifier                | Result   | Conclusion  | Limitation   |
|------------------------|--|--|---------------------------|--|---|--|
| Ferrer et al. 2016 [5] | Study of Senone-Based Deep Neural Network Approaches for Spoken Language Recognition | Various methods for applying deep neural networks (DNNs) trained to forecast senone posteriors to the problem of spoken language recognition are compared (SLR). | DNN, GMM                  | DNN performs 40 to 70% better than GMM/iv system | It was decided that the DNN/iv technique, which substitutes DNN-based posteriors for GMM-based posteriors during i-vector extraction, was initially used for speaker recognition and achieved remarkable improvements in telephone voice.   | This method seemed extremely sensitive to data mismatch, as it was seen for the speaker recognition task.                        |
| Kaur et al. 2017 [12]  | Speaker and Speech Recognition using Deep Neural Network work                        | Isolated Words were recognized using the Deep neural network based on the MFCC features.   | Deep Neural Network (DNN) | 97.05%   | With the help of the MFCC feature extraction technique, the suggested recognition system was able to recognize speakers and speech more accurately. In particular, it was discovered that feature extraction and optimization were crucial yet challenging processes in any pattern recognition system. | Only five classes had been considered.   |
| Becker et al. 2019 [3] | INTERPRETING AND EXPLAINING DEEP NEURAL NETWORKS FOR CLASSIFICATION OF AUDIO SIGNALS | Layer-wise relevance propagation (LRP) technique was suggested for exploring the interpretability of neural networks in audio                                    | AlexNet                   | 95.82%   | It was suggested that the interpretability method that works best for neural networks for audio categorization is layer-wise relevance propagation.   | LRP demonstrated that the networks' judgments for classifications based on raw waveforms depend on a minimal subset of the data. |



**Table 2** (continued)

| Author & Year             | Title  | Proposed Work   | Classifier                 | Result  | Conclusion   | Limitation  |
|---------------------------|--|---|----------------------------|---|--|---|
| Sharmin et al. 2020 [29]  | Bengali Spoken Digit Classification: A Deep Learning Approach Using Convolutional Neural Network | Bengali spoken digits were collected from different users along with different ages and finally classified by using the CNN   | CNN                        | 98.37%  | In this research work, a novel method was provided for categorising Bengali spoken digits using CNN. The proposed model successfully categorised spoken digits from ten different speakers across diverse age groups, genders, and accents with an accuracy of 98.37%, demonstrating the validity of the suggested strategy. | Data Size was inadequate to train using deep learning model. Always a high chance to memorize the data. |
| Vani H Y et al. 2020 [36] | Fuzzy Variable Frame analysis for Speech Recognition   | This study suggested a thresholding and local maxima strategy at the framing level for the recognition process, along with a fuzzy interpretation frame analysis method utilising LSTM classifier for noisy speech at the word level. | ANN, SVM with kernel, LSTM | 99%   | It was concluded that in fixed frame analysis, every frame must be processed. According to the proposed approach, fuzzy interpretation, local maxima, and thresholding principles result in a 20% reduction in frame processing.   | Differences in accuracy were especially noticeable for frames of fixed and varying lengths.             |
| Shukla et al. 2021 [30]   | A novel stochastic deep resilient network for effective speech recognition                       | A unique stochastic deep resilient network (SDRN) for speech recognition was proposed in this paper. To forecast the input speech signal, a deep neural network (DNN) for classification was used here.                               | NOWOA                      | 99.6% for isolated words & 98.7% for continuous speech signal | It was finalised that the obtained results showed that NOWOA selects the ideal set of characteristics with excellent classification accuracy.  | 375 input features were quite expensive resulting the model complexity grows.                           |



**Fig. 1** Schematic diagram of the proposed model

From the feature matrix, the feature from each frame is engendered into a vector in the next phase. The feature vector is split into the train, validation, and test set. The model is prepared with the train and validation set. Finally, the prediction result obtained from the model for the test set is analyzed for different evaluation metrics. The overall working flow of the whole proposed work is depicted by the flowchart given in Fig. 2.

### 3.1 Dataset creation

For the proposed experiment, we have created a speech corpus of 14000 audio signals of ten Bengali spoken numerals (given in Table 1) uttered by 28 people, among them 18 males and ten females from different geographical locations of the state West Bengal (A state in India). Each word is spoken 50 times by every speaker. We have recorded using the Easy Voice Recorder Apps ([https://play.google.com/store/apps/details?id=com.coffeebeanventures.easyvoicerecorder&hl=en\\_IN&gl=US](https://play.google.com/store/apps/details?id=com.coffeebeanventures.easyvoicerecorder&hl=en_IN&gl=US)) through android smartphones. The sampling frequency of 16 KHz and mono channel with 32-bit floating representation has been fixed. The recording files are of .wav extension. The duration of the clips is within the range of 0.6 to 2.8 seconds. The data are recorded in a regular room environment with the

#### Algorithm 1 (Audio sample $x$ )

---

```

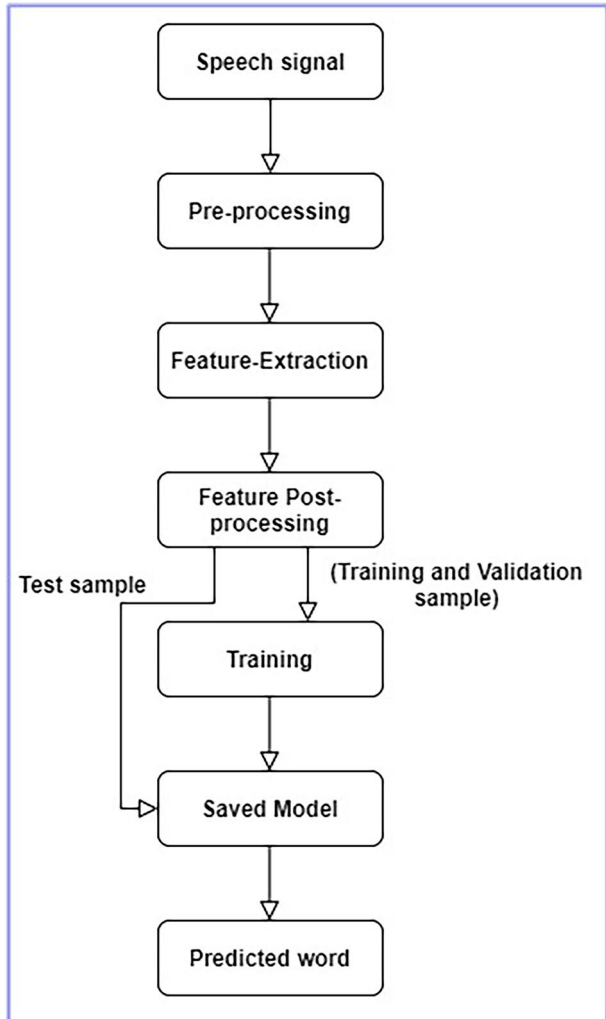
Step 1:  $x_1 \leftarrow$  Pre-process  $x$  by adjusting the length to make it into one sec
           //Apply threshold zero-crossing and energy)
Step 2:  $f[m][n] \leftarrow$  extract MFCC, SSE and LSSE in frame-wise from  $x_1$ 
           //  $m$  and  $n$  denote the number of frames (62 here) and number of features (30 here), respectively
Step 3:  $F[K] \leftarrow$  Convert  $f[m][n]$  matrix into a vector
           //Concatenate each row of  $f[m][n]$ , such that  $K = m \times n$ 
Step 4: Split  $F$  into  $F_1, F_2$  and  $F_3$  in 70:15:15 ratio
           // Split the feature into the train, validation, and test set
Step 5: model  $\leftarrow$  Train ( $F_1, F_2$ ) using proposed CNN model
           // the proposed one-dimensional CNN model given in section 3.4
Step 6:  $Y \leftarrow$  predict model ( $F_3$ )
           // Predict test set,  $F_3$  from the model

Step 7: Return ( $Y$ )
Step 8: End

```

---

**Fig. 2** Working Flowchart of the proposed method



fan/AC off. As the experiment has been done in Deep Learning (DL) model, we haven't discarded the clips containing some minor additive noise signal within the recorded clips. But large noise clips have been removed during the data cleaning.

The second, a standard popular speech corpus audio-MNIST [3] (<https://www.kaggle.com/datasets/sripaadsrinivasan/audio-mnist>) [31, 35], has 30000 audio samples of spoken digits (0-9) of 60 different speakers in English pronunciation with 3000 samples in each class. The audios were recorded at 48 kHz with the mono channel in .wav format.

### 3.2 Preprocessing

After studying the signal, it has been observed that the clips' duration is of different lengths. So it is unable to feed it into CNN. In this phase, the audio clips are split into one second by adjusting the size. By analyzing the signal, it has been noticed the actual voice zone of the

uttered words is less than one second. So, the signal with less than one second is stretched to one second by adding zero amplitude. The signal with more than one second is broken down into one second by two steps. First, the voice zone is found by finding the average energy and zero crossing using Eq. 1 [22, 25] and 2 [22, 24, 25], respectively, and removing the silence zone from raw clips. Next, zero amplitude is added at both ends of the voiced site to make the length one second. The objective of making speech signal equalization is to extract the equal dimension of the feature from each clip.

$$P_i(k) = \frac{1}{N} |S_i(k)|^2 \tag{1}$$

where,  $S_i(k)$  is a single frame  $i$ , and  $P_i(k)$  is the corresponding power, and  $N$  is the frame length.

$$ZCR = \frac{1}{2N} \sum_{j=i-N+1}^i | \operatorname{sgn}(x(j) - \operatorname{sgn}(x(j - 1))) | \cdot w(i - j)$$

Where,

$$\operatorname{sgn}(x(j)) = \begin{cases} 1, & \text{if } x(j) \geq 0 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$\operatorname{sgn}(x(j))$  and  $\operatorname{sgn}(x(j - 1))$  are the sign of the  $j^{\text{th}}$  and  $(j-1)^{\text{th}}$  data sample of a frame, respectively and  $w(\cdot)$  is the hamming window.

### 3.3 Feature extraction

One frequency-domain and two time-domain features have been extracted for the proposed experiment. A very popular frequency domain speech feature, namely MFCC of 20 dimensions, has been used for this study. Figure 3 shows the phases of MFCC calculation.

To calculate MFCC, the first phase is framing [21, 23], where each audio sample is split into 32ms duration with 50% overlapping, indicating 62 frames/sec. The next phase is Windowing [21, 25], where each frame is multiplied by a hamming window with the frame size. In the next step, to find the frequency domain components of the signal, the Fourier transform is applied by Discrete Fourier Transform (DFT) [14]. The next step of MFCC calculation is the Mel Scale Filter Bank [23, 25], where the spectrum is mapped on the Mel Scale using 26 numbers of triangular overlapping filter banks [27–29]. The final step is the Discrete Cosine Transform, where the Mel frequency spectrum is converted back into the

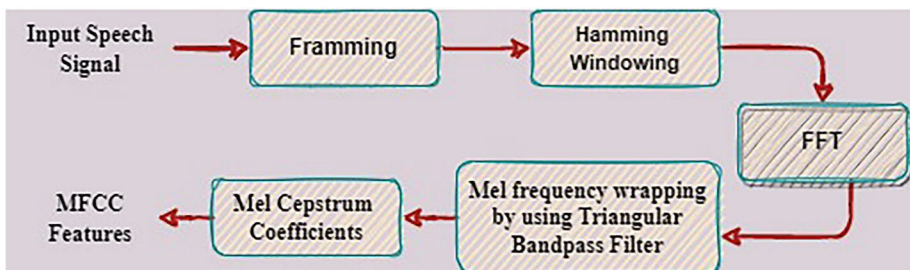


Fig. 3 Steps for MFCC feature extraction

time domain using DCT of the logarithmic Mel power spectrum [14, 25]. The first 20 coefficients of this Mel power spectrum are taken as our MFCC feature for a single frame. This sequence of steps is presented in Fig. 3.

### 3.3.1 Spectral Subband Energy (SSE)

The general idea behind the multistream approach is to divide the entire frequency band, which is described as crucial bands, into a fixed number of subbands. This can be done through the following steps.

1. A bandpass filter splits the input signal into several non-overlapping (frames) frequency bands.
2. Each frame is multiplied by Hamming Window and converted into the frequency domain.
3. The spectrum is further processed using a triangular mel-scale filter bank.
4. From the filter bank output, the subband energy is obtained.

The Power Spectral Density (PSD) of  $i^{\text{th}}$  frame  $x_i(n)$  is measured from the DFT function using Eq. 3 [19].

$$P_i(k) = \frac{1}{N} \left| \sum_{n=0}^{K-1} x_i(n)w(n)e^{-j2\pi kn/K} \right|^2 \quad (3)$$

Where  $K$  is the DFT length,  $N$  is the frame length, and  $w(n)$  is the window function. Using a bank of  $B$  triangular-shaped critical band filters placed equally on the mel scale, the frame's Spectral Subband Energy (SSE) coefficients are estimated from the PSD using Eq. 4 [4].

$$X_i(b) = \sum_k h_b(b)P_i(k), \quad 0 \leq b \leq B - 1, \quad (4)$$

Where,  $h_b$  refers to the  $b^{\text{th}}$  filterbank.

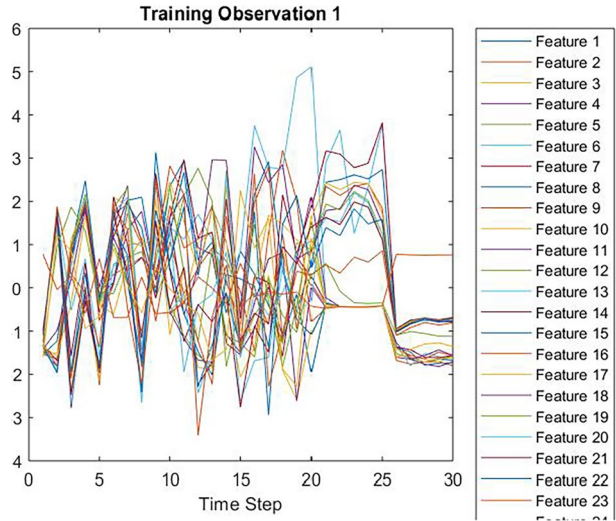
### 3.3.2 Log Spectral Subband Energy (LSSE)

Log Spectral Subband Energies are local in both time and frequency and can be computed using formula 5 [4, 19].

$$LSSE_i(b) = \log \sum_k h_b(b)P_i(k), \quad 0 \leq b \leq B - 1, \quad (5)$$

Thus from a single frame, 30 features are extracted. Since we have transformed all clips into 1 sec, now for each audio sample, there will be 62 frames of 32ms frame length with 50% overlapping. Then from each clip, a vector of length  $30 \times 62 = 1860$  is formed. The data values are normalized by subtracting its mean and then dividing by the standard deviation. A sample training observation from Bengali and audio-MNIST datasets is shown in Figs. 4 and 5, respectively. In Fig. 4 shows the feature matrix of the tenth audio sample from the training set of the Bengali dataset. Each colored line indicates one feature (For example, Feature 1 implies MFCC-1). Similarly, Fig. 5

**Fig. 4** Training observation of the 10<sup>th</sup> audio sample in the Bengali dataset



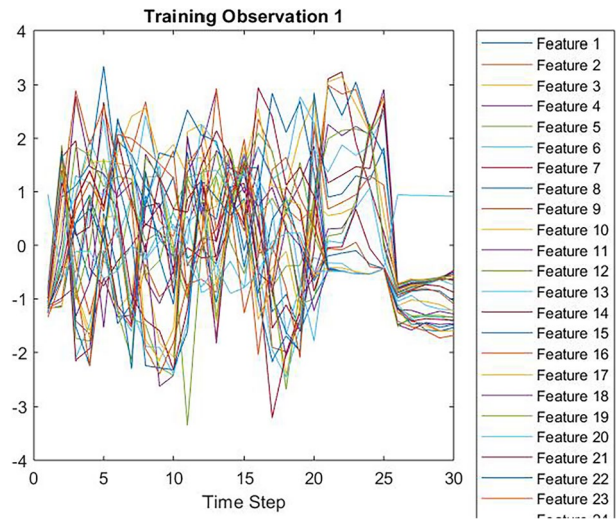
represents the feature matrix of the fifteenth audio sample from the training set of the audio-MNIST dataset.

### 3.4 Layer architecture for Training using CNN

The neural architecture of the proposed one-dimensional CNN model is depicted in Fig. 6, and the block architecture of the same is shown in Fig. 7, with different numbers of filters and kernel sizes at every layer.

In Fig. 6, the input is a one-dimensional vector of length 1860, fed to the four convolution and pooling layers, called the feature extractor block. It then generates 1152 flattened output

**Fig. 5** Training observation of the 15<sup>th</sup> audio sample in the audio-MNIST dataset



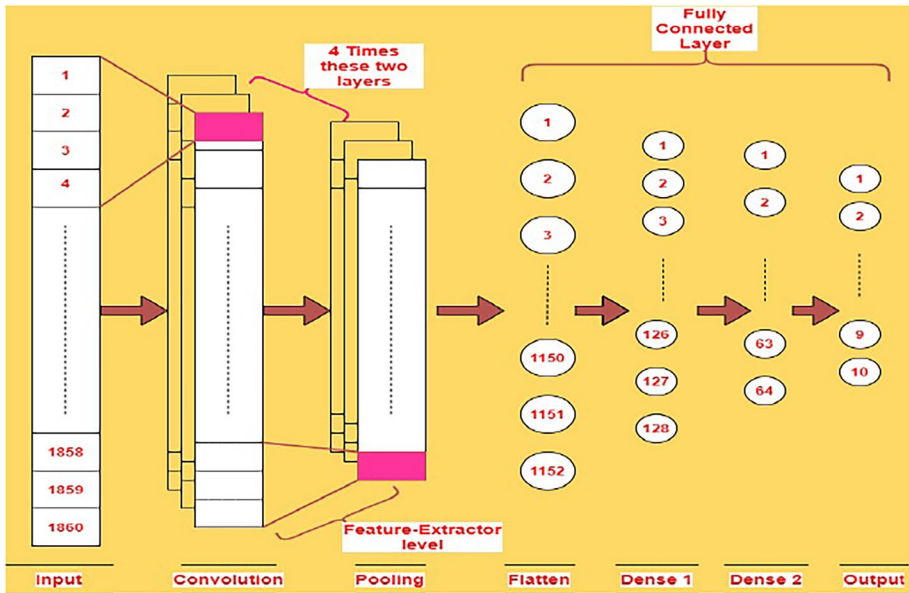


Fig. 6 Neural architecture of the proposed 1D CNN

provided to three levels of dense layers, called a fully connected layer. The outcome represents the ten output classes for ten words.

In Fig. 7, the feature extractor level of the proposed CNN model, four consecutive layers of convolution and max-pooling have been designed for training purposes with a different number of the filter kernel. In the first Convolution layer (1860,1), input is fed. The last max pooling layer generates 1152 flattened output and is then fed to a fully connected network of two dense layers of 128 and 64 neurons, respectively, followed by the ten output classes. The different parameters and hyper-parameters for the proposed architecture are given in Table 3. The output shape and number of parameters in every layer of the proposed model are shown in Table 3.

### 4 Result and discussion

The network model is individually trained for the two datasets of Bengali and English. Each dataset is split into 70% training, 15% validation, and 15% test sets for the experiment. The model is trained for 50 epochs with 32 minibatch normalizations using adam optimizer.

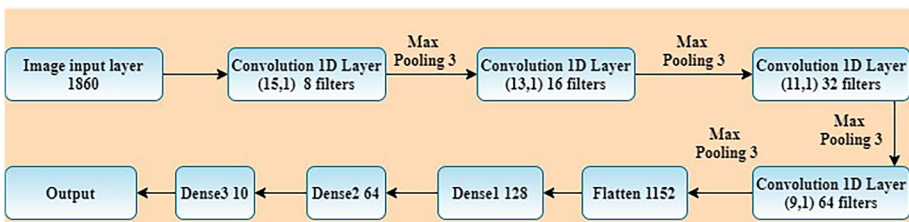


Fig. 7 Block architecture of each layer of the proposed model

**Table 3** Layer-wise input-output shape and #parameter for proposed CNN model

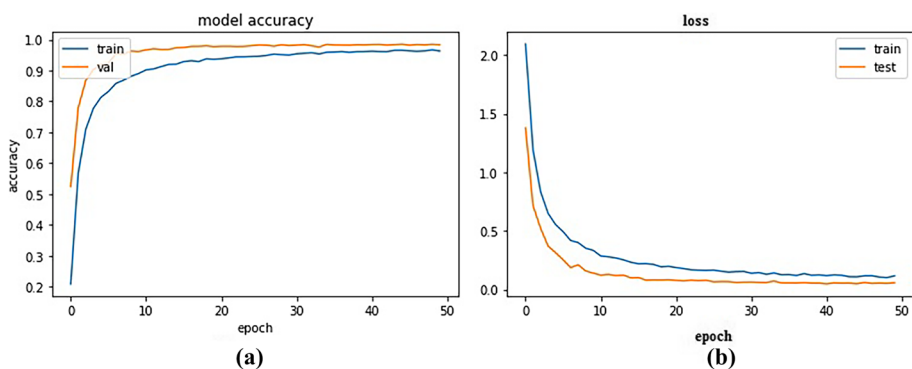
| Layer (type)                    | Output Shape      | Param # |
|---------------------------------|-------------------|---------|
| input_1 (InputLayer)            | [(None, 1860, 1)] | 0       |
| conv1d (Conv1D)                 | (None, 1846, 8)   | 128     |
| max_pooling1d (MaxPooling 1D)   | (None, 615, 8)    | 0       |
| dropout (Dropout)               | (None, 615, 8)    | 0       |
| conv1d_1 (Conv1D)               | (None, 603, 16)   | 1680    |
| max_pooling1d_1 (MaxPooling 1D) | (None, 201, 16)   | 0       |
| dropout_1 (Dropout)             | (None, 201, 16)   | 0       |
| conv1d_2 (Conv1D)               | (None, 191, 32)   | 5664    |
| max_pooling1d_2 (MaxPooling 1D) | (None, 63, 32)    | 0       |
| dropout_2 (Dropout)             | (None, 63, 32)    | 0       |
| conv1d_3 (Conv1D)               | (None, 55, 64)    | 18496   |
| max_pooling1d_3 (MaxPooling 1D) | (None, 18, 64)    | 0       |
| dropout_3 (Dropout)             | (None, 18, 64)    | 0       |
| flatten (Flatten)               | (None, 1152)      | 0       |
| dense (Dense)                   | (None, 128)       | 147584  |
| dropout_4 (Dropout)             | (None, 128)       | 0       |
| dense_1 (Dense)                 | (None, 64)        | 8256    |
| dropout_5 (Dropout)             | (None, 64)        | 0       |
| dense_2 (Dense)                 | (None, 10)        | 650     |

Total params: 182,458

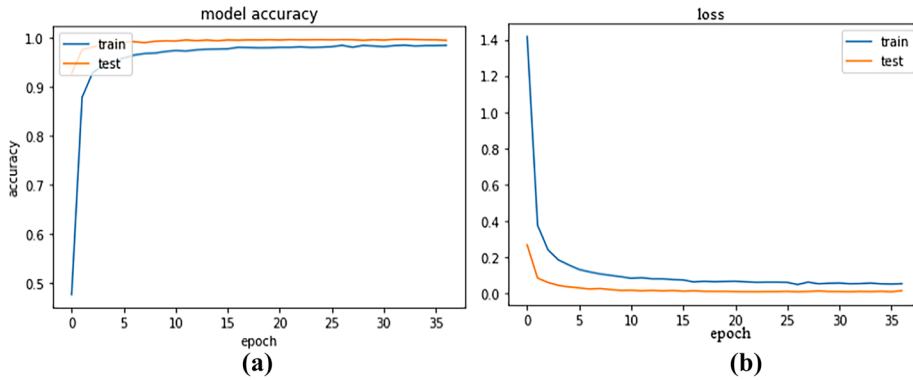
Trainable params: 182,458

Non-trainable params: 0

The training progress and the corresponding loss function are given in Figs. 8 and 9 for the Bengali and the audio-MNIST datasets, respectively. The x and y axis in Fig. 8a represents the epoch and the training and validation accuracy progress during the training of the Bengali dataset. Figure 8b represents the corresponding epoch-wise loss. Similarly, Fig. 9a represents the epoch-wise training and validation accuracy of the audio-MNIST dataset, and Fig. 9b represents the corresponding loss. The train, validation, and test accuracy for the Bengali dataset are 96.68%, 97.8%, and 98.52%, respectively; similarly, 98.47%, 99.84%,

**Fig. 8** (a) Epoch-wise training progress and (b) epoch-wise corresponding loss for the Bengali dataset





**Fig. 9** (a) Epoch-wise training progress and (b) epoch-wise corresponding loss for the audio-MNIST dataset

and 99.52% for the audio-MNIST dataset, respectively. Though each dataset is trained for 50 epochs, the audio-MNIST reaches the global minima, which means it attains the minimum loss after 37 epochs.

When the Bengali dataset is fed to an end-to-end CNN [1, 7, 27, 32], the parameters used in different layers of model architecture are given in Table 4. Before feeding, the audio signals are resampled into 8000, which is half the original recorded frequency. The

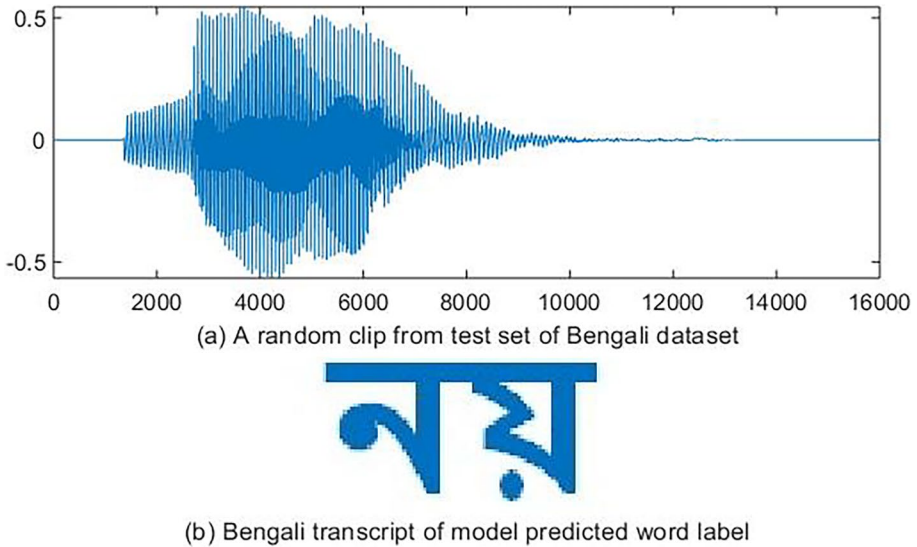
**Table 4** Layer-wise input-output shape and #parameter for end-to-end CNN model

| Layer (type)                 | Output Shape     | Param # |
|------------------------------|------------------|---------|
| input_1 (InputLayer)         | (None, 8000, 1)  | 0       |
| conv1d_1 (Conv1D)            | (None, 7988, 8)  | 112     |
| max_pooling1d_1 (MaxPooling1 | (None, 2662, 8)  | 0       |
| dropout_1 (Dropout)          | (None, 2662, 8)  | 0       |
| conv1d_2 (Conv1D)            | (None, 2652, 16) | 1424    |
| max_pooling1d_2 (MaxPooling1 | (None, 884, 16)  | 0       |
| dropout_2 (Dropout)          | (None, 884, 16)  | 0       |
| conv1d_3 (Conv1D)            | (None, 876, 32)  | 4640    |
| max_pooling1d_3 (MaxPooling1 | (None, 292, 32)  | 0       |
| dropout_3 (Dropout)          | (None, 292, 32)  | 0       |
| conv1d_4 (Conv1D)            | (None, 286, 64)  | 14400   |
| max_pooling1d_4 (MaxPooling1 | (None, 95, 64)   | 0       |
| dropout_4 (Dropout)          | (None, 95, 64)   | 0       |
| flatten_1 (Flatten)          | (None, 6080)     | 0       |
| dense_1 (Dense)              | (None, 256)      | 1556736 |
| dropout_5 (Dropout)          | (None, 256)      | 0       |
| dense_2 (Dense)              | (None, 128)      | 32896   |
| dropout_6 (Dropout)          | (None, 128)      | 0       |
| dense_3 (Dense)              | (None, 10)       | 1290    |

Total params: 1,611,498

Trainable params: 1,611,498

Non-trainable params: 0



**Fig. 10** (a) An arbitrary clip from the Bengali dataset (b) model generated class label

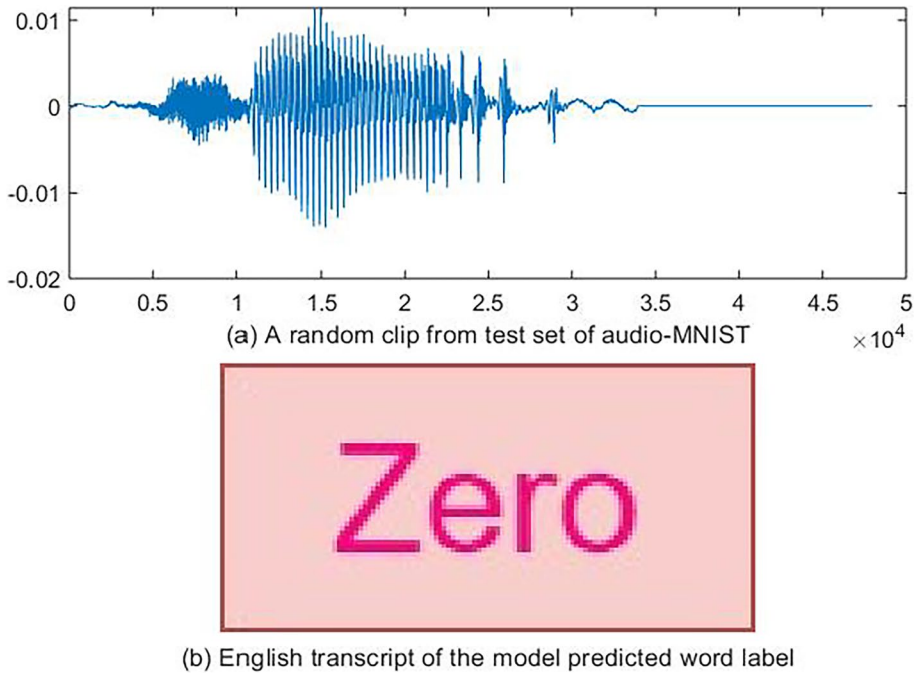
output shape and number of parameters in every layer of an end-to-end CNN model are shown in Table 4.

The predicted train, validation, and test accuracy from this end-to-end model are 97.72%, 96.64%, and 96.51%, respectively, for the Bengali dataset. When the audio-MNIST dataset is trained, as the frequency is 48000 Hz, even though the audios are resampled to 24000, the total trainable parameters are 4,839,146. The train, validation, and test accuracy are 99.64%, 98.37%, and 98.86%, respectively. To validate the proposed technique, the insightful functionality of the model is depicted in Figs. 10 and 11. A random clip from the test set of the Bengali dataset is fed to the model. The predicted word label is shown in Fig. 10. Figure 10a implies the raw audio clip from the Bengali dataset, and Fig. 10b shows the visual presentation of the predicted label of the corresponding audio as “Noi” (English equivalent nine). Similarly, Fig. 11 illustrates a random clip from the audio-MNIST dataset and its model-generated class. Figure 11a indicates a random audio clip from the audio-MNIST dataset and Fig. 11b signifies the visual presentation of the predicted label of the corresponding audio as “Zero”.

The outcome of the proposed architecture is compared with several pre-trained models, given in Table 5, to justify the robustness of the proposed architecture. As all pre-trained model takes at least two-dimensional input with some restriction of the minimum input size, we made the input feature of 62 x 30 into 75 x 75 by zero-padding. Table 5 shows that the proposed architecture generates almost equal test accuracy (in some cases higher) compared to the well-known pre-trained models ResNet50, MobileNet, VGG-19, and InceptionNet3 for both datasets. Moreover, the proposed architecture takes fewer parameters as compared to these models.

#### 4.1 Discussion

The superiority of the proposed method has been analyzed to justify the robustness of the proposed architecture in different aspects. The main objective of any deep learning model is to minimize the number of trainable parameters with satisfactory prediction

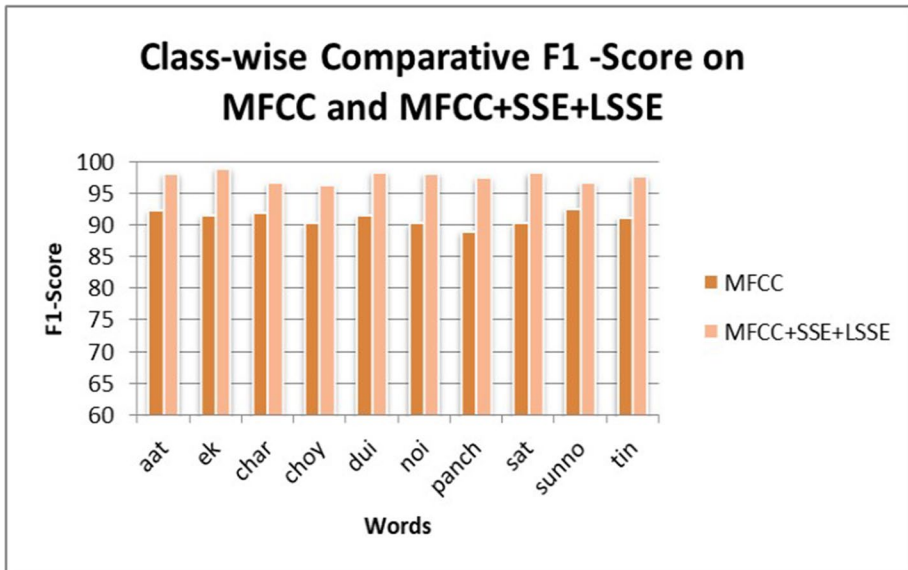


**Fig. 11** (a) An arbitrary clip from the audio-MNIST dataset (b) model generated class label

accuracy. As the parameters increase, the complexity and size of the model also increase accordingly. It has been observed that the number of trainable parameters for end-to-end speech recognition models (raw signal) proportionally increases its frequency. The total trainable parameters for the end-to-end model for the two above datasets are 1,611,498 and 4,839,146, respectively, which is almost 9 and 27 times more than the proposed feature extracted one-dimensional CNN. One additional advantage of the proposed feature-extracted CNN model is the input shape and other intermediate layers' parameters are fixed. They need not be tuned for the dataset of different frequency levels. The

**Table 5** Outcome of the proposed architecture with some pre-trained models

| Dataset                        | Models                | Trainable parameters | Test Accuracy (%)                  |
|--------------------------------|-----------------------|----------------------|------------------------------------|
| Bengali Isolated Digit Dataset | ResNet50              | 23.54M               | 98.84 $\pm$ 0.3                    |
|                                | MobileNet             | 3.216M               | 98.27 $\pm$ 0.3                    |
|                                | VGG-19                | 0.508M               | 97.04 $\pm$ 0.4                    |
|                                | InceptionNet3         | 16.045M              | 96.75 $\pm$ 0.4                    |
|                                | <b>Proposed model</b> | <b>182K</b>          | <b>98.52 <math>\pm</math> 0.3</b>  |
| audio-MNIST                    | ResNet50              | 23.54M               | 99.73 $\pm$ 0.1                    |
|                                | MobileNet             | 3.216M               | 99.43 $\pm$ 0.2                    |
|                                | VGG-19                | 0.508M               | 99.90 - 0.2                        |
|                                | InceptionNet3         | 16.045M              | 97.76 $\pm$ 1.20                   |
|                                | <b>Proposed model</b> | <b>182K</b>          | <b>99.52 <math>\pm</math> 0.36</b> |



**Fig. 12** Comparative F1-Score on MFCC and MFCC+SSE+LSSE of Bengali dataset

experimental result shows that the proposed model generates almost similar accuracy compared to the end-to-end model with 9 and 27 times less trainable parameters.

The most difficult is the selection of features to keep the accuracy level the same. Analyzing the speech signal, it has been observed that when the data records are of low-frequency range or collected from different recording devices, the recognition accuracy doesn't reach a satisfactory level when feeding a single type of feature. That's why hybrid features have been used for this study. The frequency domain MFCC features and the time domain SSE and LSSE are fused in the experiment. To justify this statement, we have extracted only 20 MFCC features from each clip of both datasets and fed them into CNN. The obtained average validation accuracy is 91.20% for the Bengali dataset. However, the audio-MNIST dataset generates 96.65% validation accuracy for 20 MFCC features. It is because the audio-MNIST samples are very high-frequency clips; moreover, they are almost noise-free and are recorded from a single device. However, our created datasets are recorded from the different specifications of devices and are not fully noise-free. The classwise comparative f1-score from 20 (only MFCC) and 30 (MFCC, SSE, LSSE) features for the Bengali dataset is shown in Fig. 12. In Fig. 12, for every word, two vertical lines, where brown and pink color represents the F1-score obtained by applying 20MFCC only and MFCC, SSE, and LSSE together.

## 4.2 Comparative study

To justify the superiority of the model, Table 6 shows the comparative accuracy obtained from the proposed model with the existing models used in the literature survey for the recognition of isolated words of two different languages. In Table 6, the proposed method generates 99.8% validation accuracy, which is higher than the existing method in [3, 6, 12, 17, 26, 30] for the audio-MNIST dataset. Similarly, for the Bengali Isolated Word dataset, the proposed approach generates 98.52% prediction accuracy, higher than [2, 10, 15, 29, 34].

**Table 6** Comparative accuracy with the existing models

| Dataset               | Author Name            | Feature  | Classifier                       | Accuracy |
|-----------------------|------------------------|--|----------------------------------|----------|
| Audio-MNIST           | Pawar et al. [26]      | MFCC   | HMM toolkit                      | 95%      |
|                       | Becker et al. [3]      | Spectrogram, waveform                                  | LRP based DNN                    | 77%      |
|                       | Gamit et al. [6]       | MFCC, LPC, STE, ZCR                                    | ANN                              | 86.66%   |
|                       | Mahalingam et al. [16] | MFCC   | LSTM                             | 96%      |
|                       | Masmoudi et al. [17]   | MFCC   | Frame-by-Frame<br>Neural Network | 98.8%    |
|                       | Kaur et al. [12]       | MFCC   | DNN                              | 97.05%   |
|                       | Shukla et al. [30]     | Spectrogram  | DNN                              | 98.1%    |
|                       | <b>Proposed model</b>  | MFCC+SSE+LSSE  | CNN                              | 99.8%    |
| Bengali isolated word | Lisa et al. [15]       | MFCC & Local Feature                                   | HMM                              | 89.47%   |
|                       | Ahammad et al. [2]     | MFCC   | BPNN                             | 98.46%   |
|                       | Gupta et al. [10]      | MFCC followed by PCA<br>for feature summariza-<br>tion | SVM, MLP, RF                     | 91.67%   |
|                       | Sharmin et al. [29]    | MFCC   | CNN                              | 98.37%   |
|                       | Sumon et al. [34]      | MFCC   | CNN                              | 74.01%   |
|                       |                        | <b>Proposed Model</b>                                  | MFCC+SSE+LSSE                    | 1D CNN   |

The network size of each model is calculated from its number of trainable parameters. First, for the end-to-end model total number of parameters is 1,611,498. Thus to train a single audio sample, the network would have to train 1,611,498 parameters. If the 32-bit floating point number is used for each parameter, the size will be 1,611,498 x 4 bytes to train. Thus the network will require 6.147MB of RAM. For the minibatch train, the network trains 70% data of 14000= 9800 samples/minibatch = 306 samples at a time. Thus approximately, it requires 306 x 6.147 = 1.837 GB of RAM.

On the other hand, our proposed network needs 182,458 parameters to train a single audio sample, and the network size would have 182,458 x 4 bytes of RAM. Even though to train the same Bengali dataset with 32 minibatch, the model requires 182,458 x 4 x 306 = 212.98 MB of RAM, which is almost nine times less than the end-to-end model.

When calculating the time complexity, it depends on the number of parameters (p), number of epochs (e), and number of training observations in each epoch(n). So, the time complexity can be measured as a notation  $O(p \times e \times n)$ . Since p will be large for the end-to-end model, the training time will also be prolonged for the end-to-end model as compared to the proposed CNN model.

Though the model generates a high accuracy, there is still scope to develop a generalized model to recognize a large set of spoken numeral data in a noisy environment with less trainable parameters.

## 5 Conclusion and future work

A reduced feature-extracted one-dimensional CNN has been designed in the proposed isolated spoken digit recognition method. Two datasets of isolated words in two different languages have been used for the experiment to justify the efficacy of the proposed technique. Two different

domain features, 20 MFCC (frequency-domain) and 5 SSE, 5 LSSE (time-domain), have been extracted from each audio sample and fed to the proposed architecture of the CNN model. The average train, validation, and test accuracy for the Bengali spoken digit dataset are 96.68%, 97.8%, and 98.52%, respectively. Similarly, 98.47%, 99.84%, and 99.52% for the audio-MNIST dataset, respectively, indicate the superiority of the proposed technique. The datasets are also trained using an end-to-end CNN model. The trainable parameters are approximately 9 and 27 times more than the proposed architecture for the two datasets, respectively.

The comparative model size and computational time complexity have been derived analytically. The proposed model requires almost nine times less memory than an end-to-end model to train 14000 Bengali audio clips and 27 times less to train 30000 English audio clips. Though the proposed approach has experimented with two isolated speech corpus of two different languages, we can claim this hybrid feature and model outperforms in classifying isolated words of any language. The proposed model is irrelevant to the language and the samples' frequency, meaning the model's complexity (time and space cost) does not grow on different frequency levels and languages. The major drawback of the proposed approach is the performance of the feature extracted CNN model degrades when excessive noise signals within clips. The noise within the audio signal didn't focus on cancellation. Data clips in the crowd may reduce the recognition rate by the proposed method. Further study is necessary to enhance the recognition in the crowd and further reduce the model size. The different combinations of features and model architecture may be designed in the future to get the expected level of accuracy.

**Funding** The authors did not receive support from any organization for the submitted work. No funding was received to assist with the preparation of this manuscript. No funding was received for conducting this study. No funds, grants, or other support was received.

**Data availability** The audio-MNIST dataset generated during and/or analysed during the current study are available in the kaggle repository from the web link: <https://www.kaggle.com/datasets/sripaadsrinivasan/audio-mnist>. That dataset is originally git repository and details of the data set are available in [https://github.com/soerenab/AudioMNIST/blob/master/data/audioMNIST\\_meta.txt](https://github.com/soerenab/AudioMNIST/blob/master/data/audioMNIST_meta.txt). All data generated or analysed during this study are included in this published article [3] (and its supplementary information files).

Our own created dataset of spoken Bengali isolated digit dataset is generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

## Declarations

**Conflict of interest** There is no conflict of Interest between the authors regarding the manuscript preparation and submission.

## References

1. Abdel-Hamid O, Mohamed AR, Jiang H, Deng L, Penn G, Yu D (2014) Convolutional neural networks for speech recognition. *IEEE/ACM Trans Audio, Speech, Language Process* 22(10):1533–1545
2. Ahammad K, Rahman MM (2016) Connected bangla speech recognition using artificial neural network. *Int J Comput Appl* 149(9):38–41
3. Becker S, Ackermann M, Lapuschkin S, Müller KR, Samek W (2018) Interpreting and explaining deep neural networks for classification of audio signals. *arXiv preprint arXiv:1807.03418*
4. Dikmese S, Sofotasios PC, Renfors M, Valkama M (2015) Subband energy based reduced complexity spectrum sensing under noise uncertainty and frequency-selective spectral characteristics. *IEEE Trans Signal Process* 64(1):131–145
5. Ferrer L, Lei Y, McLaren M, Scheffer N (2015) Study of senone-based deep neural network approaches for spoken language recognition. *IEEE/ACM Trans Audio, Speech, Language Process* 24(1):105–116

6. Gamit MR, Dhameiliya K (2015) Isolated words recognition using MFCC, LPC and neural network. *Int J Res Eng Technol* 4(6):146–149
7. Girshick R (2015) Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision* (pp 1440–1448)
8. Grozdić ĐT, Jovičić ST, Subotić M (2017) Whispered speech recognition using deep denoising autoencoder. *Eng Appl Artif Intell* 59:15–22
9. Guiming D, Xia W, Guangyan W, Yan Z, Dan L (2016) Speech recognition based on convolutional neural networks. In *2016 IEEE International Conference on Signal and Image Processing (ICSIP)* (pp 708–711). IEEE
10. Gupta A, Sarkar K (2018) Recognition of spoken bengali numerals using MLP, SVM, RF based models with PCA based feature summarization. *Int Arab J Inf Technol* 15(2):263–269
11. Kadyan V, Mantri A, Aggarwal RK, Singh A (2019) A comparative study of deep neural network based Punjabi-ASR system. *Int J Speech Technol* 22(1):111–119
12. Kaur G, Srivastava M, Kumar A (2017) Speaker and speech recognition using deep neural network. *Int J Emerg Res Manag Technol* 6:8
13. Kondhalkar H, Mukherji P (2019) A novel algorithm for speech recognition using tonal frequency cepstral coefficients based on human cochlea frequency map. *J Eng Sci Technol* 14(2):726–746
14. Krishnamoorthy P, Prasanna SM (2011) Enhancement of noisy speech by temporal and spectral processing. *Speech Commun* 53(2):154–174
15. Lisa NJ, Eity QN, Muhammad G, Huda MN, Rahman CM (2010) Performance evaluation of Bangla word recognition using different acoustic features. *Int J Comput Sci Netw Secur* 10:96–100
16. Mahalingam H, Rajakumar M (2019) Speech recognition using multiscale scattering of audio signals and long short-term memory of neural networks. *Int J Adv Comput Sci Cloud Comput* 7:12–16
17. Masmoudi S, Frikha M, Chtourou M, Hamida AB (2011) Efficient MLP constructive training algorithm using a neuron recruiting approach for isolated word recognition system. *Int J Speech Technol* 14(1):1–10
18. Nagajyothi D, Siddaiah P (2018) Speech recognition using convolutional neural networks. *Int J Eng Technol* 7(4.6):133–137
19. Nicolson A, Hanson J, Lyons J, Paliwal K (2018) Spectral subband centroids for robust speaker identification using marginalization-based missing feature theory. *Int J Signal Process Syst* 6(1):12–16
20. Palaz D, Doss MM, Collobert R (2015) Convolutional neural networks-based continuous speech recognition using raw speech signal. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (pp 4295–4299). IEEE
21. Paul B, Adhikary DD, Dey T, Guchhait S, Bera S (2022) Bangla Spoken Numerals Recognition by Using HMM. In *Computational Intelligence in Pattern Recognition* (pp 85–97). Springer, Singapore
22. Paul B, Bera S, Paul R, Phadikar S (2021) Bengali spoken numerals recognition by MFCC and GMM technique. In *Advances in Electronics, Communication and Computing* (pp 85–96). Springer, Singapore
23. Paul B, Dey T, Adhikary DD, Guchhai S, Bera S (2022) A novel approach of audio-visual color recognition using KNN. In *Computational Intelligence in Pattern Recognition* (pp 231–244). Springer, Singapore
24. Paul B, Mukherjee H, Phadikar S, Roy K (2019) MFCC-Based Bangla Vowel Phoneme Recognition from Micro Clips. In *International Conference on Intelligent Computing and Communication* (pp 511–519). Springer, Singapore
25. Paul B, Phadikar S, Bera S (2021) Indian regional spoken language identification using deep learning approach. In *Proceedings of the Sixth International Conference on Mathematics and Computing* (pp 263–274). Springer, Singapore
26. Pawar GS, Morade SS (2014) Isolated English language digit recognition using hidden markov model toolkit. *Int J Adv Res Comput Sci Softw Eng Jaunpur-222001, Uttar Pradesh, India*, 4(6)
27. Qadir JA, Al-Talabani AK, Aziz HA (2020) Isolated spoken word recognition using one-dimensional convolutional neural network. *Int J Fuzzy Logic Intell Syst* 20(4):272–277
28. Sarma M (2017) Speech recognition using deep neural network-recent trends. *Int J Intell Syst Des Comput* 1(1-2):71–86
29. Sharmin R, Rahut SK, Huq MR (2020) Bengali spoken digit classification: A deep learning approach using convolutional neural network. *Proc Comput Sci* 171:1381–1388
30. Shukla S, Jain M (2021) A novel stochastic deep resilient network for effective speech recognition. *Int J Speech Technol* 1–10
31. Si S, Wang J, Sun H, Wu J, Zhang C, Qu X, Cheng N, Chen L, Xiao J (2021) Variational information bottleneck for effective low-resource audio classification. *arXiv preprint arXiv:2107.04803*
32. Siniscalchi SM, Yu D, Deng L, Lee CH (2013) Exploiting deep neural networks for detection-based speech recognition. *Neurocomputing* 106:148–157

33. Song Z (2020) English speech recognition based on deep learning with multiple features. *Computing* 102(3):663–682
34. Sumon SA, Chowdhury J, Debnath S, Mohammed N, Momen S (2018) Bangla short speech commands recognition using convolutional neural networks. In 2018 international conference on bangla speech and language processing (ICBSLP) (pp 1–6). IEEE
35. Tripathi AM, Paul K (2022) When sub-band features meet attention mechanism while knowledge distillation for sound classification. *Appl Acoust* 195:108813
36. Vani HY, Anusuya MA (2020) Fuzzy speech recognition: a review. *Int J Comput Appl* 177(47):39–54
37. Veisi H, Mani AH (2020) Persian speech recognition using deep learning. *Int J Speech Technol* 23(4):893–905

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.