Journal of Ambient Intelligence and Humanized Computing (2023) 14:16903–16919
https://doi.org/10.1007/s12652-023-04706-6

ORIGINAL RESEARCH

Cepstral and acoustic ternary pattern based hybrid feature extraction approach for end-to-end bangla speech recognition

Mohit Dua[1] · **Akanksha**[1] · **Shelza Dua**[2]

Received: 28 November 2022 / Accepted: 19 September 2023 / Published online: 9 October 2023

Abstract
In the last three decades, a lot of work has been done for building Automatic Speech Recognition (ASR) systems for well-established languages such as English, Chinese, etc. However, for implementing a Large Vocabulary Continuous Speech Recognition (LVCSR) system for low resource languages, the research work is also growing rapidly in corpus-focused areas. Hence, there is a requirement of benchmarking large corpus in case of low-resource language like Bangla such that prejudice results can be avoided and limitations can be handled. In the proposed work, an openly available large-scale Bangla speech corpus provided by Google has been used. The work in this paper proposes a combination of image inspired features with well explored cepstral features to build front end feature extraction phase. It uses integration of Convolutional Neural Network (CNN) and bi-directional Long-short Term Memory (bi-LSTM) with Connectionist Temporal Classification (CTC) loss function to implement backend acoustic model. The experiments employ static and dynamic features of Mel-frequency Cepstral Coefficients (MFCC), Constant Q Cepstral Coefficients (CQCC), and Gammatone Cepstral Coefficients (GTCC) techniques, one by one, with Acoustic Ternary Patterns (ATP) features. The proposed work investigates the effect of these various hybrid front-end approaches with CNN, bi-LSTM and integration of these two models. The novelty of this paper lies in the fact that fusion of ATP with cepstral features improves performance of the proposed low resource language ASR system, where the proposed combination of ATP-dynamic CQCC features with integrated backend acoustic model shows a relative improvement of 10–15% in Word Error Rate (WER) over all other experimented combinations. Further to exploit the noise robust nature of GTCC features, the ATP-dynamic GTCC features with integrated CNN-bi-LSTM back-end model are evaluated in noisy scenario, also.

Keywords Bangla ASR · WER · MFCC · CQCC · GTCC · ATP · CNN · Bi-LSTM

1 Introduction

Speech contemplates as the primary medium of human interaction. However, interaction is no longer restricted to people but also includes machines. Automatic Speech Recognition (ASR) is a technology for facilitating human–machine interaction. For example, speech-based conversational assistants such as Apple Siri, Google Assistant, and Amazon Alexaare immensely popular, providing a wide range of services such as managing smart home devices and doing various activities via voice requests (Dua et al. 2022; Izbassarova et al. 2020). In recent decades, researchers showed so much interest in researching the automation of simple function that requires human–machine interaction (Nassif et al. 2019). A lot of studies, data collection, and research have performed by so many researchers for high-elevated languages like Spanish, English, and many others. Whereas in case of regional and low-resource languages such as Bangla and Punjabi etc., have tremendous opportunities for improvement (Kadyan. et al. 2018; Bhatt et al. 2021).

An ASR system is built mainly consisting of the front-end and back-end. At the front-end, the feature extraction techniques are applied to extract the relevant features,

✉ Shelza Dua
shelza_ecn@yahoo.com

Mohit Dua
er.mohitdua@nitkkr.ac.in

Akanksha
akanksha_32013102@nitkkr.ac.in

[1] Department of Computer Engineering, National Institute of Technology, Kurukshetra, India

[2] Department of Electronics Communication and Engineering, Punjab Engineering College Chandigarh, Chandigarh, India

and at the back-end part, all the prediction work happens. Traditionally, Mel frequency Cepstral Coefficients (MFCC) (Mohan 2014) has been introduced as speech feature extraction. However, in noisy environments, the reliability of MFCC diminishes. As a result, noise-resistant algorithms like Gammatone Frequency Cepstral Coefficient (GFCC) (Shao et al. 2009) and hybrids of other approaches are becoming popular. Originally, Constant Q Cepstrum Coefficients (Yu et al. 2017) were designed to extract the features in the Automatic Speaker Verification (ASV) System to perform spoof detection (Wang et al. 2017). As a hybrid approach, Acoustic Ternary Patterns (ATP) (Malik et al. 2020) has been also fused with other feature extraction techniques in ASV, computer vision, and image processing systems. The research in this paper also focuses on different front-end combinations, MFCC + ATP, GTCC + ATP, and CQCC + ATP, to extract valuable information from the speech signal.

Similarly, at the back-end of an ASR system, the Hidden Markov Model (HMM) (Renals et al. 1994) and Gaussian Mixture Model (GMM) (Pujol et al. 2004) have been the common popular approaches for the classification of the speech samples. Moreover, in recent few years, researchers have experimented with the variety of deep-learning and machine learning-based models such as Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), and Support Vector Machine (SVM) (Ganapathiraju et al. 2004) for audio classification, bi-gram, tri-gram and n-gram (Isotani et al. 1994) as a language model.

The introduction of external language models has recently demonstrated a considerable increase in end-to-end ASR accuracy and neural machine translation. This method is known as shallow fusion, in which it combines the external language model with a decoder in the logged probability dimension for decoding (Kumar et al. 2014; Mori et al. 2021).A lot of research has been done on ASR for Bangla language utilizing different speech corpus, researching feature extraction strategies such as Mel-frequency Cepstral Coefficients(i.e., MFCC), Linear Prediction Coefficients (i.e., LPC), and Dynamic Time Wrapping (i.e., DTW), acoustic models like Deep Neural Networks (i.e., DNN), Hidden Markov Model-Gaussian Mixture Model (HMM-GMM), and language modeling techniques like N-grams, mono phone, and tri phoneme models. Most of the ASR systems in the Bangla language are developed using small datasets like digits, isolated words, and phonemes. However, in the case of large vocabulary continuous speech recognition (LVCSR), we have many challenges due to the non-availability of a large corpus, morphological parsing difficulty, and accent variability in the Bangla language (Samin 2021; Kibria et al. 2020).

The work presented in this paper contributes by using modified feature extraction techniques with the help of hybrid acoustic modeling and decoding strategy to develop an ASR system more robust.

The remainder of the article is structured as follows: Sect. 2 discusses the recent work and contribution, and then Sect. 3 discusses the fundamental techniques required to develop the proposed system. Section 4 represents the architecture of the proposed ASR system; Sect. 5 explains the experimental details and the performance analysis, followed by Sect. 6, which concludes the whole proposed work.

2 Related work and contribution

This section examines the relevant works and our contribution to this field. Experiments on multiple audio feature extraction approaches at the frontend and diverse classification models at the backend expand the literature. For ASR systems to operate moderately effective, a vast amount of training data is required. Large languages like as English, Chinese, and Spanish have several state-of-the-art speech corpora for successful the ASR system training. Whereas, such a big amount of data for regional languages like Bangla, Punjabi, Hindi, and so on is not available and only a few scholars are working upon them (Wang et al. 2019; Jain et al. 2019). In reality, it is predicted that just approximately 1% of the world's languages have the essential voice corpus needed to train an ASR system (Scharenborg et al. 2017).

In 1920, the first voice recognition system was developed. Later, the voyage of voice recognition technology was enhanced by the independent effort of researchers from all around the world. In the late 2000s, research on the Bengali ASR system began. In (Karim et al. 2002), proposed a model for the recognition of Bengali Spoken letters. The pioneering work in this field relied heavily on self-created small datasets and statistical methodologies. The first use of Neural Networks was seen in 2009, by authors .Paul et al. (2009). The authors began by pre-processing the input speech with pre-emphasis and the hamming window. Then, to generate voice characteristics, a 12-dimensional Linear Predictive Coding (LPC) is utilized. Finally, the speech characteristics are sent into an Artificial Neural Network (ANN), which is used to recognize speech. However, the study was done with a small sample size of four people, and no evidence of performance evaluation was provided. Muhammad et al. (Muhammad et al. 2009), proposed a model for Bangla digits recognition, where authors implemented the model using MFCC at front-end and HMM at back-end and achieved the word accuracy of more than 90% and proposed that a portion of the performance decrease was caused by a dialectical divergence. Female uttered digits exhibited better accurate rates than male uttered digits in gender dependent trials.

In year 2010, (Rahman et al. 2010) introduced a segment method to segment the continuous waveform from Bangla

Speech. The authors used mean windows for separating each word from continuous speech. Based on the gaps in every fragmented word, each segmented word was then assigned to one of three groups: mono, di, or tri syllable. However, the size of the dataset was very small at which the researchers achieved 98.48% accuracy. (Rahman and Khatun 2011), proposed a model for the isolated words recognition in Bangla corpus. They implemented the MFCC at front-end and included the direct Euclidean distance measuring technique and achieved recognition rate of 84.28% and 96% respectively for multiple and single speakers. In (Ahmed et al. 2015), they implemented the model using the deep belief network. In their model, they applied MFCC at the front-end and then those features were trained by applying them to the generative ANN model, which was composed using the HMM with Boltzmann machine's multiple layers. On the proposed model authors achieved the overall accuracy of 94.05%. (Nahid et al. 2016), proposed a noble approach to recognize the Bangla digits. To implement this, they used the software like Avro, which is used is Unicode based writing software and CMU Sphinx4 named speech recognition API and achieved the 75% accuracy.

In (Bhowmik et al. 2017), used the DNN model for the first time, in case of Bangla corpus. The architecture used deep convolutional autoencoders' stacks that used pre-trained MFCC as input. Furthermore, after auto encoders' training, a three-layer multi-layer perceptron was employed to forecast the phoneme probabilities. In a self-created data-set that is no longer available, the baseline obtained 82.5% phonetic categorization accuracy. (Al Amin et al. 2019) proposed the hybrid model approach by combining the GMM and DNN model with HMM and proposed that the GMM-HMM performed better.

Since the year 2020, a new era of research started in the area of Bangla Speech Corpus, where so many researchers performed their experiments and proposed their work. (Paul et al. 2021) proposed a system for continuous speech recognition. Researchers stated that a subspace Gaussian mixture model is used in combination with quad-gram as the language model LM and HMM as the AM in the ASR system. In a triple cross-validation trial, the system's WER utilizing an appropriately trained LM was observed as low as 5%. (Mandal et al. 2020) proposed an end-to-end model, in which authors implemented model by using the CTC-based CNN-RNN model and they achieved the total WER of 13.67%.

CQCC is one the most popular choice for feature extraction in the area of ASV systems. In (Cai et al. 2017) explored the CQCC features that are later fed to GMM classifier, fully connected DNN and bi-LSTM model. The proposed architecture improves the system's performance significantly in spoofed environment. In (Saranya et al. 2018) proposed a novel approach using CQCC + MFCC for replay attack detection in ASV system, where GMM model was used as at the back-end to perform the relevant task (Chakravarty et al. 2023b).

In (Javed, A. et al. 2021) explored the joint ATP and GTCC features to protect the Voice-controlled systems from voice spoofing attacks in single-hop and multi-hop networks.

Recently, in (Adhikary et al. 2021) proposed an approach using Long short-term memory (LSTM) and gated recurrent units (GRU) at the back-end (Joshi et al. 2023) and MFCC at the front-end and achieved overall accuracy of 47% using GRU and 45.81% using LSTM model. In (Das et al. 2021) used a mixed-language corpus for their work, in which authors have used the Bangla-English spoken digits and in noisy environment created their dataset and used CNN model and MFCC at the back-end and front-end respectively.

Recently in, (Yang et al. 2022) proposed a model based on conformer, wherein, the acoustic modeling has been performed using the convolutional augmented based attention mechanism. The authors used CHiME-4 corpus and achieved the WER of 6.25%. The advantages of the proposed model are that it reduces the total training time, while consuming the relatively smaller model size. The disadvantage of this proposed model is that on increasing the number of encoders the WER also increases, which down-grades the model performance. In (Dua and Akansha 2023), author used hybrid features by combining MFCC and CQCC for Gujarati Language Automatic Speech Recognition.

Also, in, (Rakib et al. 2022) proposed the work for improving the Bangla ASR system by utilizing N-gram language model. In the research, authors implemented the pre-trained wav2vec2 model on Bengali Common Voice 9.0 speech dataset. The advantage of the proposed work is that the WER has improved to 4.66% for robust ASR system, which is relatively lesser compared to available systems. However, the authors have used only fine-tuned the available pre-trained model by performing the hyper-parameter tuning to obtain the optimal values parameter of training dataset. In (Showrav et al. 2022) proposed the work to improve the ASR system's performance for Bengali language corpus by applying transfer learning framework on end to end (E2E) structure. The proposed work freezes the feature extractor and uses the learning rate of 0.0003, which may down-grade the system's performance.

After performing the literature survey, it has been observed that most of the researchers have used generally MFCC as front-end and GMM-HMM based model as the backend. Motivated by the above approaches employed in ASR system domain (Muhammad et al. 2009; Ahmed et al. 2015; Bhowmik et al. 2017; Rakib et al. 2022) the proposed system uses the three Cepstral coefficients-based feature extraction techniques i.e., MFCC, CQCC, GTCC (Chakravarty et al. 2023a) and one image-based ATP technique propose the novel combinations of MFCC + ATP, CQCC + ATP, GTCC + ATP. Furthermore, as shown in the preceding discussion, a deep learning-based categorization model improves the system's performance. Hence, the proposed system makes use of Convolutional Neural Network (CNN) and bi-directional long-short model (bi-LSTM) as their back-end. The dataset used for training and evaluation purpose in Bangla Speech dataset provided by Google. The following given points represent the contribution of the proposed system:

- The proposed system makes use of static features of MFCC, GTCC and CQCC based Cepstral coefficients with image-based ATP features. Dynamic features of MFCC, GTCC and CQCC techniques also fused with ATP features separately to make the ASR system.
- Two standalone acoustic models (2D-CNN, bi-LSTM) are used to make the back-end of the proposed ASR system. In the proposed system, hybrid CNN + bi-LSTM model followed by CTC loss function are also implemented at the backend as the acoustic model.
- The Bangla Speech Dataset provided by Google is used to train and evaluate state-of-the-art ASR systems.
- The performance of these hybrid feature based ASR system is evaluated using the WER as the performance metric. The novelty of this paper lies in the fact that fusion of ATP with cepstral features improves performance of the proposed low resource language ASR system, where the proposed combination of ATP-dynamic CQCC features with integrated backend acoustic model shows a relative improvement of 10–15% in Word Error Rate (WER) over all other experimented combinations.
- To test the robust nature of GTCC features, we inject two forms of noise—multiplicative street noise and additive babble noise into the clean dataset at two different signal-to-noise ratios (SNRs): 0 dB SNR and 5 dB SNR. To replicate noisy situations, noise is purposefully added to clean dataset. In this noisy environment, the ATP-dynamic GTCC characteristics with an integrated CNN-bi-LSTM back-end model are evaluated. This study aids in understanding how well the model operates in the presence of noise and indicates the resilience of the GTCC features under noisy settings.

3 Preliminaries

The fundamentals used to implement the proposed ASR system are discussed in this section.

3.1 Mel-frequency cepstral coefficients (MFCC)

For implementing the ASR systems Mel-frequency Cepstral Coefficients (MFCC) is the most widely used Cepstral analysis-based feature extraction technique. The reason behind is that MFCC has the capability to capture the phonetically important features from the speech (Kumar et al. 2014, Vergin et al. 1999).

Windowing technique has used to slice the waveform of audio into the sliding frames and then Fast Fourier transform (FFT) or Discrete Fourier transform (DFT) and Mel-filter bank has applied and helps in providing the required data in frequency domain and mapping the observed frequency respectively. The filter and the glottal source are separated in the cepstrum using the log of power spectrum acquired from the Mel filter bank. The step-wise approach to extract the MFCC features is given below:

- Initially, we perform the pre-emphasis operation on the recorded speech signal $S(n)$. As a result, we get the pre-emphasized signal containing higher frequencies.
- Then we slice the input $S(n)$ into the smaller frames and then apply framing and windowing $W(n)$ to remove edge discontinuities.
- After windowing operation, to separate the energy contained within every frequency spectrum Discrete Fourier Transform (DFT) is applied. The equation to represent the speech signal into frequency range having power spectrum is:

$$|F(j)|^2 = \left|\sum_{j=1}^{P} f(j).e^{\left(-k2\pi jl/P - 1\right)}\right. \tag{1}$$

where,
$1 \le l \le -1$, and $|F(j)|^2$ = Power Spectrum.

- Diverse band pass filters are used to filter the spectrum produced by the DFT, and each frequency band's power is enumerated. Then, on the produced signal, we apply the log method with Mel filter bank to achieve the Spectrogram.
- The Mel coefficients are transformed back into the time domain using the discrete cosine transform (DCT). Later, 13-MFCC feature coefficients are produced for each frame from the DCT results.

The following is a representation of the filter bank's output equation when used to the energy spectrum:

Let $\propto_i(j)$ = filter response

$$e(i) = \sum_{j=1}^{\left(P/2\right)} |F(j)|^2.\alpha_i(j) \tag{2}$$

We can represent the obtained MFCC features as:

$$M(f) = \sqrt{{}^{2}/{}_{X}} \sum_{j=0}^{X-1} \log\left[e(j1)\right].\cos\left[.\left({}^{2k} - {}^{1}/{}_{2}\right).{}^{\pi}/{}_{X}\right] \tag{3}$$

3.2 Gammatone cepstral coefficients (GTCC)

Gammatone Cepstral Coefficients (GTCC) falls under the group of the noise resilient feature extraction approach. It is more through model based on the scale of Equivalent Rectangular band (ERB) and the collection of Gammatone Filter banks (Arafa et al. 2018; Rademacher et al. 2006).

The early operations, such as windowing and Fourier transform, are comparable to those done by MFCC. Then, the produced output after the DFT function is filtered with the Gammatone filter bank and the final feature vector generated which contains the total 14 coefficients having features as 13 Cepstral coefficients and one energy coefficient. To extract the GTCC feature coefficient, we can use the following given equation:

$$G(x) = \sqrt{\frac{2}{K}} \sum_{L=1}^{K} \log\left(S_p\right) \mathrm{Cos}\frac{l\pi}{K}\left(j - \frac{1}{2}\right)\ 1 \le j \le J \tag{4}$$

where,

K= Quantity of filters contained in the filter bank

$S_p =$ pth Energy band of Spectral

x= Total Coefficients' Number

3.3 Constant Q cepstral coefficients (CQCC)

During the implementation of an ASV system, Constant Q Cepstral Coefficients (CQCC) feature extraction is utilized to extract meaningful information from the captured speech signal. In recent time, this method has recently been shown to be the most feasible for the creation of reliable and accurate ASV systems (Oh et al. 2014; Mittal et al. 2021).

The Constant Q Transform (CQT) is used in the CQCC feature extraction procedure, which then takes the log of the powered spectrum. It also uses resampling before computing the DCT(Oh, S. Y & Chung, K. 2014; Mittal et al.2021). It returns CQCC features after setting the number of feature coefficients. The following is a mathematical depiction of the CQCC feature extraction approach:

$$C_{PQR}(s) = CQT(p(t)) \tag{5}$$

$$C_{CQCC}(i) = \sum\nolimits_{s=0}^{I} log|C_{PQR}(s)|^2 cos\{i(s - 0.5)\pi|I\} \tag{6}$$

In this case, Eq. (5) determines the Constant Q Transform (CQT) of the input speech signal $p(t)$ in $C_{PQR}(s)$, Eq. (6) determines i, total number of CQCC parameters in CCQCC(j), where I is the number of linearly spaced bins and e denotes the number of bins to index into.

3.4 Acoustic ternary patterns (ATP)

Local Ternary Patterns (LTP) has been studied as one of the most popular descriptors in the field of image analysis and computer vision. The primary premise of LTP is to compare each pixel in a picture to its surroundings. When an image pixel is compared to its neighbors, it generates binary values of '0' or '1'. This aids in the summary of a local structure in an image as well as the generation of powerful feature descriptions (Aziz et al. 2019). Face recognition, texture analysis, and ASV systems are some of the examples of promising uses. LTPs have a minimal processing cost and are resistant to monotonous grey scale variations. Similar to image descriptor 1-D LTP features, we calculate the Acoustic Ternary Patterns (ATP) features (Malik et al. 2020) for acoustic signals by dividing the whole speech into the frames, and then inside each frame compare the neighboring values from threshold values. Instead of taking the next pixel in image processing, we take the next speech signal value. A local ATP response is calculated by taking a constant linear distance $\pm\propto$ from central sample and quantized between $+1$ & -1 and the below three-valued method is obtained as:

$$F\left(n^j, c, \propto\right) = \begin{cases} -1, & n^j - \ (c- \propto) \le 0 \\ 0, & (c+ \propto) < n^j < (c- \propto) \\ +1, & n^j - \ (c+ \propto) \ge 0 \end{cases} \tag{7}$$

The detailed procedure to extract these ATP features has explained in Sect. 4.2.

3.5 Two-dimensional convolutional neural network (2D-CNN)

A Convolutional Neural Network (CNN) (Haque, M. A., 2020) is particularly adept at processing input with matrix architecture, like an image. Lately, CNN models have been used in the ASR systems as a classification model. A typical CNN model is made of so many different layers, such as, convolutional, polling and fully-connected layers. In case of two-dimensional CNN model, the model uses the two-dimensional convolutional kernels as it moves across 2-D on the input.

3.6 Bi-directional long-short term memory (bi-LSTM) model

The Long Short-Time Memory (LSTM) (Kim et al. 2019) is a kind of RNN that can learn long-term dependencies. LSTM excels at memorizing information for lengthy periods of time. A bidirectional LSTM (bi-LSTM) differs from a conventional LSTM in that our input operates in both directions i.e. forwards and backwards. Due to its flow nature the bi-LSTM model makes the information sequence in both, past to future and future to past.

3.7 Connectionist temporal classification (CTC) loss

Connectionist Temporal Classification (CTC) loss (Scheidl et al. 2018) is a technique often used in DNN for sequence like problems such as handwriting and speech recognition etc. This method is required when we suppose to align the input with the desired output. This is accomplished by adding up the likelihood of potential input to goal alignments, which results in a loss value that may be differentiated with regard per input node.

4 Proposed architecture

The architecture of the proposed method is described in this section. The suggested ASR system's design is depicted in Figs. 2, 4, 6. The frontend uses three different cepstral features, MFCC, GTCC, CQCC, and then combined them with image-based features ATP. The proposed system is trained and evaluated using the state-of-the-art Open SLR's Bangla language dataset. The Bangla speech corpus is divided into three parts: training, validation and testing. Training accounts for 70% of the corpus, 20% for validation, while testing accounts for the remaining 10% using unique set of datasets.

The processes in GTCC with CQCC feature extraction methods are quite analogous to those in the MFCC method. As mentioned before, at the back-end the proposed work investigates the three different models. These models are 2D-CNN, bi-LSTM with hybrid CNN based encoder bi-LSTM model. As a decoder we have used the CTC based greedy decoder provided by Keras and CTC loss as the loss function. Bangla based dictionary has been used in the data generation. We have classified our proposed systems broadly into three systems, such as system 1 is based on MFCC + ATP based features, system-2 is based on the GTCC + ATP based features and system-3 is based on the CQCC + ATP feature techniques.

All these three models take the Cepstral coefficients as an input with the image-based ATP features and generate the recognized values. Both static (first-order) and dynamic (delta Δ, delta-delta Δ–Δ) features of cepstral coefficients have used for the evaluation of these models. Out of all these features CQCC and ATP features are going to be used for the first time, in case of speech recognition. All three of the suggested ASR systems are discussed in the next sub-section.

4.1 MFCC + ATP based ASR system (System-1)

In Fig. 1, we have illustrated the procedure to extract the MFCC and ATP features. In MFCC, the first 13 coefficients expanded to get the maximum 39 features.

The first 13 features are known as static features, while the rest are dynamic formed by evaluating first derivatives and second order derivatives of the static features respectively known as delta (Δ) and delta-delta ($\Delta - \Delta$) features.

```
Function 1: $mfcc\_Features()$
Input: Acoustic signal termed as "speech"
Audio files are in .flac file format
$mfcc\_Features(speech)$ {
        $[vi, f_v] =$ audioread(speech)
    $O_i = mfcc\ (vi, f_v)$ // log energy = ignore
    $O_m = median\ (O_i)$ // to quantize matrix $O_i$ as (columns' size, 1)
    $Del = deltas(O_m)$
    $Del_{del} = deltas\ (deltas\ (Del))$ // to extract the $\Delta - \Delta$ features};
Output: 39 MFCC Cepstral coefficients // each file has (1, 39) as dimension
```

To extract the MFCC features, function 1 has used that leverages the built-in *mfcc*() function and assign the energy to ignore.

The output generated by the *mfcc*() will be in the form of (Dua and Akansha 2023; Das et al. 2021) dimensions. Hence, to merge it with other features median function has used. Motivated by the use of 2D- LTP in image processing (Das et al. 2021), we implemented this hypothesis for 1D- voice signal to adequately describe the acoustic signal and called them Acoustic Ternary Patterns (ATP). When applied to 1-D signals such as audio, the ATP approach aids in obtaining important information about the audio's local temporal dynamics. In total we have calculated 20 ATPs' feature, including 10 upper and 10 lower features in the dimension of (Dua et al. 2022; Jain et al. 2019). Function 2 has leveraged to extract the required ATP features. It takes the audio signal as an input and produces the ATP features as output.

Two-Dimensional Convolutional Neural Network (2D-CNN) trained using the extracted static, delta and delta-delta features of MFCC by fusing them with 20-ATP features. The suggested 2D-CNN design is made up of four types of layers: Convolutional, Max Pooling,

Flatten and Dense layer. Two units with soft-max activation make up the final dense layer. Dropout layers of 20% are also included to the design to prevent overfitting. The Learning rate of 0.01 has used in the model. Figure 2 represents architecture of 2D-CNN model implemented in our ASR system.

```
Function 2: ATP_Features()
Input: Acoustic signal termed as "speech"
Audio files are in .flac file format
$ATP\_Features(speech)${
Neighbor = $n(j + (p - (\frac{Q}{2}))$ ;
$Ltp_{lower}(1, i) = 1 * frame_{lower}(1,i) + 2 * frame_{lower}(2,i) + 4 * frame_{lower}(3,i) + 8 * frame_{lower}(4,i) + 16 * frame_{lower}(5,i) + 32 * frame_{lower}(6,i) + 64 * frame_{lower}(7,i) + 128 * frame_{lower}(8,i)$;
$Ltp_{upper}(1,i) = 1 * frame_{upper}(1,i) + 2 * frame_{lupper}(2,i) + 4 * frame_{upper}(3,i) + 8 * frame_{upper}(4,i) + 16 * frame_{upper}(5,i) + 32 * frame_{upper}(6,i) + 64 * frame_{lupper}(7,i) + 128 * frame_{upper}(8,i)$
$i = i + 1$;
$Ltp_{lower\_hist} = hist(Ltp_{lower}, 10)$;
$Ltp_{upper\_hist} = hist(Ltp_{upper}, 10)$;
$Ltp_{full} = [Ltp_{upper\_hist} Ltp_{lower\_hist}]$;
$return\ Ltp_{full}$
}
Variables:
$n = input\ audio\ signal$
$Q = no.of\ neighbors$
$j, i, p = looping\ variables$
        $Ltp_{upper_{hist}}$ = LTP upper histogram
        $Ltp_{lower\_hist}$ = LTP lower histogram
Output: Matrix of ATP features // each audio has (1, 20) as dimension
```

Proposed bi-LSTM model used the 5 layers of bi-LSTM, in having 50, 100, 150, 200, and 250 units each and relu as activation function. In which we apply the input on the first layer. And then we imposed the 20% dropout after each LSTM layer, the output of such layers is transmitted to a dense layer of 24 units. The output of this dense layer is sent to the final layer, which is a dense layer with soft-max activation function. And the proposed model is trained using the static and dynamic features of above-described feature extraction method with 20-ATP features. Figure 3 represents the proposed ASR system on the MFCC + ATP based features by using bi-LSTM as the back-end model.

In our proposed hybrid model, we have implemented an end-to-end model using the Deep-CNN layers with the very Deep Convolutional Network (VGG Net) architecture as an encoder, after that, stacked bidirectional long short-term memory (BLSTM) layers are added. Three two-dimensional (2-D) convolution layers, each having 512 filters of shape 1 × 7, encode the input text sequence embedding followed by layer normalization. After the first convolutional layer, we have added a 2-D max pooling layer. After this encoder three layers of 256-unit bidirectional LSTM layer in each direction followed by layer normalization in each layer. Dense layer with soft-max as activation function has also applied at the end. Figure 4 represents the proposed-on hybrid 2D-CNN + bi-LSTM as the back-end model in our ASR system.

4.2 GTCC + ATP based ASR System (System-2)

Figure 5, illustrates the procedure to extract the GTCC and ATP features. The back-end acoustic model are same as we have implemented in the system-1. In case of GTCC, the early operations, such as windowing and Fourier transform, are comparable to those done by MFCC.

Then the produced output after DFT function is filtered with the Gammatone filter bank and final feature vector generated which contains the total 42 coefficients. Later, these features have fused with the 20-ATP features having 10 lower and 10 upper bound features. Function 3 shows the speech signal as input and at first generates the $coef_i$ [] containing in total 14 GTCC feature coefficients (1 energy coefficient + first 13 GTCC features) extracted GTCC features as output in the form of Del_{del}[].

```
Function 3: gtcc_Features()
Input: Acoustic signal termed as "speech"
Audio files are in .flac file format
$gtcc\_Features(speech)$ {
        $[vi, f_v]$ = audioread(speech) ;
        $[coef]_i = gtcc\ (vi, f_v(f_v * 0.03), (f * 0.02))$ //setting windows and overlapping length respectively to $(f_v * 0.03), (f * 0.02)$
        $O_m = median\ (coef_i)$ // to quantize matrix $coef_i$ as (columns' size, 1)
        $Del = deltas(O_m)$
        $Del_{del} = deltas\ (deltas\ (Del))$ // to extract the $\Delta - \Delta$ features};
Output: 42 GTCC Cepstral coefficients // each audio file has GTCC features in (1, 42) dimension
```

4.3 CQCC + ATP based ASR System (System-3)

The Constant Q Transform (CQT) is used in the CQCC feature extraction procedure, which then takes the log of the powered spectrum. It also uses resampling before computing the DCT (Rahman et al. 2011, Rakib et al. 2022). It returns CQCC features after setting the number of feature coefficients. Function 4 shows the speech signal as input and extracted CQCC features as output in the form of Del_{del}[]. In function, $coef_i$[]

gives the first 20 CQCC features, whereas output contains the in-total of 60 CQCC features.

This is the first time we are introducing the CQCC features in the field of the Speech recognition, before that this feature has utilized only in the area of ASV systems. We calculated total 60 features of CQCC by extracting the 20 static, and 20 first order derivative (Δ) and 20 s order derivatives ($\Delta - \Delta$). Figure 6, illustrates the procedure to extract the CQCC and ATP features, where-as for developing the ASR system the back-end model we have implemented are similar to the models used with system-1.

```
**Function 4:** $cqcc\_Features()$
**Input:** Acoustic signal termed as "speech"
Audio files are in .flac file format
$cqccc\_Features(speech)$ {
        $[ai, f_v] = $ audioread(speech) ;
        $[coef]_i = cqcc\ (ai, f_v(f_v * 0.03))$ //setting windows to $(f_v * 0.03)$
        $CQCC = median\ (coef_i\ )$ // to quantize matrix $coef_i$ as (columns' size, 1)
        $Del = deltas(CQCC)$
        $Del_{del} = deltas\ (deltas\ (Del))$ // to extract the $\Delta - \Delta$ features};
**Output:**60 CQCC Cepstral coefficients // each audio file has CQCC features in (1, 60) dimension
```

5 Experimental details & performance analysis

To extract the features at the front-end MATLAB tool has used. For the back-end code python has used and for the execution environment Jupiter notebook has utilized. We complied our model using the "Adam" optimizer. Further, the connectionist temporal classification (CTC) loss function is used to train the model. Model-Checkpoint is used to save model weights such that the weights that produced the best performance can be utilized later.

5.1 Dataset

The dataset used to implement our models are Large Bengali ASR dataset provided by OpenSlr. It is the existing largest dataset for the Bangla language. The total duration of this dataset is 250 h (Xiao et al. 2018). We have divided the whole corpus into three parts i.e., training dataset, validation and testing dataset, where all three datasets are having different speech instances with respect to each other. Training accounts for 70% of the corpus, 20% of the corpus has been used for validation, while testing accounts for the remaining 10% using unique set of datasets. Each folder has two subfolders one named as train_clean, test_clean, val_clean accordingly in which we have.flac files of the audio, text file and other subfolder named as train_all, test_all and val_all in which we have collected the pickle dumb and text file. Table 1 summarizes about the required dataset.

We injected random street noise and babble noise into the audio dataset using different techniques—multiplicative and additive approaches. The both noises taken from the NOIZEUS dataset (Hirsch et al. 2000). This dataset contains noise at 0 dB, 5 dB, 10 dB and 15 dB. However, we are using only 0 dB and 5 dB noise for our experiments. The length of both noisy audio sample and clean audio samples are different. Function 5 provides the pseudo code to add the street noise to the clean audio sample.

```
**Function 5:** $noisyDataset()$
**Input:** Acoustic signal, street noise at 0 dB or 5 dB SNR
    1.  $[a, f] = audioread(speech)$
    2.  $[b, f1] = audioread(noise)$
    3.  $Min\_len = min([len(a), len(noise)])$
    4.  $q = a(1{:}Min\_len).* b(1{:}Min\_len)$
    5.  $return\ q$
**Output:** $q(noisy\ signalafter\ multiplying\ street$
        $noise\ to\ clean\ speech\ signal)$
```

- Line 1: The function *audioread* is used to read the audio data from the files specified in the speech and noise variables. After execution, a will be a matrix containing the audio samples of the clean speech, and f will be the frequency of the clean speech audio. Similarly,b will be a matrix containing the audio samples of the street noise, and $f1$ will be the corresponding frequency.
- Line 3: This line calculates the minimum length between the audio samples of the clean speech and the street noise.
- Line 4: This line creates the noisy dataset by multiplying the audio samples of the clean speech and the street noise element-wise. The $.*$ operator performs element-wise multiplication on the corresponding elements of the two matrices a and b. The resulting matrix q will contain the mixed audio samples of the noisy dataset.

Similarly, we have added additive babble noise to our clean audio dataset. the babble noise audios are selected from NOIZEUS dataset (Hirsch et al. 2000). The line wise description of the Function 6 is given below:

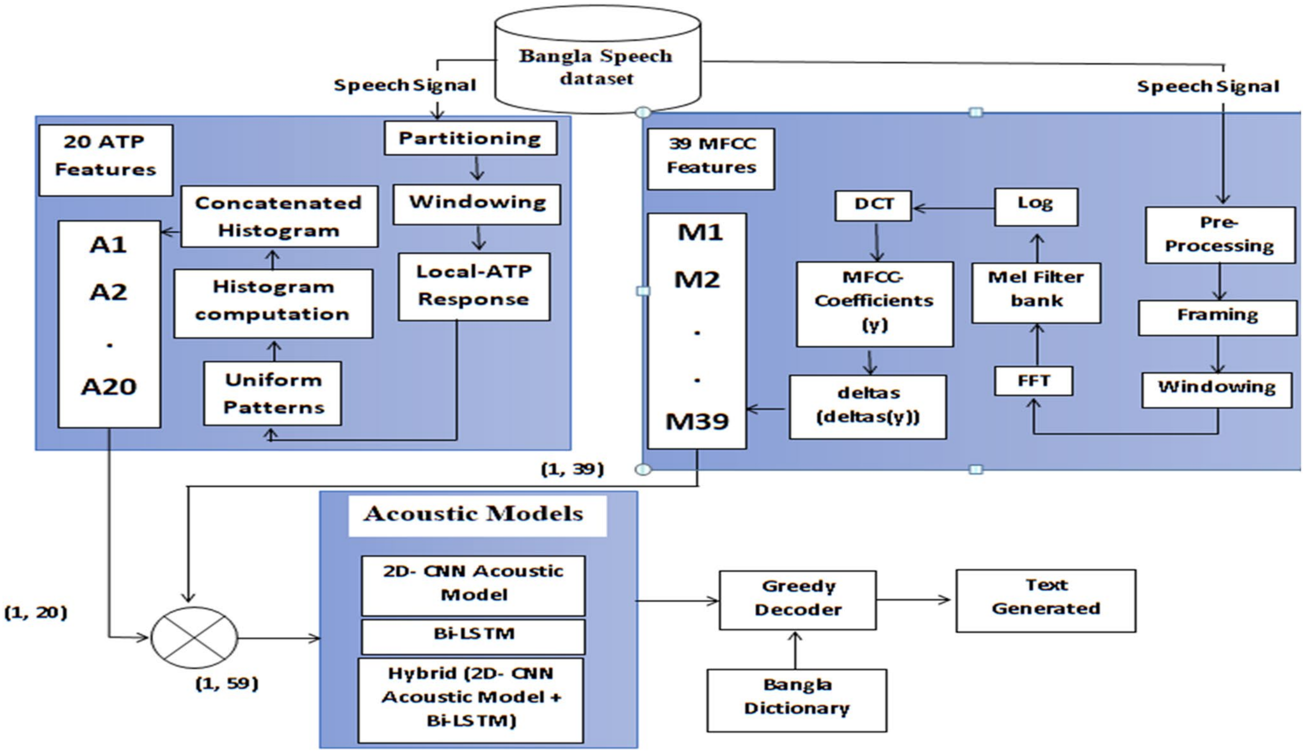

Fig. 1 Proposed Joint MFCC + ATP based ASR system (System 1)

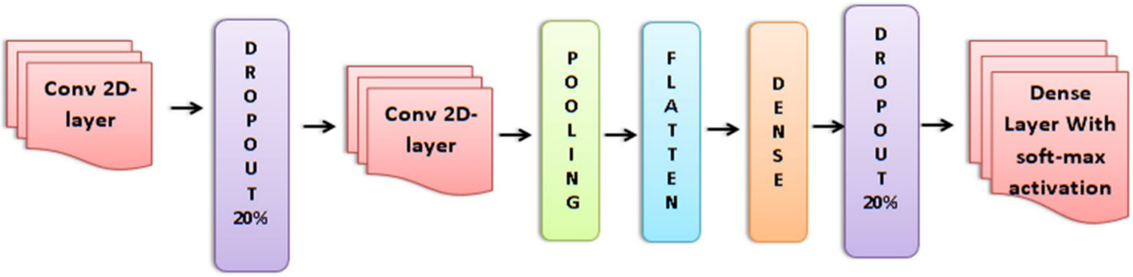

Fig. 2 Architecture of 2D-CNN Model

Function 6: ***addBNoiseToCleanSignal(clean _ signal, babble_noise, SNR_dB)***

Input: Acoustic signal, babble noise at 0 dB or 5 dB SNR

```
1.  if length(clean_signal) != length(babble_noise):
2.  if sampling_frequency(clean_signal) != sampling_frequency(babble_noise):
3.  babble_noise = resample(babble_noise, length(clean_signal)
    else:
4.  babble_noise = trim_to_len(babble_noise, length(clean_signal))
5.  babble_noise = normalize(babble_noise, amplitude(clean_signal))
6.  clean_signal_power = mean(square(clean_signal))
7.  target_noise = clean_signal_power / ratio(SNR_dB)
8.  babble_noise_power = mean(square(babble_noise))
9.  scaled_bn = scale(babble_noise, target_noise_power / babble_noise_power)
10. noisy_signal = clean_signal + scaled_bn
11. return noisy_signal
```

Output: *noisy signal (after adding babble noise to clean signal)*

- Line 1, Line 2: Check if the lengths of *clean_signal* and *babble_noise* are not equal. If they are not equal, it means that the babble noise and clean signal have different lengths.
- Line 3: Checks if the sampling frequencies of *clean_signal* and *babble_noise* are not equal. If they are not equal, it means that the two audio signals have different sampling frequencies. If the sampling frequencies are different, the *babble_noise* is resampled using the resample function to match the length of the *clean_signal*.
- Line 4: If the sampling frequencies are the same, the *babble_noise* is trimmed to the length of the *clean_signal* using the *trim_to_len* function.
- Line 5: The *babble_noise* is normalized to match the amplitude of the *clean_signal*. The normalize function scales the *babble_noise* by the ratio of the amplitude of the *clean_signal*.
- Line 6: The *clean_signal_power* is calculated as the mean of the squared values of the *clean_signal*.
- Line 7: The *target_noise_power* is computed as the *clean_signal_power* divided by the power ratio calculated from the desired SNR (0 dB and 5 dB).
- Line 8: The *babble_noise_power* is computed as the mean of the squared values of the *babble_noise*.
- Line 9: The *babble_noise* is scaled by the ratio of *target_noise_power* to *babble_noise_power*. The scale function multiplies each sample of the *babble_noise* by the scaling factor.
- Line 10: Finally, the *scaled_bn* is added to the *clean_signal* to create the *noisy_signal*.
- Line 11: The *noisy_signal* represents the clean audio signal with the babble noise added at the desired SNR.

5.2 Metric

The system performance has evaluated using the Word Error Rate (WER) metric. The WER of the model has decreased with every increased epoch number. To calculate the WER, we have used the Levenshtein distance concept. It determines the smallest number of alteration operations required to change one string into another one (Scharenborg et al. 2017). The formula to calculate the WER is given below:

$$WER = \frac{I + D + S}{N} = \frac{I + D + S}{D + S + H} \tag{8}$$

Here, D is no. of deletions operation performed, I = no. of insertions operation used, H = total no. of hits, S = no. of substitutions operations, and N = total no. of input.

We can also calculate the WER using the "Percentage Accuracy (PA)"and "Percentage Correct (PC)" concepts where PA represents the rate of word accuracy, whereas PC represents the rate of word correction (Adhikary, R. et al. 2021).

$$WER = 100\% - PA \tag{9}$$

5.3 Result and analysis

To analysis the performance of our model in total eighteen experiments has carried out. In the given approach, before classification, a combination of extracted feature vectors is constructed. As discussed earlier, all the experiments have been carried out in three systems. There are two classification tasks in each system i.e., WER on static features and WER on delta-delta features.

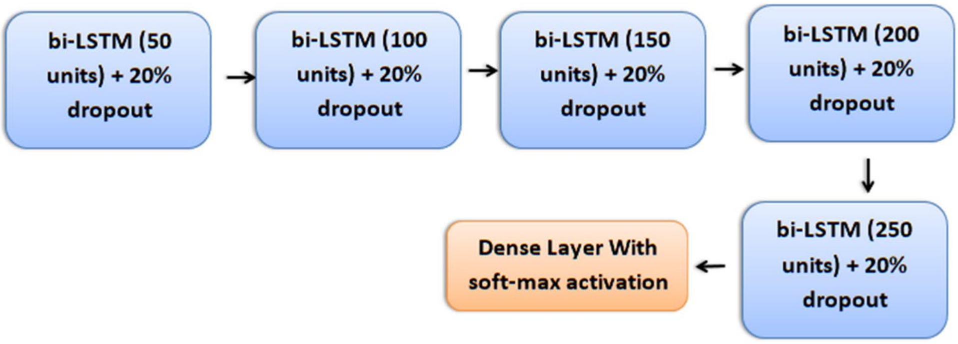
bi-LSTM (50 units) + 20% dropout
bi-LSTM (100 units) + 20% dropout
bi-LSTM (150 units) + 20% dropout
bi-LSTM (200 units) + 20% dropout
bi-LSTM (250 units) + 20% dropout
Dense Layer With soft-max activation

Fig. 3 Proposed bi-LSTM Model Architecture

5.3.1 Experiment 1: stand-alone model scenario

In the first experiment, two models the bi-LSTM and the 2D-CNN were tested on the same datasets to calculate the WER.

5.3.1.1 Performance analysis on 2D-CNN model To demonstrate the efficiency of the Cepstral coefficients with the image-based features, we have evaluated the static and dynamic features of each described system on the 2D-CNN model and tried to analyze the performance of these systems with the available back-end model. Table 2 summarizes the results achieved on the static features, whereas Table 3 summarizes the result obtained on the dynamic features.

We can observe that on taking the dynamic Cepstral coefficients in system-2, it outperforms the other cepstral features, having the 2D-CNN model as the back-end. Table 4 and Table 5 summarize the system-2 results while using the street and babble noisy dataset on static and dynamic features.

5.3.1.2 Performance analysis on bi-LSTM model Similar to the above experiment, for the evaluation of the all the system's performance we have carried out to experiments on bi-LSTM model. In these experiments we evaluated the system's performance by taking the static and dynamic features of each cepstral technique. Table 6 summarizes the results achieved on the static features Cepstral features in systems, whereas Table 7 summarizes the result obtained on applying the dynamic Cepstral features to the system.

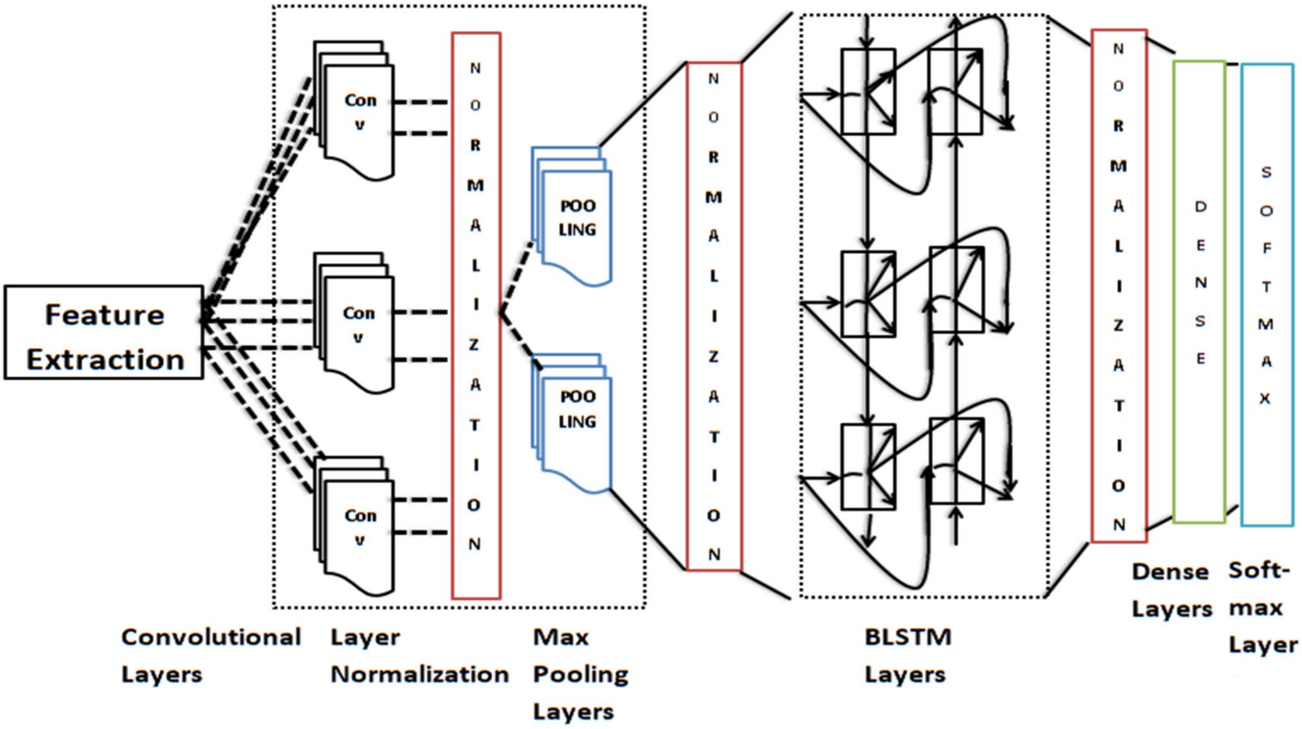

Fig. 4 Proposed Hybrid Model Architecture

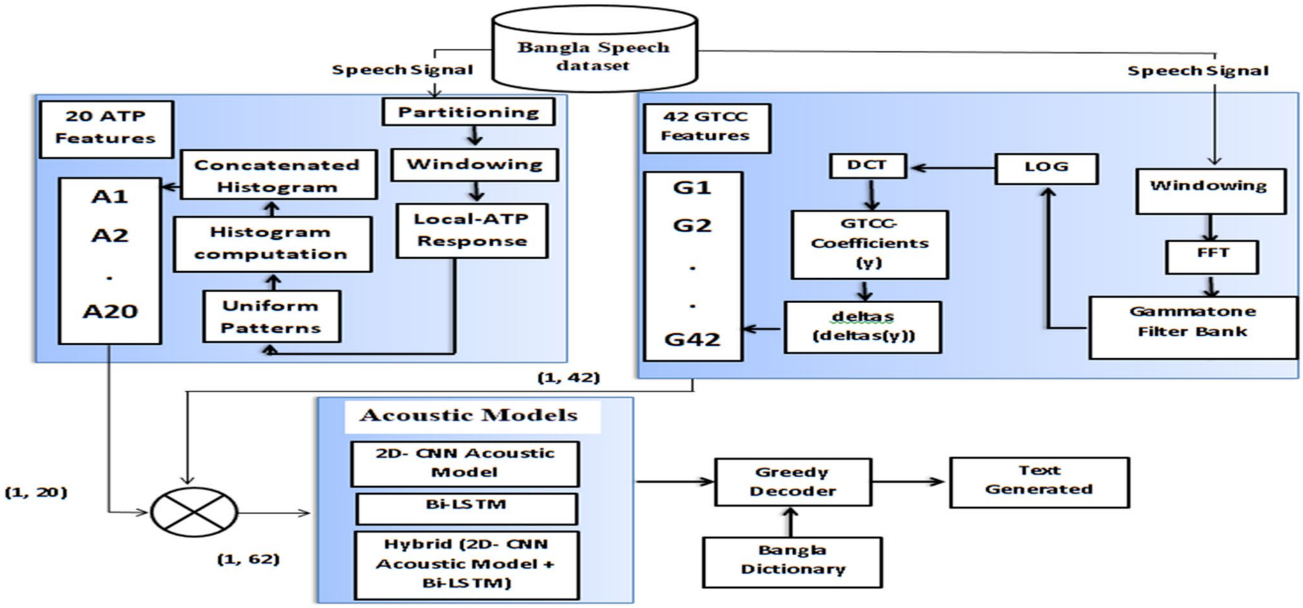

Fig. 5 Proposed Joint GTCC + ATP based ASR system (System 2)

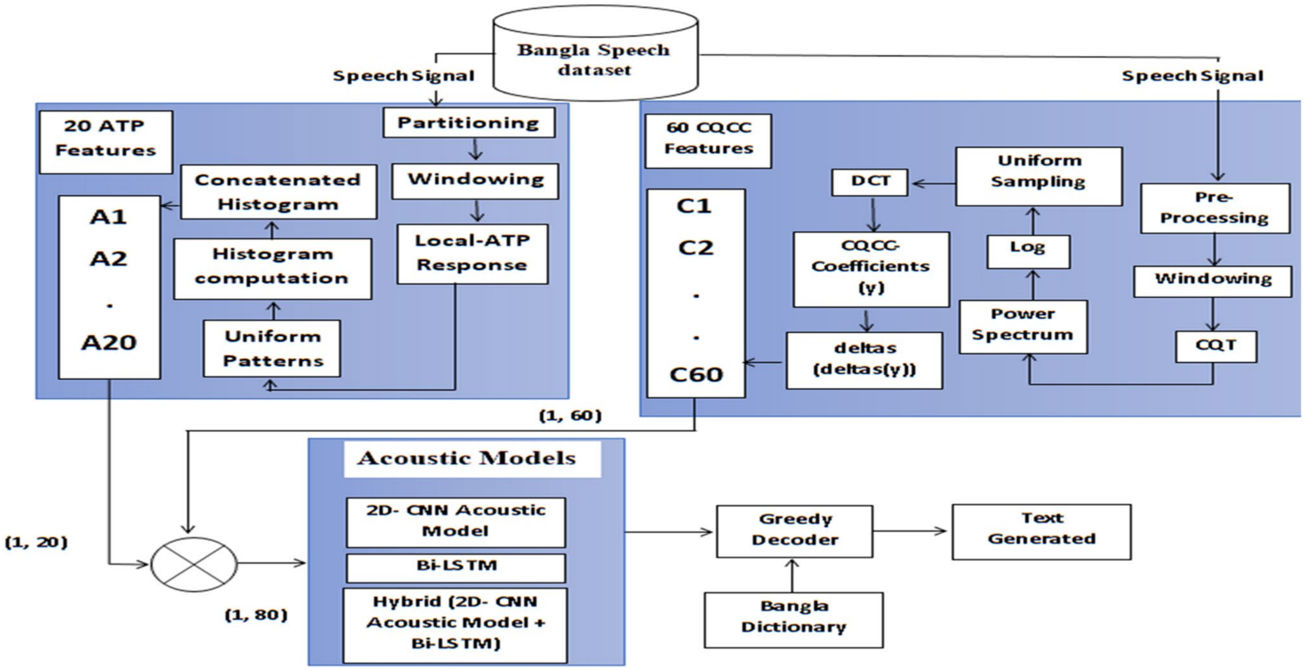

Fig. 6 Proposed CQCC + ATP based ASR system (System 3)

By observing, the Table 6 and Table 7, we can understand that the system-3 on taking the dynamic Cepstral coefficients values outperform other systems with the bi-LSTM model as the back-end. Table 8 and Table 9 summarizes the system-2 results on static and dynamic features while using the noisy dataset which is created by combing street noise and babble noise to clean signals respectively.

5.3.2 Experiment 2: hybrid model scenario

In the scenario of hybrid model, we have followed the same approach for the evaluation i.e., we have experimented the model on both static and dynamic features of the all the three systems. Table 10 summarizes the results achieved on the static features, whereas Table 11

Table 1 Characteristics of the OpenSLR' Bangla Speech dataset (Hasan et al. 2019)

Attributes	Values
Duration	250
Unique words	58,564
Total sentences	2,10,914
Unique sentences	1,07,606
Speaker	527
Female	128
Male	399

Table 2 WER for 2D-CNN model using static features

Back-end technique	WER (%)
System-1	29.26
System-3	22.01
System-2	20.59

Table 3 WER for 2D-CNN model using dynamic features

Front-end technique	WER (%)
System-1	23.28
System-3	16.43
System-2	15.73

summarizes the result obtained on the dynamic features on taking the hybrid CNN + bi-LSTM model as back-end.

By observing, the Table 10 and Table 11, we can understand that the system-3 on taking the dynamic Cepstral coefficients values outperform other systems with the hybrid model as the back-end. Table 12 summarizes the system-2 results while using the noisy dataset (new signal created by combining street noise and clean signal) on static and dynamic features. Table 13 shows the result system -2 using noisy signal created by combing babble noise to clean signal on static and dynamic features.

Table 4 WER for 2D-CNN model using multiplicative street noisy dataset on system-2

Features	5 dB %	0 dB %
Static	22.82	27.65
Dynamic	20.92	24.01

Table 5 WER for 2D-CNN model using Additive Babble Noisy dataset on system-2

Features	5 dB %	0 dB %
Static	25.04	28
Dynamic	22.3	26.02

Table 6 WER for bi-LSTM model using static features

Front-end technique	WER (%)
System-1	24.74
System-2	18.93
System-3	16.02

Table 7 WER for bi-LSTM model using dynamic features

Front-end technique	WER (%)
System-1	19.4
System-2	15.25
System-3	14.32

Table 8 WER for bi-LSTM model using Multiplicative Street Noisy dataset on system-3

Features	5 dB %	0 dB %
Static	21.03	24.42
Dynamic	18.75	23.43

5.4 Discussion

The outcomes mentioned above show that system-1 on 2D-CNN model provides the 29.26% and 23.28% WER respectively on static and dynamic Cepstral coefficients. Whereas system-2 gives the WER of 22.01% and 16.43% on the same model while taking the static and dynamic cepstral features respectively. Out of all these techniques the system-3 outperforms on the 2D-CNN model having WER of 20.59% and 15.73% on its static and dynamic features. In case of stand-alone bi-LSTM model, it was observed that the system-3 outperforms the other systems having the lowest WER of 16.02% and 14.32% on its static and dynamic CQCC features sequentially. In the scenario of hybrid model, again the system-3excels the other system's performance while producing the WER of 3.8% and 0.998% correspondingly on its static and dynamic CQCC features. From the all above experiments, it was observed that system-2 and system-3 performs closely to each other having relative difference of 4%-5% in WER. Whereas, the system-1 produces the highest WER on any of the available back-end model. Figure 7 represents the comparative analysis between all the systems on their lowest WER.

5.5 Computational cost analysis

In analyzing computational costs for three audio feature extraction methods, ATP-MFCC, ATP-GTCC, and ATP-CQCC, it is crucial to consider factors such as time complexity, memory usage, and computational requirements. The overall computational requirements depend on hardware and implementation optimizations. Memory usage is directly proportional to the number of frames and feature vector dimensionality, and temporary variables and buffers are

Table 9 WER for bi-LSTM model using additive Babble Noisy dataset on system-3

Features	5 dB %	0 dB %
Static	25	26
Dynamic	23.06	24

Table 10 WER for CNN + bi-LSTM model using static features

Front-end technique	WER (%)
System-1	15.8
System-2	7.3
System-3	3.8

Table 11 WER for CNN + bi-LSTM model using dynamic features

Front-end technique	WER (%)
System-1	9.9
System-2	1.4
System-3	0.998

Table 12 WER for CNN + bi-LSTM model using Multiplicative Street Noisy dataset on system-3

Features	5 dB %	0 dB %
Static	14.68	16.81
Dynamic	10.02	13.76

Table 13 WER for CNN + bi-LSTM model using Additive Babble Noisy dataset on system-3

Features	5 dB %	0 dB %
Static	16	18
Dynamic	12.03	13.4

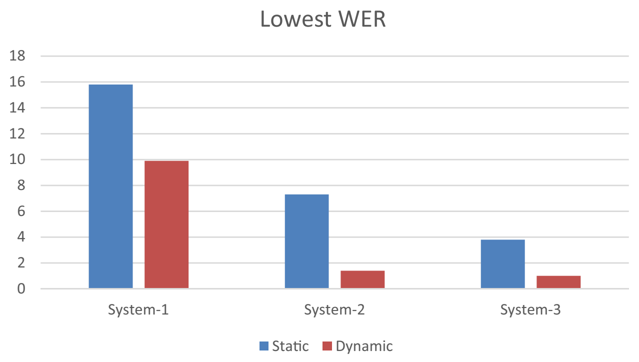

Fig. 7 Lowest WER on static and dynamic features of all the systems

needed during computation. ATP-MFCC and ATP-GTCC have time complexity of approximately *O (NF log(F))*, due to similar processing steps Chakravarty et al. (2022).

ATP-GTCC involves computationally demanding operations like filtering, envelope extraction, and DCT, which require computational resources such as floating-point arithmetic and memory access. ATP-CQCC, on the other hand, has higher computational requirements due to constant Q transform (CQT) computation and cepstral coefficient computation. The overall time complexity can vary, but is often comparable to ATP-MFCC or ATP-GTCC.

The system described in the study employs 300 epochs for training. The computational complexity is measured in terms of the average time per epoch. According to the proposed work, the hybrid 2DCNN-BiLSTM acoustic model-based system requires approximately 7 min per epoch. On the other hand, for the individual level acoustic model-based system, BiLSTM approach takes around 10 min per epoch, while the 2DCNN approach takes approximately 12 min per epoch. All these models have almost comparable time in case of training and testing.

5.6 Comparative analysis with existing approaches

In this section, we compare the suggested research work with some of the existing methodologies. Due to the technology developments, majority of the researchers have done researches in the area of the high-resource languages like Mandarin, English, and Spanish etc. As a result, there are enough amount of data available to carry-out the research work in these areas. Whereas, in case of Indian languages like Tamil, Kannada, and Bangla, etc., this cutting-edge dataset is not available. Therefore, in case of these language it is very difficult to implement state-of-the-art ASR systems.

In (Islam et al. 2021), proposed an ASR model build with using the MFCC at the front-end and LSTM model at the back-end. To train the model researchers used the Bangla-word dataset and achieved the over-all WER of 20%. In (Sen et al. 2021), proposed their work on the Bangla digits dataset. Researchers extracted the features using the MFCC technique and then employed those extracted features to the CNN model and achieved the overall accuracy of 97.1%. The dataset utilized by the researchers have been recorded in the noisy environment and in total 400 samples of both noisy and non-noisy values collected. Authors also tried to check the system's evaluation using cross-validation of tenfold and noticed the accuracy of 96.7%. (Kibria. et al. 2022), proposed their system trained on Bangladeshi Bangla dataset named as SUBAK.KO. To implement their model, they utilized an End-to-End model using RNN and CTC with an ASR algorithm. As a decoder authors implemented two approaches i.e., beam-decoder and greedy decoder. They achieved the lowest WER of 15.78% using the beam-decoder. In (Guchhait et al.2022) proposed seven different model for the comparison purpose on the Bangla digits

Table 14 Comparison between existing and proposed approach

Authors/Year	Front-end	Back-end	Dataset	WER (%)	Observation
Islam and Abujar 2021	MFCC	LSTM	Bangla-Word	20	
Sen and Roy 2021	MFCC	CNN	Bangla-Digit	2.9	The dataset was recorded in both noisy and clean environment and over-all of 400 samples were collected by Bangladeshi people
Kibria et al. 2022	Spectrogram	DeepSpeech2 (CNN + GRU)	SUBAK.KO	14.15	As the decoder, two different methods have implemented i.e., beam and greedy decoding
Guchhait et al. 2022)	MFCC	DNN + HMM + Li-GRU	Google Bangla Speech Corpus provided by OpenSlr	4.16	Make the use of pytorch and kaldi toolkits and. To explore performance implemented a Grapheme to phoneme model using RNN
Proposed System	Static & Dynamic features of MFCC, GTCC, CQCC with ATP	CNN + biLSTM	Google Bangla Speech Corpus provided by OpenSlr	0.9	Implemented six different front-end techniques and then applied them to three different models (2D-CNN, bi-LSTM, CNN + bi-LSTM) and achieved the lowest WER on CQCC $(\Delta - \Delta)$ + ATP features using hybrid model

speech corpus. They utilized the Kaldi toolkit and in one of their models' researchers used Grapheme to phoneme (G2P) module with the Recurrent Neural Network (RNN). Researchers proposed that the DNN-HMM based acoustic model with Light-Gated Recurrent Unit (Li-GRU) NN achieved the best WER of 4.16%, while using the feature extraction techniques of Kaldi. Table 14 compares our proposed system to the already implemented one.

6 Conclusion

The work in the paper proposed the combination of cepstral-based coefficients with the image-based features. The work applied these features to the CNN, bi-LSTM, and more potent hybrid acoustic model having 2D-CNN as an encoder with bi-LSTM as an acoustic model using the Connectionist Temporal Classification (CTC) loss decoder to build the state of art ASR system for Bangla Large Vocabulary Continuous Speech Recognition (LVCSR) corpus. It implemented three different systems comprising of MFCC-ATP, CQCC-ATP, and GTCC-ATP fused features. The work also tested the system with GTCC-ATP features on noisy dataset by adding the random street noise to the clean dataset at 0 dB and 5 dB SNRs. The results showed that CQCC-ATP features with the hybrid backend model outperformed all other systems by having a relative improvement of 8–10% in WER at Δ–Δ features. Also, we observed that the performance of system in noisy conditions have improved relatively by 8% to 10% with hybrid model as compared with stand-alone model. We can further extend this work by employing these robust features to other low-resource noisy datasets by using the deep conversion technique.

Declarations

Conflict of interest I, Ms. Akanksha and Ms. Shelza Dua, on the behalf of all the authors declare that: This study did not receive any finding from any resource. All the authors and the submitted manuscript do not have any conflict of interest.

Research involving human and animal participants This article does not contain any studies with human participants or animal performed by any of the authors.

References

Adhikary R, Fatema, K, Yousuf MA (2021). A Deep Learning Approach for Bangla Speech to Text Conversion. In 2021 Joint 10th International Conference on Informatics, Electronics & Vision (ICIEV) and 2021 5th International Conference on Imaging, Vision & Pattern Recognition (icIVPR) (pp. 1–8). IEEE

Ahmed M, Shill PC, Islam K, Mollah, MAS, Akhand MAH. (2015). Acoustic modeling using deep belief network for Bangla speech recognition.In 2015 18th international conference on computer and information technology (ICCIT) (pp. 306–311).IEEE

Al Amin, MA, Islam MT, Kibria S, Rahman MS. (2019). Continuous bengali speech recognition based on deep neural network. In 2019 international conference on electrical, computer and communication engineering (ECCE) (pp. 1–6).IEEE

Arafa MN, Elbarougy R, Ewees AA, Behery GM (2018) A dataset for speech recognition to support Arabic phoneme pronunciation. Int J Image Gr Signal Process 10(4):31

Aziz S, Awais M, Akram T, Khan U, Alhussein M, Aurangzeb K (2019) Automatic scene recognition through acoustic classification for behavioral robotics. Electronics 8(5):483

Bhatt S, Jain A, Dev A (2021) Feature extraction techniques with analysis of confusing words for speech recognition in the Hindi language. Wireless Pers Commun 118(4):3303–3333

Bhowmik T, Choudhury A, Mandal SKD (2017) Deep neural network based recognition and classification of bengali phonemes: a case study of bengali unconstrained speech. In: Bhattacharyya P, Sastry HG, Marriboyina V, Sharma R (eds) International conference on next generation computing technologies. Springer, Singapore, pp 750–760

Cai W, Cai D, Liu W, Li G, Li M (2017). Countermeasures for automatic speaker verification replay spoofing attack: on data augmentation, feature representation, classification and fusion. In Interspeech (pp. 17–21)

Chakravarty N, Dua M (2022) Noise robust ASV spoof detection using integrated features and Time Delay Neural Network. SN Comput Sci 4(2):127

Chakravarty N, Dua M (2023a) Data augmentation and hybrid feature amalgamation to detect audio deep fake attacks. Phys Scr 98(9):096001

Chakravarty N, Dua M (2023b) Spoof Detection using Sequentially Integrated Image and Audio Features. Int J Comput Digit Syst 13(1):1–1

Das S, Yasmin M, Arefin M, Taher KA, Uddin MN, Rahman MA (2021) Mixed bangla-english spoken digit classification using convolutional neural network. In: Kaiser S, Kasabov N, Iftekharuddin K, Zhong N (eds) In international conference on applied intelligence and informatics. Springer, Cham, pp 371–383

Dua M, Akanksha (2023) Gujarati language automatic speech recognition using integrated feature extraction and hybrid acoustic model. In: Proceedings of fourth international conference on communication, computing and electronics systems: ICCCES 2022. Springer Nature Singapore, Singapore, pp 45–54

Dua M, Kadyan V, Banthia N, Bansal A, Agarwal T (2022) Spectral warping and data augmentation for low resource language ASR system under mismatched conditions. Appl Acoust 190:108643

Ganapathiraju A, Hamaker JE, Picone J (2004) Applications of support vector machines to speech recognition. IEEE Trans Signal Process 52(8):2348–2355

Guchhait S, Hans ASA, Augustine J (2022) Automatic Speech Recognition of Bengali Using Kaldi. In: Shakya S, Ke-Lin D, Haoxiang W (eds) Proceedings of Second International Conference on Sustainable Expert Systems. Springer, Singapore, pp 153–166

Haque MA, Verma A, Alex JSR, Venkatesan N (2020) Experimental evaluation of CNN architecture for speech recognition. In: Gao XZ, Singh D (eds) In First International Conference on sustainable technologies for computational intelligence. Springer, Singapore, pp 507–514

Hasan MM, Islam MA, Kibria S, Rahman MS. (2019). Towards Lexicon-free Bangla Automatic Speech Recognition System. In 2019 International Conference on Bangla Speech and Language Processing (ICBSLP). IEEE. pp. 1–6

Hirsch HG, Pearce D. (2000). The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In ASR2000-Automatic speech recognition: challenges for the new Millenium ISCA tutorial and research workshop (ITRW).

Islam SM, Abujar S (2021) Semantics exploration for automatic bangla speech recognition. In: Bhattacharyya S, Chakrabati S, Bhattacharya A, Dutta S (eds) Emerging Technologies in Data Mining and Information Security. . Springer, Singapore, pp 171–179

Isotani R, Matsunaga S (1994) A stochastic language model for speech recognition integrating local and global constraints. in Proceedings of ICASSP'94. IEEE Int Conf Acoust Speech Signal Process 2:2–5

Izbassarova A, Duisembay A, James AP (2020) Speech recognition application using deep learning neural network. In: Learning D (ed) Classifiers with Memristive Networks. Springer, Cham, pp 69–79

Jain A, Singh VP, Rath SP (2019). A multi-accent acoustic model using mixture of experts for speech recognition. In INTERSPEECH, pp. 779–783

Javed A, Malik KM, Irtaza A, Malik H (2021) Towards protecting cyber-physical and IoT systems from single-and multi-order voice spoofing attacks. Appl Acoust 183:108283

Joshi S, Dua M (2023) Multi-order replay attack detection using enhanced feature extraction and deep learning classification. In: Mahapatra RP (ed) In Proceedings of International Conference on Recent Trends in Computing. Springer Nature Singapore, Cham, pp 739–745

Kadyan V, Mantri A, Aggarwal RK (2018) Refinement of HMM model parameters for Punjabi automatic speech recognition (PASR) system. IETE J Res 64(5):673–688

Karim R, Rahman MS, Iqbal MZ. (2002). Recognition of spoken letters in Bangla. In Proc. 5th international conference on computer and information technology (ICCIT02)

Kibria S, Rahman MS, Selim MR, Iqbal MZ (2020) acoustic analysis of the speakers' variability for regional accent-affected pronunciation in Bangladeshi Bangla: a study on Sylheti accent. IEEE Access 8:35200–35221

Kibria S, Samin AM, Kobir MH, Rahman MS, Selim MR, Iqbal MZ (2022) Bangladeshi Bangla speech corpus for automatic speech recognition research. Speech Commun 136:84–97

Kim K, Lee K, Gowda D, Park J, Kim S, Jin S, Kim C. (2019). Attention based on-device streaming speech recognition with large speech corpus. In 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU) (pp. 956–963). IEEE

Kumar A, Dua M, Choudhary T (2014) Continuous hindi speech recognition using monophone based acoustic modeling. Int J Comput Appl 24:1–5

Malik KM, Javed A, Malik H, Irtaza A (2020) A light-weight replay detection framework for voice controlled IoT devices. IEEE J Sel Top Signal Process 14(5):982–996

Mandal S, Yadav S, Rai A. (2020). End-to-End Bengali Speech Recognition. arXiv preprint arXiv:2009.09615.

Mittal A, Dua M (2021). Static–dynamic features and hybrid deep learning models based spoof detection system for ASV. Complex Intell Syst, 1–14.

Mohan BJ (2014). Speech recognition using MFCC and DTW. In 2014 international conference on advances in electrical engineering (ICAEE) (pp. 1–4).IEEE

Mori D, Ohta K, Nishimura R, Ogawa A, Kitaoka N. (2021). Advanced language model fusion method for encoder-decoder model in Japanese speech recognition. In 2021 Asia-Pacific

Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC). IEEE. (pp. 503–510)

Muhammad G, Alotaibi YA, Huda MN. (2009). Automatic speech recognition for Bangla digits.In 2009 12th International Conference on Computers and Information Technology. IEEE. pp. 379–383

Nahid MMH, Islam MA, Islam MS. (2016). A noble approach for recognizing bangla real number automatically using cmu sphinx4. In 2016 5th international conference on informatics, electronics and vision (ICIEV). IEEE. pp. 844–849

Nassif AB, Shahin I, Attili I, Azzeh M, Shaalan K (2019) Speech recognition using deep neural networks: a systematic review. IEEE Access 7:19143–19165

Oh SY, Chung K (2014) Improvement of speech detection using ERB feature extraction. Wireless Pers Commun 79(4):2439–2451

Paul AK, Das D, Kamal MM (2009). Bangla speech recognition system using LPC and ANN.In 2009 Seventh International Conference on Advances in pattern recognition IEEE pp. 171–174

Paul R, Samudravijaya K (2021) A Continuous Speech Recognition System for Bangla Language. In: Biswas A, Wennekes E, Hong TP, Wieczorkowska A (eds) Advances in Speech and Music Technology. Springer, Singapore, pp 435–447

Pujol P, Pol S, Nadeu C, Hagen A, Bourlard H (2004) Comparison and combination of features in a hybrid HMM/MLP and a HMM/GMM speech recognition system. IEEE Trans Speech Audio Process 13(1):14–22

Rademacher J, Mertins A. (2006). Auditory filterbank based frequency-warping invariant features for automatic speech recognition. Proc. ITG-FachtagungSprachkommunikation, Kiel.

Rahman M, Khatun F (2011) Development of isolated speech recognition system for bangla words. Int J Appl Res Info Tech Comp 1:272

Rahman MM, Khan MF, Moni MA (2010) Speech recognition front-end for segmenting and clustering continuous bangla speech. Daffodil Int Univ J Sci Technol 5(1):67–72

Rakib M., Hossain M, Mohammed N ,Rahman F (2022). Bangla-Wave: Improving Bangla Automatic Speech Recognition Utilizing N-gram Language Models. arXiv preprint arXiv:2209.12650.

Renals S, Morgan N, Bourlard H, Cohen M, Franco H (1994) Connectionist probability estimators in HMM speech recognition. IEEE Trans Speech Audio Process 2(1):161–174

Samin AM, Kobir MH, Kibria S, Rahman MS (2021) Deep learning based large vocabulary continuous speech recognition of an under-resourced language Bangladeshi Bangla. Acoust Sci Technol 42(5):252–260

Saranya MS, Padmanabhan R, Murthy HA (2018). Replay attack detection in speaker verification using non-voiced segments and decision level feature switching. In 2018 international conference on signal processing and communications (SPCOM) IEEE. pp. 332–336

Scharenborg O, Ciannella F, Palaskar S, Black A, Metze F, Ondel L, Hasegawa-Johnson M. (2017). Building an ASR system for a low-research language through the adaptation of a high-resource language ASR system: preliminary results.In Proc. Internat. Conference on Natural Language, Signal and Speech Processing (ICNLSSP) (pp. 26–30)

Scheidl H, Fiel S, Sablatnig R. (2018). Word beam search: A connectionist temporal classification decoding algorithm. In 2018 16th International Conference on Frontiers in Handwriting Recognition (ICFHR) (pp. 253–258).IEEE

Sen O, Roy P. (2021). A convolutional neural network based approach to recognize bangla spoken digits from speech signal. In 2021 International Conference on Electronics, Communications and Information Technology (ICECIT) (pp. 1–4).IEEE

Shao Y, Jin Z, Wang D, Srinivasan S (2009). An auditory-based feature for robust speech recognition.In 2009 IEEE International Conference on Acoustics, Speech and Signal Processing (pp. 4625–4628).IEEE

Showrav TT. (2022). An Automatic Speech Recognition System for Bengali Language based on Wav2Vec2 and Transfer Learning. arXiv preprint arXiv:2209.08119

Vergin R, O'Shaughnessy D, Farhat A (1999) Generalized mel frequency cepstral coefficients for large-vocabulary speaker-independent continuous-speech recognition. IEEE Trans Speech Audio Process 7(5):525–532

Wang D, Wang X, Lv S (2019) An overview of end-to-end automatic speech recognition. Symmetry 11(8):1018

Wang X, Xiao Y, Zhu X (2017). Feature Selection Based on CQCCs for Automatic Speaker Verification Spoofing. In Interspeech (pp. 32–36)

Xiao B, Wang K, Bi X, Li W, Han J (2018) 2D-LBP: an enhanced local binary feature for texture image classification. IEEE Trans Circuits Syst Video Technol 29(9):2796–2808

Yang Y, Wang P, Wang D (2022). A conformer based acoustic model for robust automatic speech recognition. arXiv preprint arXiv:2203.00725.

Yu H, Tan ZH, Ma Z, Martin R, Guo J (2017) Spoofing detection in automatic speaker verification systems using DNN classifiers and dynamic acoustic features. IEEE Trans Neural Netw Learn Syst 29(10):4633–4644

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