IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received November 25, 2021, accepted December 13, 2021, date of publication December 16, 2021,
date of current version January 4, 2022.

Digital Object Identifier 10.1109/ACCESS.2021.3136251

Bangla Speech Emotion Recognition and Cross-Lingual Study Using Deep CNN and BLSTM Networks

SADIA SULTANA[1], M. ZAFAR IQBAL[1], M. REZA SELIM[1], MD. MIJANUR RASHID[2], AND M. SHAHIDUR RAHMAN[1], (Senior Member, IEEE)

[1]Department of Computer Science and Engineering, Shahjalal University of Science and Technology, Sylhet 3114, Bangladesh
[2]REPL Group, Accenture, Henley-in-Arden B95 5QR, U.K.

Corresponding author: M. Shahidur Rahman (rahmanms@sust.edu)

ABSTRACT In this study, we have presented a deep learning-based implementation for speech emotion recognition (SER). The system combines a deep convolutional neural network (DCNN) and a bidirectional long-short term memory (BLSTM) network with a time-distributed flatten (TDF) layer. The proposed model has been applied for the recently built audio-only Bangla emotional speech corpus SUBESCO. A series of experiments were carried out to analyze all the models discussed in this paper for baseline, cross-lingual, and multilingual training-testing setups. The experimental results reveal that the model with a TDF layer achieves better performance compared with other state-of-the-art CNN-based SER models which can work on both temporal and sequential representation of emotions. For the cross-lingual experiments, cross-corpus training, multi-corpus training, and transfer learning were employed for the Bangla and English languages using the SUBESCO and RAVDESS datasets. The proposed model has attained a state-of-the-art perceptual efficiency achieving weighted accuracies (WAs) of 86.9%, and 82.7% for the SUBESCO and RAVDESS datasets, respectively.

INDEX TERMS Bangla SER, deep CNN, RAVDESS, SUBESCO, time-distributed flatten.

I. INTRODUCTION

Identifying human emotions from voice signals, using a machine learning approach, is important to construct a natural-like human-computer interaction (HCI) system. Robotics, mobile services, contact centers, computer games, and psychological examinations are just a few of the examples where speech emotion recognition (SER) is used. Research on the development of a successful SER system is emerging in recent years, though it has been in action since the last two decades [1]. The SER system as a whole, is a collection of methodologies for analyzing and classifying speech data in order to discover the embedded emotions. The first step in developing it is to create a dataset that is appropriate for the target language and modality. The emotional database can be acted, simulated, or elicited for audio-only, audio-visual or facial expressions [2]. However, selecting the appropriate features for classifying emotions accurately is the most crucial design decision. Acoustic features of speech are considered the most essential and extensively used features for speech emotion representation. Different kinds of acoustic features such as prosodic, spectral, voice quality, energy operator, etc. have been employed to construct SER in various studies [3]. Those features can be further classified as temporal (time-domain) and spectral (frequency-domain) features. It is also important to understand how emotions are represented in discrete or dimensional emotional models to explain the functions of similar emotions [4]. In the discrete approach, emotions are represented as different emotional states e.g. sadness, happiness. Emotions are represented in the dimensional model by the levels of positive to negative arousal and low to high valence. The ultimate result of SER is obtained by the use of a classifier, which allows the system to determine the best match for input emotional speech. Selecting an efficient classifier is a crucial part of the SER. As a result, numerous types of classifiers have emerged to date, and the research is still ongoing. Hidden Markov Model (HMM), Support Vector Machines (SVM), Gaussian Mixture

The associate editor coordinating the review of this manuscript and approving it for publication was Wei Jiang.

Model (GMM), Artificial Neural Network (ANN), decision trees, and ensemble approaches are some well-known classifiers that have been employed in previous studies. The recent tendency is to use deep learning-based classifiers like CNN, DNN, and RNN, as well as deep learning-based augmentation techniques like auto-encoders, multitask learning, attention mechanism, transfer learning, adversarial training, etc [3]. In this study, two datasets from distinct languages were investigated. The first one is SUST Bangla Emotional speech corpus (SUBESCO) which has recently been developed and made publicly available for the Bangla language [48]. The second corpus is the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS), which was created for the American English language [49]. Log mel-spectrogram has been used for input feature vector to the CNN layer. Experiments demonstrate that mel-spectrogram shows better performance as an effective audio feature than others like MFCCs, STFTs [5]–[7]. In recent studies, a CNN-LSTM combination has been frequently used to construct an end-to-end SER system as it gives promising outcomes for the spectral-temporal features [8]. CNNs have wide use in the field of computer vision as well as in speech-related researches [9]. Deep CNN has the powerful ability to learn from a large number of samples and represent a higher-level task-specific knowledge. It is also important to capture the sequential information of speech for emotion recognition. Long short-term memory of recurrent neural network architecture (LSTM-RNN) can exploit this information [10]. A BLSTM layer with a DCNN block has been utilized in this study for effective features extraction to classify emotions. Transfer learning with deep learning models is comparatively a new and efficient tool for cross-lingual research [11], which has also been reported in this paper.

The primary contributions of this study are as follows: i) This work examined a deep learning-based model for the low-resource language Bangla, which achieved a high perception accuracy of 86.86% utilizing the largest emotional speech corpus SUBESCO available for this language. ii) A novel architecture DCTFB (deep CNN with Time-distributed flatten and BLSTM layers) has been proposed consisting of a feature learning block DCNN, a time-distributed flattening layer and a BLSTM layer that can acquire both local and sequential information of the emotional speech. iii) A comparative analysis is presented to show that the proposed architecture effectively enhances emotion detection in comparisons with other similar models. The model obtained state-of-the-art performances for SUBESCO and RAVDESS datasets. iv) A detailed cross-lingual study experimenting cross-corpus training, multi-corpus training along with a transfer learning technique using Bangla and English emotional audio corpora has been presented here for the first time.

The rest of this paper is organized as follows: section II highlights related works done in recent years, while section III focuses the methodology, including preprocessing, spectrogram generation, architectures, and training methods. Experimental details are presented in section IV. Sections V and VI explain the findings and discussions. Finally, a conclusion is drawn in section VII.

II. RELATED WORKS

Though there have been numerous studies in the field of SER for other languages, particularly English, just a few attempts have been made to establish SER for Bangla. In 2018, Rahman *et al.* proposed a Dynamic time warping assisted SVM emotion classifier for Bangla words [12]. The first and second derivatives of MFCC features were extracted as features for classification. The system achieved 86.08% average accuracy for a small dataset of only 200 words. In 2017, Badshah *et al.* proposed a CNN architecture consisting of three convolutional layers and three FC layers [13]. The model was trained on Berlin emotional corpus to discriminate between seven emotions based on the spectrograms collected from the stimuli. The average prediction accuracy for this system was 56%. Satt *et al.* presented an SER model which calculates log-spectrograms as feature vectors [14]. They experimented with two architectures: convolution-only and convolution-LSTM deep neural networks achieving prediction rates of 66% and 68%, respectively, for the IEMOCAP dataset. Etienne *et al.* employed a CNN-LSTM architecture to classify emotions using spectrogram information in 2018 [15]. They trained the model using the improvised part of the IEMOCAP dataset and got a WA of 64.5%. Three scenarios were considered for their experiment: shallow CNN with deep BLSTM, deep CNN with shallow BLSTM, and deep CNN with deep BLSTM. They achieved the best result for the combination of 4 convolutional and 1 BLSTM layer. With 3-D attention-based convolutional recurrent neural networks (ACRNN), Chen *et al.* used deltas and delta deltas of log mel-spectrogram for emotion identification [16]. The model was trained on Emo-DB corpus and improvised data of IEMOCAP corpus, yielding recognition accuracies of 82.82% and 64.74% for the two databases, respectively. The researchers trialed combinations of different numbers of convolution layers with LSTM. Among them, the combination of 6 convolutional layers with LSTM performed the best. Zhao *et al.* [17] have presented another CNN-LSTM based deep learning model for end-to-end SER. The model was trained and tested using spectrograms taken from the audios of the IEMOCAP dataset, yielding a WA perception accuracy of 68%. The model was composed of attention-based BLSTM layers to extract the sequential features and fully convolutional network (FCN) layers to learn the spectro-temporal locality of the spectrograms. Another FCN model incorporating an attention mechanism was evaluated on the IEMOCAP corpus in 2019 and claimed to outperform state-of-the-art models with an WA of 63.9% [18]. 2D CNN based architecture was employed for feature extraction from audios and SVM was used for emotion classification. Ghosal *et al.* [19] proposed a graph neural network-based technique for emotion recognition in conversation called dialogue graph convolutional network (DialogueGCN). They compared the

performance of the architecture with baseline CNN models and others for the three datasets IEMOCAP, AVEC, and MELD. Perceived weighted accuracy for the IEMOCAP dataset was 64.18%. Zhao *et al.* [20] used a connectionist temporal classification (CTC) with attention-based BLSTM for SER. The system outperformed existing systems with a 69% accuracy for the IEMOCAP dataset in 2019. They extracted log mel-spectrogram for each audio to perform the classification task. Another model, composed of a combination of BLSTM with FCN, reported a weighted accuracy of 68.1% for the IEMOCAP dataset and an unweighted accuracy of 45.4% for the FAU-AEC dataset [21]. Mel-spectrograms were fed into attention-based FCNs to classify emotions using LSTM-RNNs. Mustaqeem and Kwon proposed a state-of-the-art SER model using a deep stride CNN (DSCNN) with special strides in 2020 [22]. Spectrogram features were extracted from clean speech to classify emotions from the datasets IEMOCAP and RAVDESS. The system reported average accuracies of 81.75% and 79.5% for IEMOCAP and RAVDESS datasets, respectively. An edge and cloud-based emotion recognition system using the Internet of Things (IoT) was proposed in [23], where deep-learned features were extracted using CNN in the core cloud. The system achieved unweighted accuracies of 82.3% and 87.6% for the RML database and eNTERFACE'05 database, respectively. The same authors presented another deep learning based emotion recognition system for Big Data containing both speech and video [24]. A recent study based on dilated causal convolution with context stacking for end-to-end SER was proposed by Tang *et al.* [25]. The proposed architecture consists of dilated causal convolution blocks that are stacked with various dilation numbers. The stacked structure consists of three learnable sub-networks and uses local conditioning related to input frame for end-to-end SER. Experimented datasets were RECOLA and IEMOCAP, and the extracted feature was log-mel spectrogram. For improvised utterances from the IEMOCAP dataset, the system obtained a WA of 64.1%. For the RECOLA dataset, this design increased the WA by 10.7%.

Our research differs from the previous works mentioned above as it proposes a new architecture experimented on the RAVDESS and SUBESCO datasets showing state-of-the-art performances.

III. METHODOLOGY

A. DATA PREPROCESSING

The **librosa** [26] framework was used to read and re-sample each wav file at a sampling rate of 44KHz. Silence was removed by trimming the signal under 25dB. Both the datasets were created in studio environments and have a minimum amount of noise in stimuli. To remove additive noise from the audios, a **Wiener** filter was utilized. This filter is a linear minimum mean square error estimator (LMMSEE) and it performs very well with less speech distortion for single-channel audios [27]. It estimates the desired signal $y(n)$ from the observed signal $x(n)$ using the optimal filter coefficients w^* assuming that the desired signal and noise $v(n)$ are uncorrelated.

$$x(n) = y(n) + v(n) \tag{1}$$

The signal obtained after filtering is:

$$\hat{y}(n) = w^{*T} x(n) \tag{2}$$

Before creating the mel-spectrograms, all audios were limited to 3 seconds in length. Fixing the length does not trim any important information because the stimuli in the RAVDESS dataset are already 3 seconds long, and almost all of the stimuli in SUBESCO terminate in 3 seconds, if we exclude the silence at the end of each recording.

B. MEL-SPECTROGRAM GENERATION

Humans have a logarithmic perception of auditory frequencies. Mel-scale represents signal frequencies in the logarithmic scale, which is similar to this notion. Spectrogram visually represents how the frequencies evolve over time. This time-frequency representation of a signal is very important for some experiments where the time alone or the frequency domain descriptions are not enough to provide comprehensive information for classification [28]. Mel-spectrogram represents the power spectrogram for each mel against time, and it can illustrate the relative importance of different frequency bands similar to the way of human ear perception. The relationship between the mel spectrum frequency f_{mel} and the signal frequency fHz is defined as:

$$f_{mel} = 2595 \cdot log_{10}\left(1 + \frac{f}{700}\right) \tag{3}$$

In this study, mel-spectrogram for each sentence was calculated for 128 mel filter banks using a function of the **librosa** framework. The power log mel-spectrogram was extracted by converting the magnitude of the spectrogram in logarithmic scale decibel. Figure 1 illustrates examples of extracted mel-spectrograms for four different emotional audios: anger, happiness, neutral, and sadness, all spoken by the same male speaker for the same sentence. It is evident, the four spectrograms in this figure differ from one another. The time is displayed by the x-axis, while the converted log mel-scale is represented by the y-axis (frequency). The color dimension represents the magnitude of the decomposed frequency components of the signal, corresponding to the mel-scale. Dark colors indicate low amplitudes, while stronger amplitudes are denoted by brighter colors.

C. CONVOLUTIONAL NEURAL NETWORK (CNN)

The convolutional neural network is a special kind of artificial neural network that can learn special hierarchical features adaptively [29]. Convolutional layers, pooling layers, and fully linked layers are the fundamental building components of a CNN. Convolution layers use arrays of numbers, called kernels, to transform input data into feature maps. Before beginning the training procedure, two hyperparameters, the **size** and the **number of kernels** are defined. The parameter **stride** indicates the amount of kernel's movement on

the input matrix. **Padding** is applied to the input matrix to allow kernels to overlap the outermost elements. The **pooling layer** down-sizes the feature maps by subsampling them and retaining only the dominant information. Common pooling methods are: max pooling, min pooling, global average pooling, etc. An **activation function** determines whether or not to fire a neuron based on the preference of mapping of input to the desired output. Sigmoid, tanh, ReLU are commonly used activation functions. This function is employed after all non-linear convolutional layers and fully-connected layers. **Dropout** is a regularization technique that avoids overfitting by ignoring some randomly selected neurons. It can be used on neurons in the input layer as well as hidden layers. The output feature maps of the last convolution layer are the input to the fully connected layer (FC). The FC layer connects all of its neurons to all neurons of the previous layer, and it flattens the input into a one-dimensional array of numbers. Finally, to complete the classification task, an activation function is used. CNN has the advantage of reducing the number of network parameters in training by allowing weight sharing. Concurrent learning of feature extraction in this network makes it highly organized and easier to implement than other networks [30].

D. LONG SHORT-TERM MEMORY (LSTM)

Long Short-Term Memory (LSTM) is a kind of RNN that is composed of recurrently connected memory blocks that contain memory cells with self-connections to store the temporal states of the network. In memory blocks, there are special multiplicative units, called **gates**, that control the flow of information [31]. There are three gate units in each memory block: input, output, and forget gates. The input gate multiplies the cell input by the activation function to perform read, the cell output is multiplied by the activation of the output gate to perform write. And, to perform reset, the activation of the forget gate is multiplied by the previous cell values [32]. LSTM solves the problem of long-term dependence in the RNN, and it is more capable in implementing a refined internal processing unit to effectively store and update context information [33]. It also overcomes the standard RNN's problem of gradient vanishing or exploding during training [34].

The traditional LSTM can only learn in one way, however, the bidirectional-LSTM can access context information in both forward and backward directions. It can make the system more robust by recognizing the concealed emotions through directional analysis [35]. For forward analysis, the received signal sequence is fed in its original order into one LSTM cell in the forward direction generating the sequence of hidden states as $fh_{kt} = \{fh_{k1}, \ldots fh_{kT}\}$. For backward analysis, the signal sequence is fed in reverse order into another LSTM cell in the backward direction generating the sequence of hidden states as $bh_{kt} = \{bh_{kT}, \ldots bh_{k1}\}$. As the last states of those sequences contain the information of the entire sequences, those are concatenated together to get the final state h_t at each time point t.

$$h_t = [fh_{kT}, bh_{k1}] \tag{4}$$

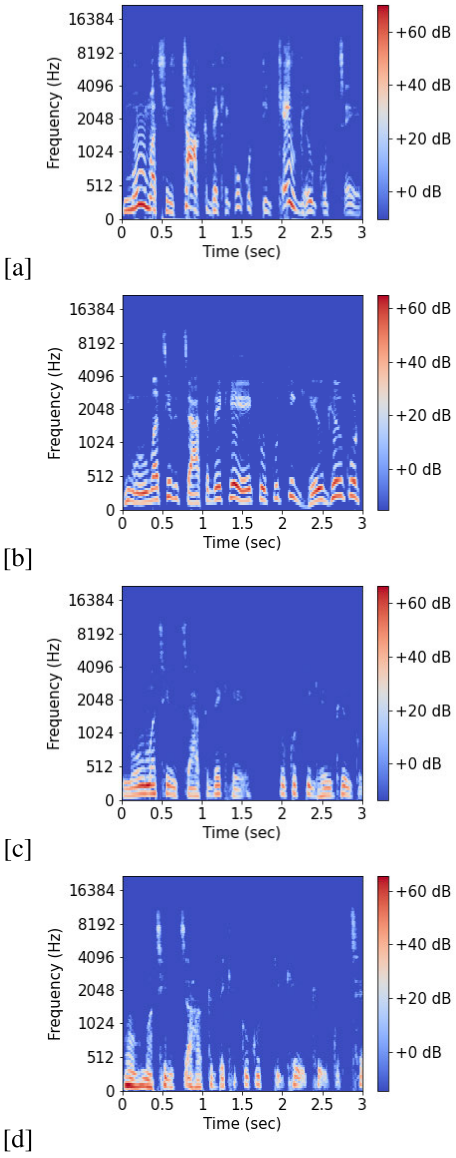

FIGURE 1. Example mel-spectrograms for emotions (a) Anger (b) Happiness (c) Neutral (d) Sadness.

E. PROPOSED DEEP CNN WITH TIME-DISTRIBUTED FLATTEN LAYER AND BLSTM LAYER (DCTFB) ARCHITECTURE

The proposed SER system (Figure 2) utilizes log mel-spectrograms extracted from the speech signals. The spectrograms are fed into the DCNN as input of size 128×259. The DCNN architecture consists of four local convolutional blocks similar to the local feature learning blocks (LFLB) described in the study [36]. But, the number of kernels and layer parameters of this model are different from those of the reference model. The 2D convolutional layer in each block is followed by a batch normalization layer, an exponential linear unit (ELU) activation, and a 2D max-pooling layer. The result

of 2D convolution $z(i, j)$ is obtained by convolving the input signal $x(i, j)$ with the kernel $w(i, j)$ of size k.

$$z(i,j) = \sum_{u=-k}^{k} \sum_{v=-k}^{k} x(u,v) \cdot w(i-u, j-v) \tag{5}$$

The batch normalization (BN) layer normalizes the activation of the convolutional layer by taking its learned features as input by mini batch. It acts as a regularizer and it accelerates the training process by reducing the internal covariate shift [37]. ELU acts as an activation function and defines the output of the BN layer. The advantages of using ELU are: it has a lower computation complexity, it can speed up the learning for a reduced bias shift effect, and it performs better with batch normalization than other activation functions [38]. ELU activation function is defined as:

$$f(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha(e^x - 1), & \text{if } x \leq 0 \end{cases} \tag{6}$$

where, $\alpha > 0$.

The max-pooling layer employs a widely used maximum pooling function that extracts the largest value from each patch of the activated convoluted feature map. It is used to prevent over-fitting and to down-sample the output features to reduce computational load. Figure 3 shows details of the suggested DCNN architecture. The first convolutional layer has 128 kernels of size 3×3 and stride of 1×1. In the first convolutional block, the max-pooling layer has kernels of 2×2 size and 2×2 stride. Each of the convolution layers in each block has the same kernel size and stride. But the numbers of filters are different for the blocks. There are 128 filters in the first two blocks and 64 filters in the latter two blocks. In the last three blocks, the max-pooling layers have a kernel of size 4×4 and a stride of 4×4. The suggested deep neural network model's layer parameters are detailed in Table 1. The output dimension indicates the height $\times$ width $\times$ filter_number in each layer. For the input of $x_1 \times x_2$ with zero padding of 1 and k kernels of stride 1 in the convolutional layer, the output feature map is $x_1 \times x_2 \times k$. In the max-pooling layer for stride of 2×2, the output feature map after pooling is $x_1/2 \times x_2/2 \times k$. The output of the DCNN block's final CNN layer is fed into a time-distributed flatten (TDF) layer which is enclosed in a time-distributed wrapper. This wrapper allows the application of the same weights and biases to each temporal time steps of a layer which is important to exploit the temporal correlations of sequential input data. This also helps LSTM to analyze sequential information obtained from the previous layer with a timing arrangement. If the input shape is $x_1 \times x_2 \times x_3$, flatten converts it into $1 \times (x_1 \cdot x_2 \cdot x_3)$ form. The output of this layer is passed into a BLSTM layer to capture both the past and future information for analysis. The output dimension of the BLSTM network is twice that of the LSTM network's hidden units. To avoid over-fitting, the BLSTM layer is followed by a 25% neuron dropout. Finally, in the fully connected (dense) layer, the softmax activation function was employed to normalize the prediction of emotion classification. This activation is a simple and effective function to assess and discriminate the features for prediction [39]. A classical softmax function for every component i in a j dimension input vector z is defined as:

$$softmax_i(z) = \frac{e^{z_i}}{\sum_j e^{z_j}} \quad \text{for } i = 1, \ldots, j \tag{7}$$

TABLE 1. Layer parameters for proposed DCTFB architecture.

Layer	Output Shape	Kernel Size	Stride
Convolution 1	128x259x128	3x3	1x1
Max-pooling 1	64x129x128	2x2	2x2
Convolution 2	64x129x128	3x3	1x1
Max-pooling 2	16x32x128	4x4	4x4
Convolution 3	16x32x64	3x3	1x1
Max-pooling 3	4x8x64	4x4	4x4
Convolution 4	4x8x64	3x3	1x1
Max-pooling 4	1x2x64	4x4	4x4
TimeDistributed (Flatten)	1x128		
Bi-LSTM	512		
Dropout (0.25)	512		
Dense + softmax	7		

1) COMPUTATIONAL COMPLEXITY

There are four convolutional blocks and a BLSTM layer in the proposed DCTFB model. Given, d is the total number of convolutional layers, the computational complexity of multiple convolutional layers is expressed as [40]:

$$\mathcal{O}(\sum_{l=1}^{d} m_l^2 \cdot k_l^2 \cdot n_l \cdot o_l) \tag{8}$$

where, for each convolutional layer l, m_l is the spatial size of the output feature map, k_l is the size of the kernel, n_l is the number of input channels, o_l is the number of output channels. As batch normalization and pooling layers take very little time, we ignore them while calculating the computational complexity of the model. Moreover, both of them speed up the overall learning process. The computational complexity of BLSTM network is $\mathcal{O}(w)$; where, w is the number of input parameters for the layer [41]. For i iterations and e epochs the overall computational complexity of the learning process is:

$$\mathcal{O}((\sum_{l=1}^{d}(m_l^2 \cdot k_l^2 \cdot n_l \cdot o_l) + w) \cdot i \cdot e) \tag{9}$$

It denotes that the suggested architecture has big **O** complexity in the asymptotic notation. Apart from the theoretical computational complexity, the execution time of a deep learning model depends on the the implementation details and the hardware configuration. Each epoch training time for this architecture is $\approx$ 20s for SUBESCO training and $\approx$ 6s for RAVDESS training using Google Colaboratory with GPU setup. The total number of learning parameters in this model is around 1.05 million. There are 768 non-trainable parameters. The number of training parameters per layer is presented in Table 2. The number of training parameters P_c

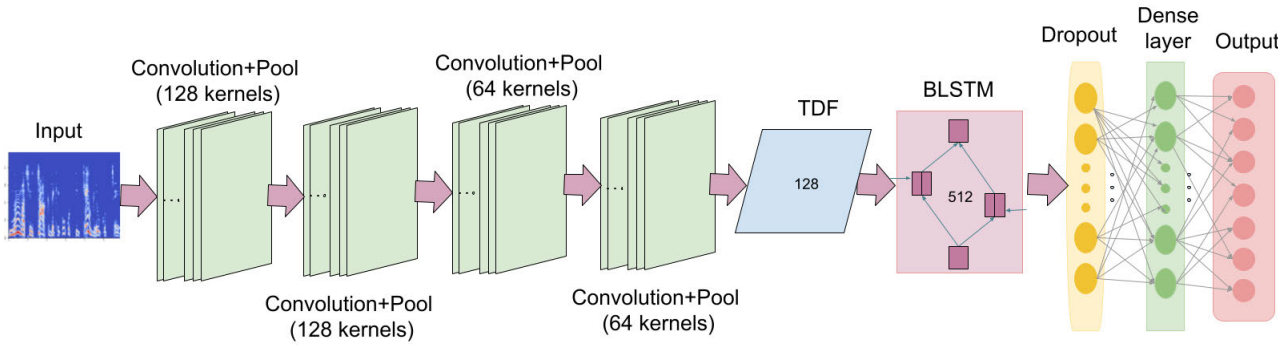

FIGURE 2. Architecture of the proposed SER system.

for a convolutional layer with u kernels of size s, v input channels, and b biases is:

$$P_c = s^2 \cdot u \cdot v + b \tag{10}$$

For each input vector, batch normalization calculates four parameters: gamma weight, beta weight, optimal mean, and standard deviation. As a result, the total number of parameters P_{bn} for N input features is:

$$P_{bn} = 4 \cdot N \tag{11}$$

The LSTM layer calculates the parameters for 3 gates (input gate, output gate, and forget gate) and a cell state. The total number of parameters P_l for this layer is calculated as:

$$P_l = 4 \cdot ((a + 1) \cdot c + c^2) \tag{12}$$

where, $a =$ input size, and $c =$ output size. BLSTM has twice the number of parameters for that of the LSTM layer. Each neuron from the previous layer is connected to each neuron in the current layer by the dense layer. If the previous layer neuron number is p_n, the current layer neuron number is c_n, and the bias is b, then the number of parameters P_d of this layer is calculated as:

$$P_d = p_n \cdot c_n + b \tag{13}$$

As the activation, max-pooling, dropout, and flatten layers do not involve back-propagation learning, the number of learning parameters is 0 for those layers.

F. OTHER NEURAL NETWORK MODELS

In addition to the proposed DCTFB model described above, we also experimented with eight other models. Among them, there are three reference models which were proposed recently for SER. The other five models are our experimented models to observe the impact of applying TDF with LSTM and BLSTM layers. The networks are briefly described below:

1) 2 CNN BLOCK ARCHITECTURE

The first model we considered as the base model for our study contained two 2D CNN and max-pooling pairs with two fully-connected layers which were presented in [42].

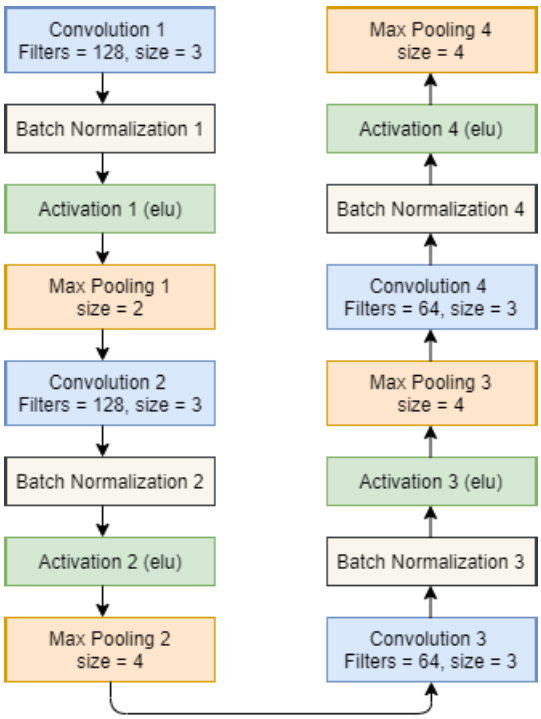

FIGURE 3. Proposed DCNN architecture.

There was a RELU activation after each convolutional layer. The first convolutional block used 128 kernels, while the second block used 64 kernels. Kernel size is 5 × 5 with stride 1 in each layer. The max-pooling layers had kernels of size 2 with stride 2. Batch normalization was added with each convolutional layer, though it was not present in the original model which improved the performance of the system. The last max-pooling layer is followed by an 85% dropout. This system contains two fully connected layers with a Softmax classifier. A Stochastic gradient descent algorithm was applied for model optimization. This model is named RM1 (reference model 1) in this study.

2) 4 CNN BLOCK ARCHITECTURES

To compare the results of four block architectures we experimented with the reference model described in [36], which is referred to as RM2 (reference model 2) in this study.

TABLE 2. Training parameters for proposed DCTFB architecture.

Layer	Kernels/ Units	Parameters
CNN1	128	1280
BN1	128	512
CNN2	128	147584
BN2	128	512
CNN3	64	73792
BN3	64	256
CNN4	64	36928
BN4	64	256
BLSTM	512	788480
Dense	7	3591
Total	-	1,053,191

The first two convolutional layers of RM2 have 64 kernels, whereas the last two have 128 kernels. Each convolutional layer is followed by a BN, an ELU activation, and a max-pooling layer. Kernel size is 3 (stride 1) for the convolutional layer and 2 (stride 1) for the max-pooling layer in each block. We experimented with three other variations of the proposed 4 CNN block DCTFB model. In the 4CNN+LSTM and 4CNN+BLSTM models, reshaping was employed instead of a TDF layer after the last convolutional blocks. For all of these models, the training parameters were the same as the proposed system.

3) 7 CNN BLOCK ARCHITECTURES

The base model for seven layer architecture was a deep stride CNN architecture which was described in [22]. This model was tested on 1440 RAVDESS audio-only speech files and yielded an accuracy of 79.5% for clean speech data. This model is referred to as RM3 (reference model 3) in this study. Seven convolutional layers, each with a batch normalization layer and ELU activation, make up the models. There is no max-pooling layer. The numbers of filters used in convolutional layers are 16, 32, 32, 64, 64, 128, 128. Kernel sizes are 7×7 for the first layer, 5×5 for the second layer, and 3×3 for the remaining convolutional layers. All of these kernels had a stride of 2. The remaining parts had a 25% dropout followed by an FC layer with a softmax activation function. The final FC layer is followed by a Softmax activation. We experimented with two variants of the RM3 system. Those variants had a TDF layer after the last convolutional layer. In one case, it was followed by an LSTM layer, whereas in the other, it was followed by a BLSTM layer.

G. TRAINING THE MODELS

The **Keras** [43] neural network package was used to implement all of the models for comparative analysis. The 'same' padding approach from Keras was utilized in each convolutional layer, implying that the output spatial dimensions for stride 1 are the same as the input spatial dimensions. To train the models, we used log-mel spectrograms derived from the audios in our target dataset as input features. The entire set of input characteristics was divided into 70%, 20%, and 10% ratios, for training, validation, and testing, respectively. To get the optimal results, it is very important to define a proper fitness for training a DNN. We trialed different learning rates and other parameters for all of the models detailed above and found that a learning rate of 10^{-3} with a decay of 10^{-6} for a batch size of 32 produced the best results. Batch normalization and dropout were utilized for regularization. The loss function used to measure the prediction error of the DNN model was cross-entropy loss [44]. A gradient-based optimization algorithm **Adam** [45] was used during training of the models and the softmax activation function was applied for classification. Adam is computationally efficient in dealing with gradients and also suitable for training with large number of parameters with less memory requirement. For our experiments, all of the models were trained and executed using Jupyter notebook [46] both on a local workstation without a GPU and on Google Colaboratory [47] which is a cloud-based service with a GPU. The local machine was a MacBook Pro notebook with a 2.4 Intel core i9 processor, 32 GB RAM, and Intel UHD Graphics 630, running OS version 10.15.7.

IV. EXPERIMENTAL SETUPS

A. DATASETS

SUBESCO and RAVDESS are the two datasets used in this research study. SUBESCO is the only verified emotional speech corpus for Bangla that is gender-balanced [48]. It is an acted emotional corpus with 7000 audios from ten male and ten female speakers. In this dataset, there are seven acted emotions: anger, disgust, fear, happiness, neutral, sadness, and surprise. The statistics of all the recognizers that were run on SUBESCO were compared to those from the RAVDESS dataset [49].

RAVDESS is an audio-visual resource for American English emotive speech and songs. The recording of stimuli was done by twelve males and twelve females. Audio video, audio-only, and video-only recordings are the three forms of recordings available for this corpus. For our research, we solely looked at audio-only speech and song recordings. This section includes 1440 speech files and 1012 song files recorded from 24 actors. Song files of a female actor are missing from the dataset. In total, 2452 audio files from RAVDESS were used to analyze the results. Speech includes neutral, calm, happy, sad, angry, fearful, surprise, and disgust expressions. The songs contain neutral, calm, happy, sad, angry, and fearful emotions. RAVDESS was chosen with SUBESCO for the experimental study of the proposed models for a reason. During the construction of SUBESCO the authors were influenced by RAVDESS's development technique. The audio-only stimuli in these two corpora are created and validated in the same way. In this regard, using RAVDESS for model comparisons and cross-lingual analysis was a sensible design decision.

B. SETUP 1: BASELINE EXPERIMENTS WITH SUBESCO

The same corpus was utilized for training, validation, and testing in the baseline experiments. For SUBESCO, all 7000 stimuli for seven emotions were taken into

TABLE 3. Instance distribution of SUBESCO.

	Setup 1				Setup 7			
Emotion	**Train**	**Validation**	**Test**	**Total**	**Train**	**Validation**	**Test**	**Total**
Anger	700	200	100	1000	328	72	170	570
Fear	700	200	100	1000	328	72	170	570
Happiness	700	200	100	1000	304	96	170	570
Neutral	700	200	100	1000	310	90	170	570
Sadness	700	200	100	1000	330	70	170	570
Disgust	700	200	100	1000	-	-	-	-
Surprise	700	200	100	1000	-	-	-	-
Total	4900	1400	700	7000	1600	400	850	2850

TABLE 4. Instance distribution of RAVDESS.

	Setup 2				Setup 7		
Emotion	**Train**	**Validation**	**Test**	**Total**	**Train**	**Validation**	**Total**
Angry	263	75	38	376	303	73	376
Fearful	263	75	38	376	305	71	376
Happy	263	75	38	376	282	94	376
Neutral	394	114	56	564	398	98	496
Sad	263	75	38	376	312	64	376
Total	1446	414	208	2068	1600	400	2000

consideration. 70% was allocated to training, 20% to validation, and 10% to testing. We used 4900 SUBESCO files to train the models. For cross-validation, another subset of 1400 files was employed. Testing was carried out on 700 files that had not been used in the training or validation stages. All the subsets were balanced in terms of speaker gender and emotion classes. Details can be found in Table 3.

C. SETUP 2: BASELINE EXPERIMENTS WITH RAVDESS

RAVDESS speech and song audio-only files were merged together for the six emotions namely neutral, calm, happy, sad, angry, and fearful. Because of the perceptual similarities highlighted in the research [49], calm audios were renamed as neutral audios. Finally, total five emotions were considered for this dataset. The training, validation, and testing ratio of the models for this setup is 70:20:10, which is the same as SUBESCO. Due to the lack of audios from a female speaker, it was not possible to segment this dataset in a balanced way in terms of emotions and speaker gender. Table 4 shows the distribution of RAVDESS for training, cross-validation, and testing.

D. SETUP 3 & 4: CROSS-CORPUS EXPERIMENTS

For the cross-corpus study, all of the explored models trained on one dataset were tested against another dataset. 1440 stimuli from RAVDESS audios were chosen to evaluate the models trained with SUBESCO (Setup 3), comprising of seven emotions (calm substituted by neutral). 850 audio samples from SUBESCO for five matching emotions: angry, fear, happiness, neutral, and sadness were used to assess the models trained with RAVDESS (Setup 4).

E. SETUP 5 & 6: TRANSFER LEARNING

To improve the experimental outcomes for cross-lingual analysis, next we used the concept of transfer learning in this step. It is the process of training a deep learning model using a large dataset of a domain, and then applying that model to solve a similar problem with a different small size dataset.

TABLE 5. Model comparisons for SUBESCO (Setup 1).

Model name	**WA%**	**F1%**
RM1 (2CNN+2FC)	83.14	82.68
RM2 (4CNN+LSTM)	76.14	76.22
4CNN+LSTM	79.14	79.13
4CNN+TDF+LSTM	85.57	85.56
4CNN+BLSTM	81.43	81.26
DCTFB (4CNN+TDF+BLSTM)	86.86	86.86
RM3 (7CNN+2FC)	78.43	78.15
7CNN+TDF+LSTM	84.43	84.26
7CNN+TDF+BLSTM	84.71	84.71

The weights of the original model are maintained unchanged for the initial layers in this technique, allowing those layers to reuse their expertise for modeling a new but related task. In Setup 5, all the models trained with RAVDESS dataset were used as a starting point to train and test 850 files of SUBESCO dataset with 80:20 split for training and testing, respectively. The weights of all layers were frozen to make them untrainable while removing the final dense layer from the trained models. Then, a new dense layer with softmax activation was introduced and trained for the SUBESCO data. For Setup 6, all of the pre-trained models from the SUBESCO dataset were utilized to transfer the knowledge of emotional features extraction to build new models, in order to train and categorize the RAVDESS dataset. There were 1440 RAVDESS audio-only speech files used in this experiment, with an 80:20 split for training and testing the new transferred models.

F. SETUP 7: EXPERIMENTS WITH MULTILINGUAL DATASETS

In the multilingual training experiment, 2000 stimuli from each corpus were considered. There are 4000 stimuli in total for the emotional states of neutral, happiness, sadness, anger, and fear. RAVDESS's neutral, happy, sad, angry, and fearful emotions were renamed to reflect this; where all the files labeled as calm for this dataset were considered as neutral. Testing was carried out on a subset of 850 audios from SUBESCO that had not been included in the training or validation stages. In the test set, each class has the same number of instances, which is 170. The file distributions of both datasets for this experiment are shown in Tables 3 and 4.

G. EVALUATION MATRICES

The classification task's performance was graded at two levels: overall accuracy and class accuracy. Weighted accuracy,

TABLE 6. Confusion matrix of SUBESCO dataset experiment for the proposed model.

Emotion	Anger	Disgust	Fear	Happiness	Neutral	Sadness	Surprise	Total
Anger	85	7	0	4	0	1	3	100
Disgust	5	76	1	7	2	3	6	100
Fear	0	2	85	1	4	3	5	100
Happiness	3	7	2	82	1	2	3	100
Neutral	0	2	4	0	88	6	0	100
Sadness	0	0	3	2	6	88	1	100
Surprise	1	6	3	0	1	0	89	100
Total	94	100	98	96	102	103	107	700

TABLE 7. Accuracy matrices (%) for SUBESCO dataset experiment for the proposed model.

Emotion	Recall (TPR)	Specificity (TNR)	Precision (PPV)	Nvalue (NPV)	F1 Score
Anger	80.00	97.56	84.21	96.77	82.05
Disgust	77.00	96.31	77.00	96.31	77.00
Fear	93.00	98.52	91.18	98.85	92.08
Happiness	83.00	97.56	84.69	97.24	83.84
Neutral	100.00	98.52	91.74	100.00	95.69
Sadness	86.00	99.50	96.63	97.72	91.01
Surprise	89.00	97.09	83.18	98.20	85.99

TABLE 8. Model comparisons for RAVDESS (Setup 2).

Model name	UA%	WA%	F1%
RM1 (2CNN+2FC)	76.92	75.41	74.76
RM2 (4CNN+LSTM)	75.96	74.70	74.43
4CNN+LSTM	77.40	76.62	76.50
4CNN+TDF+LSTM	77.88	76.97	76.84
4CNN+BLSTM	75.48	74.68	74.05
DCTFB (4CNN+TDF+BLSTM)	82.69	81.56	81.99
RM3 (7CNN+2FC)	63.94	62.05	61.88
7CNN+TDF+LSTM	73.08	71.54	71.55
7CNN+TDF+BLSTM	67.79	66.43	66.24

unweighted accuracy, and average F1 values were generated to assess overall performance. Sensitivity, specificity, precision, and negative prediction value, were employed to report class-wise accuracy. The following matrices are described in detail:

Unweighted accuracy is the ratio of total correct predictions and total instances of all classes in the dataset,

$$UA = \frac{correct\ predictions}{total\ instances} \tag{14}$$

Weighted accuracy weighs each class according to the number of correct predictions. Total correct predictions of a class $correct_i$ is divided by total instances $instance_i$ of that class i in the dataset. Then the sum of all weighted classes becomes:

$$WA = \sum_i \frac{correct_i}{instance_i} \tag{15}$$

Recall or **sensitivity** is the fraction of the number of correctly classified instances among the total instances of that class in the dataset. It also refers to the true positive rate (**TPR**) and is defined as:

$$recall = \frac{true\ positive}{true\ positive + false\ negative} \tag{16}$$

Specificity or **selectivity** is the fraction of correctly classified negative instances among all the negative instances of a class. It is also termed as the true negative rate (**TNR**) and

TABLE 9. Confusion matrix of RAVDESS dataset experiment for the proposed model.

Emotion	Angry	Fearful	Happy	Neutral	Sad	Total
Angry	35	1	0	1	1	38
Fearful	2	25	0	0	11	38
Happy	1	1	29	5	2	38
Neutral	0	0	1	53	2	56
Sad	1	4	0	3	30	38
Total	39	31	30	62	46	208

defined as:

$$specificity = \frac{true\ negative}{true\ negative + false\ positive} \tag{17}$$

Precision or positive prediction value (**PPV**) indicates the actual rate of correction. It is the fraction of correctly classified instances among the total number of classifications done for this class.

$$precision = \frac{true\ positive}{true\ positive + false\ positive} \tag{18}$$

Negative prediction value (NPV) is the rate of negative instances among all the negative classifications.

$$npv = \frac{true\ negative}{true\ negative + false\ negative} \tag{19}$$

The **F1** score is defined as:

$$F1 = 2 * \frac{precision * recall}{precision + recall} \tag{20}$$

V. RESULTS

A. ANALYSIS OF MODELS USING SUBESCO DATASET (SETUP 1)

We evaluated all the target SER models on the SUBESCO dataset. Table 5 illustrates the test results for all topologies, demonstrating that the DCTFB model has the highest accuracy amongst all of them. The WA accuracy achieved for this model is 86.86%, and the average f1 score is also 86.86%. In this situation, the WA and UA are the same because we utilized a balanced dataset for training, validation, and testing. As a result, only the WA

TABLE 10. Accuracy matrices (%) for RAVDESS dataset experiment for the proposed model.

Emotion	Recall (TPR)	Specificity (TNR)	Precision (PPV)	Nvalue (NPV)	F1 Score
Angry	92.11	97.70	89.74	98.27	90.91
Fearful	65.79	96.59	80.65	92.90	72.46
Happy	76.32	99.42	96.67	94.97	85.29
Neutral	94.64	94.41	85.48	98.06	89.83
Sad	78.95	91.40	65.22	95.51	71.43

TABLE 11. Model comparisons for cross-lingual analysis (Setup 3 & 4).

	Training with SUBESCO Testing with RAVDESS		Training with RAVDESS Testing with SUBESCO	
Model name	**UA%**	**WA%**	**UA%**	**WA%**
RM1 (2CNN+2FC)	22.29	16.74	27.65	27.65
RM2 (4CNN+LSTM)	21.60	17.16	31.41	31.41
4CNN+LSTM	23.06	18.75	34.35	34.35
4CNN+TDF+LSTM	27.43	22.94	31.53	31.53
4CNN+BLSTM	26.18	21.78	29.29	29.29
DCTFB (4CNN+TDF+BLSTM)	27.92	24.06	33.73	33.73
RM3 (7CNN+2FC)	20.21	18.25	21.76	21.76
7CNN+TDF+LSTM	24.86	21.11	22.12	22.12
7CNN+TDF+BLSTM	21.60	18.53	25.06	25.06

scores have been included in this report. With WA scores of 85.57% and 84.71%, respectively, 4CNN+TDF+LSTM and 7CNN+TDF+BLSTM performed very closely to the planned architecture. RM1 obtained a high accuracy but the training time was the longest for this model. In comparisons with other models tested here, we found that the seven-layer CNN architectures provided comparable performance at a fraction of the training time needed for other architectures. Table 6 displays a complete picture of the DCTFB model's performance on the SUBESCO dataset. Accuracy matrices for each emotion for this experiment are presented in Table 7. It shows that except for disgust, all of the emotion classes have accuracy rates above 80%.

B. ANALYSIS OF MODELS USING RAVDESS DATASET (SETUP 2)

Table 8 represents the prediction performances of all experimented models on the RAVDESS dataset. Because the dataset was not balanced, both the WA and the UA were provided with average F1 score. It can be seen that our proposed model has the highest perception accuracy (UA = 82.69%, WA = 81.56%, F1 = 81.99%). However, unlike the SUBESCO dataset, the 7CNN+TDF+BLSTM model has a substantially lower score (UA = 67.79 %, WA = 66.43 %, F1 = 66.24%) than the proposed model. RM1 achieved better result compared to RM2 and RM3, but still took the longest training time than other models. Table 9 presents the suggested model's confusion matrix for the RAVDESS dataset. Table 10 reports the accuracy matrices for this setup. It reveals that anger has the highest f1 (90.91%) and sad has the lowest (71.43%).

C. ANALYSIS OF CROSS-CORPUS TESTS (SETUP 3 & 4)

For cross-lingual analysis, all trained models of one dataset were tested on a subset of another dataset. A subset of 1440 files of RAVDESS for seven emotions was evaluated on the trained models of SUBESCO (Setup 3). While the trained models for the RAVDESS dataset were used to test the subset of SUBESCO (Setup 4) for five emotions. The prediction performance for this study is reported in Table 11. This demonstrates that for this experiment, all of the models had poor accuracy. SUBESCO (UA = 27.92%, WA = 24.06%) yielded the best testing accuracies when using the DCTFB model. 4CNN+LSTM performed slightly better in the case of RAVDESS. The perceptual performance of RM3 was the lowest. We know that there are linguistic and cultural barriers to classifying emotions of a language using a deep learning model trained on another language. Although there have been a few research on this type of experiment, the results have not been as promising as expected [50].

D. ANALYSIS OF TRANSFER LEARNING (SETUP 5 & 6)

From Table 12 it is evident that transfer learning improves the efficiency of recognition models compared to the performances of cross-corpus training. In Setup 5, 4CNN+TDF+LSTM model performed the best, the proposed model performed very closely to this result. The 4CNN block architectures show significantly improved results while using the TDF layer. Likewise the outcome of Setup 5, Setup 6 also shows that trained models of RAVDESS, as a transferred model for SUBESCO subset, boost recognition accuracy. Table 13 shows that our proposed model achieved the highest accuracy (UA = 59.72%, WA = 56.20%, F1 = 59.72%) for this experiment. 4CNN+TDF+LSTM obtained the second highest accuracy for UA accuracy of 54.51%.

E. ANALYSIS OF MODELS USING MULTILINGUAL DATASETS (SETUP 7)

It was discovered that training an SER model with several datasets of different languages improves its performance in detecting the emotions of those languages. Using the same number of files from both datasets to train the models improved the perception accuracy significantly compared to other cross-lingual experiments. Four CNN block architectures gave the best results while testing the models with

TABLE 12. Transfer learning using trained models of RAVDESS (Setup 5).

Model name	UA%	WA%	F1%
RM1 (2CNN+2FC)	37.65	38.67	33.62
RM2 (4CNN+LSTM)	42.94	42.41	39.78
4CNN+LSTM	44.12	44.45	43.22
4CNN+TDF+LSTM	62.94	63.61	62.58
4CNN+BLSTM	44.12	45.26	42.69
DCTFB (4CNN+TDF+BLSTM)	61.18	62.41	59.93
RM3 (7CNN+2FC)	38.82	38.91	38.66
7CNN+TDF+LSTM	34.12	34.71	33.37
7CNN+TDF+BLSTM	48.24	48.55	46.57

TABLE 13. Transfer learning using trained models of SUBESCO (Setup 6).

Model name	UA%	WA%	F1%
RM1 (2CNN+2FC)	38.19	32.87	29.18
RM2 (4CNN+LSTM)	40.97	39.41	36.49
4CNN+LSTM	43.75	41.18	39.11
4CNN+TDF+LSTM	54.51	51.55	51.43
4CNN+BLSTM	44.79	42.62	38.79
DCTFB (4CNN+TDF+BLSTM)	59.72	56.20	59.72
RM3 (7CNN+2FC)	36.81	33.58	30.33
7CNN+TDF+LSTM	43.40	40.27	39.04
7CNN+TDF+BLSTM	42.36	39.59	38.37

TABLE 14. Model comparisons for multi-lingual training (Setup 7).

Model name	WA%	F1%
RM1 (2CNN+2FC)	59.88	56.78
RM2 (4CNN+LSTM)	52.82	48.67
4CNN+LSTM	60.94	59.94
4CNN+TDF+LSTM	62.94	61.51
4CNN+BLSTM	52.24	48.19
DCTFB (4CNN+TDF+BLSTM)	64.94	62.80
RM3 (7CNN+2FC)	46.59	44.27
7CNN+TDF+LSTM	49.53	45.88
7CNN+TDF+BLSTM	49.53	46.32

a subset of the SUBESCO dataset for the five emotions. Table 14 shows the overall performance of all models. With WA = 64.94% and F1 = 62.80%, the DCTFB model had the highest accuracy. With WA = 46.59% and F1 = 44.27%, the RM3 model had the lowest perception rate. Because the testing subset was balanced in terms of speaker gender and emotion classes, both the WA and UA are the same. For this reason, only UA scores have been reported in this table. Table 15 shows the confusion matrix for the proposed model of this study. We see that anger achieved the highest recognition rate, while fear had the second-highest accuracy rate. Sadness is the least recognized emotion and it is largely confused with fear. The result reveals some important facts, such as anger has a common way of expressiveness through loudness, in many cultures. This may have an impact on the outcome of this study. However, these observations are not sufficient to draw any conclusions from a multilingual trial. A more in-depth investigation might be beneficial in determining the similarities of emotion expressions across cultures. Emotion-wise accuracy matrices are shown in Table 16.

VI. DISCUSSION

In this work, a deep CNN block was used to learn high-level local emotional features, while the LSTM was utilized to learn long-term contextual global features. We experimented with nine distinct topologies using a variety of

TABLE 15. Confusion matrix for multi-lingual experiment for the proposed model.

Emotion	Anger	Fear	Happiness	Neutral	Sadness	Total
Anger	163	2	5	0	0	170
Fear	12	146	3	1	8	170
Happiness	59	11	94	3	3	170
Neutral	6	13	30	97	24	170
Sadness	12	59	19	28	52	170
Total	252	231	151	129	87	850

layers and parameters. Comparing the best results for all the experimented models it was found that the proposed model (DCTFB) performed consistently better and outperformed all, almost in all cases. To demonstrate the performance of the trained models, both weighted and unweighted accuracies were calculated. WA is important to use when class distribution is not balanced in the dataset so that the reported performance is not biased towards the larger classes. For both datasets, we attempted to employ a balanced split because studies have shown that a DNN classifier trained on a balanced dataset produces the best results and is more robust [51], [52]. In the case of RAVDESS, all of the emotions have the same number of samples, except for neutral.

WA, UA, and F1 scores calculated using different setups for the proposed model, have been compared in Figure 4. It was discovered that the introduction of TDF enhanced the prediction accuracy using the salient discriminating feature regardless of the training language type. In terms of the performance of both databases, SUBESCO performed better than RAVDESS. One possible explanation for this is that SUBESCO has more balanced instances of each class than RAVDESS. This is the first SER implementation of SUBESCO, though the audio-only files of the RAVDESS dataset have recently been used in a variety of research for emotion classification. A study [22] obtained 79.5% UA for 1440 RAVDESS files using a CNN classifier. Issa *et al.* used CNN to classify RAVDESS audio files based on a combination of spectral parameters and reported a recognition rate of 71.61% [53]. For those files, Zisad *et al.* obtained an average accuracy of 82.5% employing data augmentation, and it used a CNN classifier to distinguish emotion from the dataset [54]. For the same subset of the RAVDESS dataset, a real-time speech recognition system using transfer learning techniques for the VGG16 pre-trained model showed an emotion perception rate of 62.51% [55]. Patel *et al.* presented another work in which they utilized an autoencoder to reduce dimensionality and used a CNN classifier to reach an accuracy of 80% for RAVDESS audio-only files [56]. A system consisting of CNN and head fusion multi-head attention achieved 77.8% WA for the audio-only speech files of RAVDESS in recent work [57]. The most recent SER system using this dataset was presented in [58]. The system employed a GA-optimized feature set to classify emotions using SVM trained and tested with the 1440 speech files of RAVDESS. It achieved a UA of 82.5% for a speaker-dependent experiment carried out using this subset of the corpus. In contrast, our suggested model utilized both the speech and song files for audio-only recordings of

TABLE 16. Accuracy matrices (%) for multi-lingual experiment for the proposed model.

Emotion	Recall (TPR)	Specificity (TNR)	Precision (PPV)	Nvalue (NPV)	F1 Score
Anger	95.88	88.43	64.68	98.98	77.25
Fear	85.88	88.89	63.20	96.59	72.82
Happiness	55.29	92.27	62.25	89.95	58.57
Neutral	57.06	95.51	75.19	90.31	64.88
Sadness	30.59	95.10	59.77	85.21	40.47

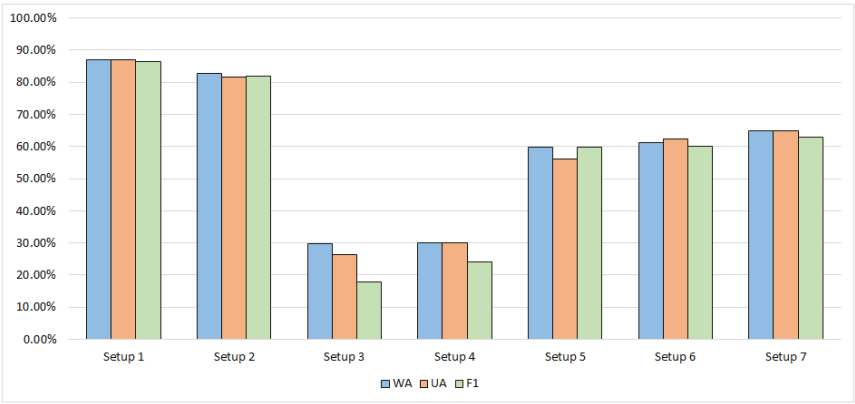

FIGURE 4. WA, UA and F1 scores for different setups using DCTFB model.

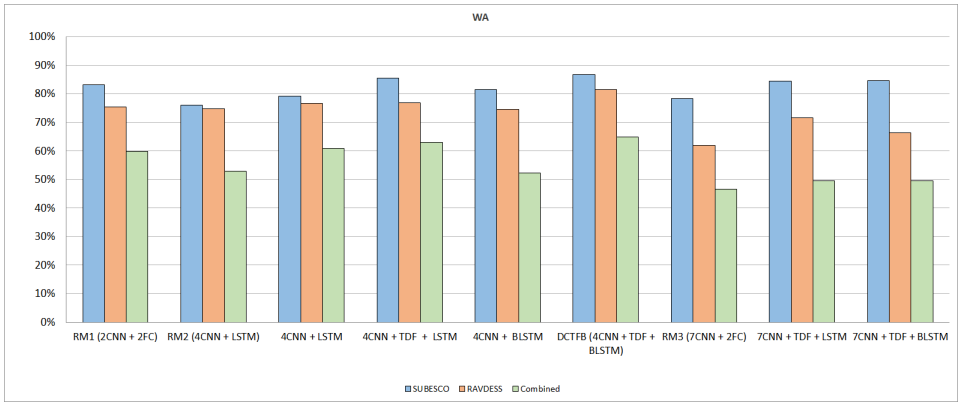

FIGURE 5. WA comparisons for different training datasets.

RAVDESS. It achieved a UA of 82.7%, which is the greatest attained accuracy for RAVDESS audio-only files recorded to date.

In the scenarios of seven layers CNN, there is a consistent improvement in WA for using TDF with LSTM/BLSTM when experimented with both the datasets. Four-layer CNN models also show the best perception accuracies when accompanied by the TDF layer. Seven-layer models, on the other hand, demonstrate good accuracies with substantially less training time. This is due to the reduced size of feature maps in these architectures. Figure 5 depicts the comparisons of the overall performances of all the tested models for setups 1, 2 and 7.

Cross-lingual study (setups 3-7) shows poor performances when the model is trained with an unknown dataset of another language. We observed that the use of pre-trained models to apply transfer learning for another dataset significantly boosts performance of the deep learning models. It was also found that training the models with aggregated datasets also enhances performance to a large extent. The line graph in Figure 6 compares the UAs of all the experimented models for all the seven setups. Subsets of SUBESCO and RAVDESS were chosen rather than the whole datasets to find the effect of deep learning models trained on a larger dataset and tested on a smaller dataset which is very useful to classify emotions for low-resource languages. The limitation of our study is that we experimented only two datasets of completely different languages of different cultures. Further research needs to be conducted for cross-lingual study involving more datasets.

VII. CONCLUSION

In this study, a novel architecture for SER has been proposed that demonstrated state-of-the-art prediction performance for the Bangla dataset SUBESCO, as well as the English dataset RAVDESS. Weighted accuracy, unweighted accuracy, precision, recall, NPV, specificity, and F1 scores were used as performance parameters to report the statistics. To combat the overfitting of training the models, batch

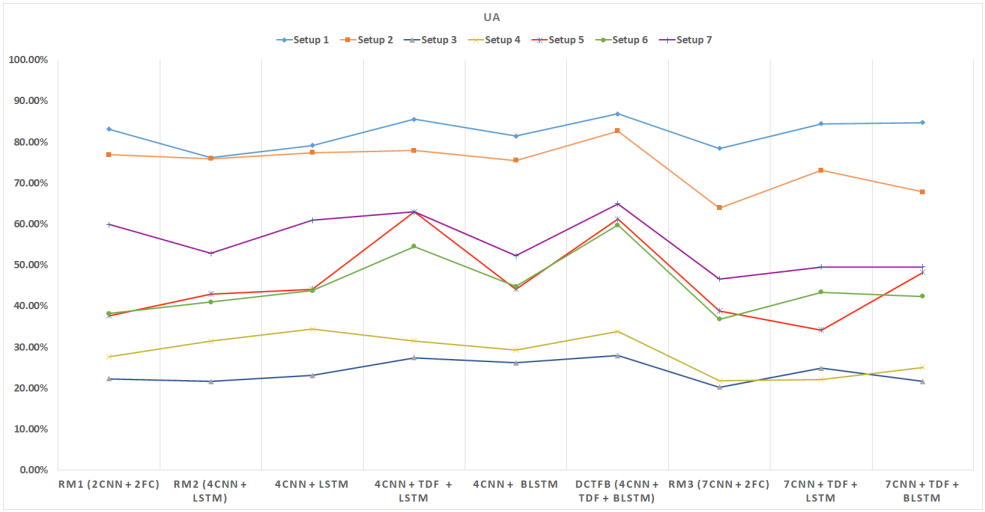

FIGURE 6. Line graph comparing UAs of all models for all setups.

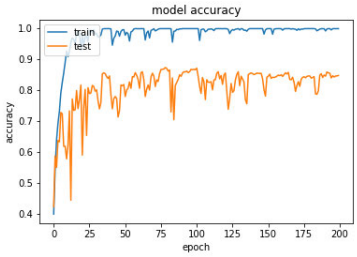

FIGURE 7. Accuracy plot of SUBESCO training and testing with the proposed model.

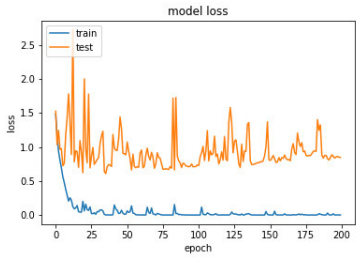

FIGURE 8. Loss plot of SUBESCO training and testing with the proposed model.

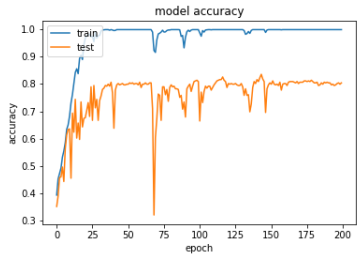

FIGURE 9. Accuracy plot of RAVDESS training and testing with the proposed model.

normalization, dropout, and cross-validation techniques were employed. Baseline experiment with the proposed model indicates a promising outcome for the deep learning evaluation of SUBESCO which is 86.86%. The proposed model also outperformed all the existing implementations of RAVDESS

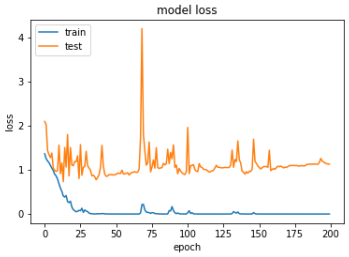

FIGURE 10. Loss plot of RAVDESS training and testing with the proposed model.

audio-only files (speech and song) with an accuracy of 82.7%. A cross-lingual study implemented using transfer learning shows that models trained on a SUBESCO dataset can be applied for other languages as well with satisfactory performance. As Bangla is still considered a low-resource language, we anticipate that this work will provide the Bangla research paradigm with a new direction. In our future research, we plan to extend this work using a multi-dimensional dataset for Bangla. Also, we wish to conduct a cross-lingual study for prominent Indo-Aryan languages.

APPENDIX
TRAINING PLOTS FOR BASELINE EXPERIMENTS

Training history of Setup 1 and 2 for the proposed model are presented in Figures 7 to 10.

ACKNOWLEDGMENT

This work has been done as a part of a Ph.D. research project which was initially supported by the Higher Education Quality Enhancement Project (AIF Window 4, CP 3888) for "The Development of Multi-Platform Speech and Language Processing Software for Bangla." The authors would like to thank Dr. Yasmeen Haque for proofreading the manuscript, and also would like to thank all of the creators of RAVDESS dataset for developing and sharing their invaluable resources.

REFERENCES

[1] B. W. Schuller, "Speech emotion recognition: Two decades in a nutshell, benchmarks, and ongoing trends," *Commun. ACM*, vol. 61, no. 5, pp. 90–99, 2018.

[2] S. Ramakrishnan, "Recognition of emotion from speech: A review," *Speech Enhancement, Modeling Recognit. Algorithms Appl.*, vol. 7, pp. 121–137, Mar. 2012.

[3] M. B. Akçay and K. Oğuz, "Speech emotion recognition: Emotional models, databases, features, preprocessing methods, supporting modalities, and classifiers," *Speech Commun.*, vol. 116, pp. 56–76, Jan. 2020.

[4] E. Harmon-Jones, C. Harmon-Jones, and E. Summerell, "On the importance of both dimensional and discrete models of emotion," *Behav. Sci.*, vol. 7, no. 4, p. 66, 2017.

[5] K. Choi, G. Fazekas, and M. Sandler, "Automatic tagging using deep convolutional neural networks," 2016, *arXiv:1606.00298*.

[6] A. J. R. Simpson, G. Roma, and M. D. Plumbley, "Deep karaoke: Extracting vocals from musical mixtures using a convolutional deep neural network," in *Latent Variable Analysis and Signal Separation*. Cham, Switzerland: Springer, 2015, pp. 429–436.

[7] S. Latif, R. Rana, S. Khalifa, R. Jurdak, and J. Epps, "Direct modelling of speech emotion from raw speech," 2019, *arXiv:1904.03833*.

[8] Z. Peng, X. Li, Z. Zhu, M. Unoki, J. Dang, and M. Akagi, "Speech emotion recognition using 3D convolutions and attention-based sliding recurrent networks with auditory front-ends," *IEEE Access*, vol. 8, pp. 16560–16572, 2020.

[9] M. Neumann and N. T. Vu, "Attentive convolutional neural network based speech emotion recognition: A study on the impact of input features, signal length, and acted speech," 2017, *arXiv:1706.00612*.

[10] C.-W. Huang and S. S. Narayanan, "Deep convolutional recurrent neural network with attention mechanism for robust speech emotion recognition," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, Jul. 2017, pp. 583–588.

[11] S. Latif, R. Rana, S. Younis, J. Qadir, and J. Epps, "Transfer learning for improving speech emotion classification accuracy," 2018, *arXiv:1801.06353*.

[12] M. M. Rahman, D. R. Dipta, and M. M. Hasan, "Dynamic time warping assisted SVM classifier for Bangla speech recognition," in *Proc. Int. Conf. Comput., Commun., Chem., Mater. Electron. Eng. (IC4ME2)*, Feb. 2018, pp. 1–6.

[13] A. M. Badshah, J. Ahmad, N. Rahim, and S. W. Baik, "Speech emotion recognition from spectrograms with deep convolutional neural network," in *Proc. Int. Conf. Platform Technol. Service (PlatCon)*, Feb. 2017, pp. 1–5.

[14] A. Satt, S. Rozenberg, and R. Hoory, "Efficient emotion recognition from speech using deep learning on spectrograms," in *Proc. Interspeech*, Aug. 2017, pp. 1089–1093.

[15] C. Etienne, G. Fidanza, A. Petrovskii, L. Devillers, and B. Schmauch, "CNN+LSTM architecture for speech emotion recognition with data augmentation," 2018, *arXiv:1802.05630*.

[16] M. Chen, X. He, J. Yang, and H. Zhang, "3-D convolutional recurrent neural networks with attention model for speech emotion recognition," *IEEE Signal Process. Lett.*, vol. 25, no. 10, pp. 1440–1444, Oct. 2018.

[17] Z. Zhao, Y. Zhao, Z. Bao, H. Wang, Z. Zhang, and C. Li, "Deep spectrum feature representations for speech emotion recognition," in *Proc. Joint Workshop 4th Workshop Affect. Social Multimedia Comput. 1st Multi-Modal Affect. Comput. Large-Scale Multimedia Data*, 2018, pp. 27–33.

[18] Y. Zhang, J. Du, Z. Wang, J. Zhang, and Y. Tu, "Attention based fully convolutional network for speech emotion recognition," in *Proc. Asia–Pacific Signal Inf. Process. Assoc. Annu. Summit Conf. (APSIPA ASC)*, Nov. 2018, pp. 1771–1775.

[19] D. Ghosal, N. Majumder, S. Poria, N. Chhaya, and A. Gelbukh, "DialogueGCN: A graph convolutional neural network for emotion recognition in conversation," 2019, *arXiv:1908.11540*.

[20] Z. Zhao, Z. Bao, Z. Zhang, N. Cummins, H. Wang, and B. W. Schuller, "Attention-enhanced connectionist temporal classification for discrete speech emotion recognition," in *Proc. Interspeech*, Sep. 2019, pp. 206–210.

[21] Z. Zhao, Z. Bao, Y. Zhao, Z. Zhang, N. Cummins, Z. Ren, and B. Schuller, "Exploring deep spectrum representations via attention-based recurrent and convolutional neural networks for speech emotion recognition," *IEEE Access*, vol. 7, pp. 97515–97525, 2019.

[22] S. Kwon, "A CNN-assisted enhanced audio signal processing for speech emotion recognition," *Sensors*, vol. 20, no. 1, p. 183, 2020.

[23] M. S. Hossain and G. Muhammad, "Emotion recognition using secure edge and cloud computing," *Inf. Sci.*, vol. 504, pp. 589–601, Dec. 2019.

[24] M. S. Hossain and G. Muhammad, "Emotion recognition using deep learning approach from audio-visual emotional big data," *Inf. Fusion*, vol. 49, pp. 69–78, Sep. 2019.

[25] D. Tang, P. Kuppens, L. Geurts, and T. van Waterschoot, "End-to-end speech emotion recognition using a novel context-stacking dilated convolution neural network," *EURASIP J. Audio, Speech, Music Process.*, vol. 2021, no. 1, pp. 1–16, Dec. 2021.

[26] B. McFee, C. Raffel, D. Liang, D. Ellis, M. McVicar, E. Battenberg, and O. Nieto, "Librosa: Audio and music signal analysis in Python," in *Proc. 14th Python Sci. Conf.*, 2015, pp. 18–25.

[27] J. Chen, J. Benesty, Y. Huang, and S. Doclo, "New insights into the noise reduction Wiener filter," *IEEE Trans. Audio, Speech, Language Process.*, vol. 14, no. 4, pp. 1218–1234, Jul. 2006.

[28] E. Sejdić, I. Djurović, and J. Jiang, "Time–frequency feature representation using energy concentration: An overview of recent advances," *Digit. Signal Process.*, vol. 19, no. 1, pp. 153–183, Jan. 2009.

[29] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," *Insights Imag.*, vol. 9, pp. 611–629, Jun. 2018.

[30] L. Alzubaidi, J. Zhang, A. J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M. A. Fadhel, M. Al-Amidie, and L. Farhan, "Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions," *J. Big Data*, vol. 8, no. 1, pp. 1–74, Dec. 2021.

[31] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Proc. Interspeech*, Feb. 2014, pp. 338–342.

[32] M. Wllmer, B. Schuller, F. Eyben, and G. Rigoll, "Combining long short-term memory and dynamic Bayesian networks for incremental emotion-sensitive artificial listening," *IEEE J. Sel. Topics Signal Process.*, vol. 4, no. 5, pp. 867–881, Oct. 2010.

[33] P. Jiang, H. Fu, H. Tao, P. Lei, and L. Zhao, "Parallelized convolutional recurrent neural network with spectral features for speech emotion recognition," *IEEE Access*, vol. 7, pp. 90368–90377, 2019.

[34] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[35] D. Li, J. Liu, Z. Yang, L. Sun, and Z. Wang, "Speech emotion recognition using recurrent neural networks with directional self-attention," *Expert Syst. Appl.*, vol. 173, Jul. 2021, Art. no. 114683.

[36] J. Zhao, X. Mao, and L. Chen, "Speech emotion recognition using deep 1D & 2D CNN LSTM networks," *Biomed. Signal Process. Control*, vol. 47, pp. 312–323, Jan. 2019.

[37] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[38] D.-A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and accurate deep network learning by exponential linear units (ELUs)," 2015, *arXiv:1511.07289*.

[39] A. Martins and R. Astudillo, "From softmax to sparsemax: A sparse model of attention and multi-label classification," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1614–1623.

[40] K. He and J. Sun, "Convolutional neural networks at constrained time cost," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 5353–5360.

[41] E. Tsironi, P. Barros, C. Weber, and S. Wermter, "An analysis of convolutional long short-term memory recurrent neural networks for gesture recognition," *Neurocomputing*, vol. 268, pp. 76–86, Dec. 2017.

[42] J. X. Chen, P. W. Zhang, Z. J. Mao, Y. F. Huang, D. M. Jiang, and Y. N. Zhang, "Accurate EEG-based emotion recognition on combined features using deep convolutional neural networks," *IEEE Access*, vol. 7, pp. 44317–44328, 2019.

[43] A. Gulli and S. Pal, *Deep Learning With Keras*. Birmingham, U.K.: Packt Publishing Ltd, 2017.

[44] Z. Zhang and M. R. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Proc. 32nd Conf. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 1–11.

[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.

[46] T. Kluyver *et al.*, *Jupyter Notebooks—A Publishing Format for Reproducible Computational Workflows*. IOS Press, 2016.

[47] E. Bisong, *Building Machine Learning and Deep Learning Models on Google Cloud Platform*. Berkeley, CA, USA: Springer, 2019.

[48] S. Sultana, M. S. Rahman, M. R. Selim, and M. Z. Iqbal, "SUST Bangla emotional speech corpus (SUBESCO): An audio-only emotional speech corpus for Bangla," *PLoS ONE*, vol. 16, no. 4, 2021, Art. no. e0250173.

[49] S. R. Livingstone and F. A. Russo, "The Ryerson audio-visual database of emotional speech and song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English," *PLoS ONE*, vol. 13, no. 5, 2018, Art. no. e0196391.
[50] J. Parry, D. Palaz, G. Clarke, P. Lecomte, R. Mead, M. Berger, and G. Hofer, "Analysis of deep learning architectures for cross-corpus speech emotion recognition," in *Proc. Interspeech*, Sep. 2019, pp. 1656–1660.
[51] P. Hensman and D. Masko, "The impact of imbalanced training data for convolutional neural networks," Degree Project Comput. Sci., KTH Roy. Inst. Technol., Stockholm, Sweden, May 2015.
[52] Y. Zhang and W. Deng, "Class-balanced training for deep face recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2020, pp. 824–825.
[53] D. Issa, M. F. Demirci, and A. Yazici, "Speech emotion recognition with deep convolutional neural networks," *Biomed. Signal Process. Control*, vol. 59, May 2020, Art. no. 101894.
[54] S. N. Zisad, M. S. Hossain, and K. Andersson, "Speech emotion recognition in neurological disorders using convolutional neural network," in *Proc. Int. Conf. Brain Informat.* Cham, Switzerland: Springer, 2020, pp. 287–296.
[55] S.-A. Naas and S. Sigg, "Real-time emotion recognition for sales," in *Proc. 16th Int. Conf. Mobility, Sens. Netw. (MSN)*, Dec. 2020, pp. 584–591.
[56] N. Patel, S. Patel, and S. H. Mankad, "Impact of autoencoder based compact representation on emotion detection from audio," *J. Ambient Intell. Humanized Comput.*, pp. 1–19, Mar. 2021.
[57] M. Xu, F. Zhang, and W. Zhang, "Head fusion: Improving the accuracy and robustness of speech emotion recognition on the IEMOCAP and RAVDESS dataset," *IEEE Access*, vol. 9, pp. 74539–74549, 2021.
[58] S. Kanwal and S. Asghar, "Speech emotion recognition using clustering based Ga-optimized feature set," *IEEE Access*, vol. 9, pp. 125830–125842, 2021.

SADIA SULTANA was born in Sylhet, Bangladesh, in 1985. She received the B.Sc. (Hons.) and M.Sc. degrees from the Department of Computer Science and Engineering (CSE). She is currently pursuing the Ph.D. degree with CSE, Shahjalal University of Science and Technology (SUST), Sylhet. She is currently an Associate Professor with the CSE, SUST. Her research interests include speech analysis, data science, artificial intelligence, and pattern recognition. Her exceptional bachelor's degree outcome earned her the Chancellor's Gold Medal. She also obtained an outstanding result in her M.Sc. degree.

M. ZAFAR IQBAL was born in Sylhet, Bangladesh, in 1952. He received the B.Sc. degree in physics and the M.Sc. degree in theoretical physics from the University of Dhaka, Bangladesh, in 1973 and 1974, respectively, and the Ph.D. degree in experimental physics from the University of Washington, Seattle, WA, USA, in 1982. He worked as a Postdoctoral Researcher with the California Institute of Technology (Caltech), from 1983 to 1988 (mainly on Norman Bridge Laboratory of Physics). Then, he joined Bell Communications Research (Bellcore), a separate corporation from the Bell Labs (now Telcordia Technologies), as a Research Scientist, and left the institute, in 1994. He is currently a Retired Professor in computer science and engineering with the Shahjalal University of Science and Technology, Sylhet. He is also a well-known Science Fiction Novelist, a Physicist, a Professor, and an Activist from Bangladesh. His current research interests include Bangla speech-to-text, Bangla natural language processing, Bangla OCR, WDM networks, optical communication, GPON, non-linear optics, and robotics. He served as the Vice President of the Bangladesh Mathematical Olympiad Committee. He played a leading role in founding the Bangladesh Mathematical Olympiad and popularized mathematics among Bangladeshi youths at local and international level. He received the Rotary SEED Award for his contributions to education, in 2011.

M. REZA SELIM was born in Jamalpur, Bangladesh, in 1976. He received the B.Sc. and M.Sc. degrees in electronics and computer science from the Shahjalal University of Science and Technology, Sylhet, Bangladesh, in 1995 and 1996, respectively, and the Ph.D. degree in information and computer sciences from Saitama University, Saitama, Japan, in 2008. He began his career as a Junior Faculty Member with the Shahjalal University of Science and Technology, in 1997, where he is currently a Professor. His research interests include natural language processing, machine translation, and P2P networks.

MD. MIJANUR RASHID was born in Sylhet, Bangladesh, in 1983. He received the B.Sc. degree in statistics from the Shahjalal University of Science and Technology, Sylhet, and the M.Sc. degree in econometrics from The University of Manchester, U.K. Since graduation, he has been working in several well respected multinational research and analysis companies (e.g., Kantar Group, and Adelphi Omnicom Group) dealing with data science and big data engineering. In addition, he worked in regulation at the Solicitors regulation authority as the Head of Analytics and was instrumental in introducing machine learning and data science solutions. He is currently working as a Senior Data Scientist at REPL, Accenture. His research interests include machine learning and big data analysis.

M. SHAHIDUR RAHMAN (Senior Member, IEEE) was born in Jamalpur, Bangladesh, in 1975. He received the B.Sc. and M.Sc. degrees in electronics and computer science from the Shahjalal University of Science and Technology, Sylhet, Bangladesh, in 1995 and 1997, respectively, and the Ph.D. degree in mathematical information systems from Saitama University, Saitama, Japan, in 2006. He began teaching with the Department of Computer Science and Engineering, Shahjalal University of Science and Technology, in 1997, where he is currently a Professor. He was a JSPS Postdoctoral Research Fellow at Saitama University, from 2009 to 2011. His research interests include voice analysis, speech synthesis, speech recognition, bone-conducted speech augmentation, and digital signal processing.