DNN-Based Cross-Lingual Voice Conversion Using Bottleneck Features

M. Kiran Reddy · K. Sreenivasa Rao

Published online: 7 November 2019
© Springer Science+Business Media, LLC, part of Springer Nature 2019

Abstract
Cross-lingual voice conversion (CLVC) is quite challenging since the source and target speakers speak different languages. It is essential for various applications such as developing mixed-language speech synthesis systems, customization of speaking devices, etc. This paper proposes a deep neural network (DNN)-based approach utilizing bottleneck features for CLVC. In the proposed method, the speaker-independent information present in the speech signals from different languages is represented by using the bottleneck features extracted from a deep auto-encoder. A DNN model is trained to learn the mapping between bottleneck features and the corresponding spectral features of the target speaker. The proposed approach can capture speaker-specific characteristics of a target speaker, and requires no speech data from the source speaker during training. The performance of the proposed method is evaluated using data from three Indian languages: Telugu, Tamil and Malayalam. The experimental results show that the proposed method can effectively convert the source speaker voice to target speaker voice in a cross-lingual scenario.

Keywords Cross-lingual voice conversion · Deep autoencoder · Deep neural network · Gaussian mixture model

1 Introduction

Voice conversion (VC) is the process of modifying the speech utterances of a source speaker so that it sounds like it was uttered by a target speaker. There are several applications of VC such as voice restoration, customization of text-to-speech systems, etc [1]. Based on the language that source and target speakers speak, VC can be divided into two categories, namely, (i) Intra-lingual VC and (ii) cross-lingual VC. Intra-lingual VC (ILVC) assumes that the source and target speakers speak in the same language. The traditional ILVC techniques are
mostly based on Gaussian mixture models (GMMs) [1–3]. In recent years, neural networks (NNs) have achieved state-of-the-art results in many application such as Image retrieval [4], human pose estimation [5], visual question answering [6], face-pose estimation [7], dimension reduction [8], image recognition [9], place recognition [10] and so on. Owing to their remarkable success in many applications, several researchers have explored the use of neural networks for ILVC [11–20]. It is observed that a NN-based VC system performs as good as, or better than, a GMM-based VC system.

Cross-lingual VC (CLVC) is a special case of VC where source and target speakers speak different languages (the languages should be acoustically closer). Here, the aim is to convert the utterance spoken by source speaker in such a way that it is spoken untranslated by the target speaker. The ILVC approaches are not suitable for CLVC since the languages spoken by source and target speakers are not the same. In the literature, very few attempts have been made to develop CLVC systems. In [21], vector quantization (VQ) based CLVC is developed for the languages, Japanese and English. This approach does not sufficiently preserve the speaker’s identity, where the feature space of the transformed envelope is limited to a discrete set of envelopes. In [22], vocal tract length normalization is employed for developing CLVC system for the languages German and English. Erro and Moreno [23] employed an iterative frame selection approach to perform cross-language VC. An eigenvoice (EV) GMM-based method for CLVC is proposed in [24]. Here, parallel data from source speaker and multiple pre-stored data from other speakers are used for training EV-GMM. The EV-GMM is adapted using few arbitrary utterances from the target speaker in a different language, to obtain the conversion model.

The aforementioned CLVC methods require either parallel or non-parallel speech data from the source speaker in the language of target speaker to build the conversion models. This is a limitation to an application where an arbitrary source speaker’s voice has to be transformed to a target speaker without recording anything apriori [11]. The recent research focuses on investigating conversion models which can capture target speaker-specific characteristics, and avoid the need for source speaker’s data in training stage. In [25], speaker-specific GMMs are trained using data from target speaker alone, to achieve CLVC for Indian languages. This approach can convert speech of an arbitrary source speaker into a given target speaker. However, it suffers from the oversmoothing effect due to GMM-based conversion. Recently, attempts are being made to capture speaker-specific characteristics using neural network (NN) models. Training speaker-specific NN models is not as straightforward as in the case of GMM. Here, first we need to efficiently extract both speaker-dependent and speaker-independent features from speech signals. Then, a NN can be used to obtain the mapping function between the speaker-dependent and speaker-independent features computed from speech signals of the target speaker. While spectral features (such as Mel-generalized cepstral coefficients (MGCCs) used in this work) can be considered for representing speaker-dependent characteristics, there is search for an efficient representation of speaker-independent information. Recently, some studies have investigated Phonetic posteriorGrams (PPGs) as speaker-independent features for CLVC [17,26]. PPGs are typically estimated by training a Speaker-Independent Automatic Speech Recognition (SI-ASR) system using a large multi-speaker database [17]. The performance of PPG-based approaches is limited owing to difficulties in training an accurate ASR system in cross-lingual scenario [17]. Another drawback is that these approaches are transcription dependent i.e., they require precise text transcriptions corresponding to the speech signals. Intensive human effort and resources are required to transcribe the speech data. Hence, an alternative representation for speaker-independent information is needed to develop an efficient and transcription-free approach for CLVC.
In this work, we propose a NN-based approach to CLVC by using bottleneck features as speaker-independent features. The bottleneck features are extracted from a deep auto-encoder (DAE) model, trained using multi-speaker speech data. It is observed that when a DAE is trained with speech signals from multiple speakers, the features obtained from bottleneck layer possess speaker-invariant characteristics.

In training phase, a DNN is trained to learn mapping between target speaker’s MGCCs and the corresponding bottleneck features extracted from DAE. During conversion, the trained DNN model is used to convert the bottleneck features extracted from speech of an arbitrary source speaker to MGCCs of the desired target speaker. The performance of the proposed method is evaluated using speech data from three Indian languages: Telugu, Tamil and Malayalam. The experimental results demonstrate excellent performance by the proposed method. The following list briefly summarizes the contributions of this work.

1. Use of a multi-speaker database for training a deep auto-encoder to perform speaker-independent encoding i.e to convert the input speaker-dependent features to speaker-independent features (referred to as bottleneck features).

2. Utilization of bottleneck features for developing a novel NN-based approach to perform CLVC. The proposed approach can capture target speaker-specific characteristics and avoid the need for speech data from a source speaker to train a voice conversion model. Unlike existing NN-based CLVC approach, the proposed approach is transcription-free i.e. it needs no manual effort in transcribing the speech data.

3. Demonstration of the efficacy of proposed method through experiments over databases from multiple Indian languages. Our experimental results show that: (a) the bottleneck features are not only speaker-independent, but also language-independent; (b) the proposed approach performs better than the state-of-the-art approaches.

This paper is organized as follows: Section 2 gives details of the baseline GMM-based CLVC approach. The proposed CLVC approach is described in Sect. 3. The performance of the proposed method is evaluated and compared with two existing CLVC techniques in Sect. 4. Section 5 summarizes and concludes the present work.
2 Baseline GMM-Based CLVC Technique

For CLVC, approaches which can capture speaker-specific characteristics of a target speaker are required. Such approaches can transform the speech of any arbitrary source speaker to a pre-defined target speaker, without recording anything apriori from the source speaker. As mentioned earlier, GMMs can be directly trained with spectral features from target speaker alone to capture speaker-specific characteristics. Hence, the GMM-based approach [25] is chosen as baseline in this work. The steps in training a GMM to capture target speaker-specific characteristics for CLVC are as follows [25]:

(1) Let $L_S$ and $L_T$ denote the source language and target language, respectively.

(2) In training phase, extract spectral features (e.g. MGCCs) corresponding to the utterances of $L_T$ and train a GMM with M mixture components that can be used as a tokenizer. The model is denoted as,

$$\lambda_{L_T} = \{w_i, \mu_i, \Sigma_i\}; \quad i = 1, 2, \ldots, M$$  \hspace{1cm} (1)

here, $w_i$, $\mu_i$, and $\Sigma_i$ are the weight, mean vector, and the covariance matrix of the $i_{th}$ mixture component, respectively.

(3) During conversion stage, the feature vectors corresponding to the utterances spoken by the source speaker in the source language are extracted. Let $J$ denote the total number of utterances in the $L_S$. From each of these utterances, extract the spectral feature vectors. Let us denote the feature vectors of the source speaker as, $f^S_k$, where $k = 1, 2, \ldots, N$. Here, $N$ is the total number of feature vectors.

(4) For each feature vector $f^S_k$, given the GMM codebook for the target language $L_T$, the GMM-tokenizer outputs the mean vector ($\mu_k$) of the Gaussian mixture component scoring the highest in GMM likelihood computation as given below.

$$\mu_k = \arg \max_{i=1, 2, \ldots, M} \left[ w_i \cdot g \left( f^S_k | \mu_i, \Sigma_i \right) \right]$$  \hspace{1cm} (2)

where

$$g \left( f^S_k | \mu_i, \Sigma_i \right) = \frac{1}{\sqrt{(2\pi)^D|\Sigma_i|}} e^{-\frac{1}{2}(f^S_k-\mu_i)^T \Sigma_i^{-1}(f^S_k-\mu_i)}$$  \hspace{1cm} (3)

(5) Feature vectors of the source speaker, $f^S_k$, is now replaced by the target feature vectors (codeword), $\mu_k$, with the highest score (likelihood) ensuring the transformation of system features of the source speaker to that of the target speaker.

(6) The fundamental frequency ($F_0$) of the source speaker is transformed to that of the target speaker by a suitable $F_0$ modification factor $F_M$ given by,

$$F_M = \frac{F_T}{F_S}$$  \hspace{1cm} (4)

where $F_T$ is the average $F_0$ of target speaker computed from all the training utterances and $F_S$ is the average $F_0$ of the source speaker for a given utterance. The transformed spectral features and pitch period are given as input to synthesis filter/vocoder for synthesizing the transformed utterance.

3 Proposed Cross-Lingual VC Framework

Training neural network models for capturing target speaker-specific characteristics is not as straightforward as with the case of GMMs. The idea in building a neural network model
to capture speaker-specific characteristics is as follows [11]. Let $l_q$ and $s_q$ be two different representations of the speech signal from a target speaker $q$. While $l_q$ could be interpreted as speaker independent representation of speech signal, $s_q$ could be interpreted as carrying message and speaker information. A mapping function $\Omega (l_q)$ has to be built for transforming $l_q$ to $s_q$. Such a function would be specific to the speaker and could be considered as capturing the essential speaker-specific characteristics. The choice of representation of $l_q$ and $s_q$ is important in developing the mapping functions.

Figure 1 shows the block diagrams of the training and conversion modules of the proposed approach. In the proposed method, $l_q$ and $s_q$ are represented by deep auto-encoder (DAE) bottleneck features and MGCCs, respectively. The encoder part can cultivate the ability of speaker-independent encoding when a DAE is trained using spectral frames from multiple speakers [27]. As a result, the encoder can convert an observed frame into latent-variable (or bottleneck features in auto-encoder terminology) which contains information that is neutral to speaker, such as phonetic information. In training phase, first MGCCs are extracted only from target speaker’s utterances. Then, bottleneck features corresponding to these MGCCs are extracted by using a pretrained DAE. The DAE is a feedforward neural network which is trained using a multi-speaker corpus, to learn speaker-independent representations (described in Sect. 3.1). Finally, a DNN is trained using back propagation algorithm to minimize the error $||s_q' - s_q||^2$, where $s_q' = \Omega (l_q)$ and $\Omega (\cdot)$ is the mapping or conversion function. During conversion phase, the trained DNN model is employed to convert $l_r$ to $s_q'$, where $l_r$ represents the bottleneck features computed from signal signals of any arbitrary source speaker $r$.

3.1 Feature Extraction

3.1.1 MGCC Features

Speech is a non-stationary signal, and hence it is generally divided into a series of short-time successive overlapping frames for further processing [28]. Even in voice conversion, the features are transformed on a frame-by-frame basis. The number of frames ($N$) to be processed in a speech signal of $T$ sec duration is given by

$$N = \frac{T - L}{\Delta L} + 1$$

(5)

where $L$ is the length of each frame in seconds and $\Delta L$ is the overlapping time interval (also called frame shift) in seconds. Typically, a frame size of 25 ms and frame-shift of 5 ms is used in several speech processing applications. Every frame, depending on application is parameterized as a set of acoustic features. Spectral features like MGCCs are most suitable as acoustic features for VC task; because they can emphasize the speaker-specific properties contained in the speech signal. Hence, in this work, 34-dimensional MGCCs are extracted for each frame using WORLD vocoder [29], to capture the target speaker-specific information.

3.1.2 Deep Auto-Encoder Bottleneck Features

An Autoencoder (AE) is a feed forward neural network (FFNN) used to learn a representation (encoding) for a set of input data [30]. It consists of two blocks: Encoder and Decoder. In a simple auto-encoder having one-hidden-layer, the encoder maps a higher dimensional input vector $x$ to a lower dimensional feature vector $y$ as follows:

$$y = f_{\theta}(x) = s(Wx + b)$$

(6)
Here, $y$ is the bottleneck feature vector representation of the input feature vector $x$. $	heta = \{W, b\}$ is the encoder parameters. $W$ and $b$ are the weight matrix and a bias vector, respectively. $s$ is a non-linear activation function. The decoder reconstructs the input by using the output $y$ given by the encoder as follows:

$$z = g_{\theta'}(y) = s(W'y + b')$$

where $\theta' = \{W', b'\}$ is the decoder parameters. $s$ is a linear or non-linear activation function. The weight matrix $W'$ is usually constrained to be the transpose of the matrix in the encoder, i.e., $W' = W^\top$.

The AE parameters $\{\theta, \theta'\}$ are typically optimized using the mean squared error (MSE) criterion. The model parameters are usually estimated using RMSprop algorithm. An AE can be extended to a deeper architecture by stacking up multiple layers of encoders and decoders, which is called deep auto-encoder (DAE) [30]. The additional hidden layers enable the AE to learn mathematically more complex patterns in the data. Figure 2 shows the basic structure of DAE. In the encoding phase, the units at each hidden layer are calculated given its previous layer as

$$y^k = s\left(W^k y^{k-1} + b^k\right)$$

where $W^k$ and $b^k$ are the parameters of the $k$-th encoder layer, and $y^0 = x$. In the decoding phase, the hidden layers are calculated as

$$y^{k-1} = s\left(W^{kT} h^k + b'^{kT}\right)$$

and

$$z = W^{1T} y^1 + b'^{1'}$$

where $W^{kT}$ and $b'^{k'}$ are the parameters of the $k$-th decoder layer, $W^{1T}$ and $b'^{1'}$ are the parameters of the last decoder layer. The training criterion of DAE is the same as AE.

In this work, we have used DAE with (empirically arrived) architecture 512-512-M/2-512-512, where the encoder has 3 layers with {512, 512, M/2} units per layer and the decoder has 2 layers with {512, 512} units per layer. The bottleneck features correspond to the output of the last encoding layer (bottleneck layer), which typically contains a small number of neurons.
Table 1 Description of the speech database

| Language    | Male        | Female        |
|-------------|-------------|---------------|
|             | #Speakers   | #Train per speaker | #Test per speaker | #Speakers   | #Train per speaker | #Test per speaker |
| Telugu-DAE  | 3           | 125           | 30               | 3           | 125               | 30              |
| Telugu-VC   | 1 (TeM)     | 125           | 30               | 1 (TeF)     | 125               | 30              |
| Tamil       | 1 (TaM)     | 125           | 30               | 1 (TaM)     | 125               | 30              |
| Malayalam   | 1 (MaM)     | 125           | 30               | 1 (MaF)     | 125               | 30              |
relative to the size of the other layers. The input features for the DAE are $M$-dimensional MGCCs. The dimension of bottleneck layer is $M/2$ corresponding to half the dimension of MGCCs, and the dimension of output layer is $M$ corresponding to the dimension of input layer. Note that the dynamic features were not incorporated into the feature set. Sigmoid activation function is used for all the layers except for the last encoding layer, which has linear activation so that the produced bottleneck features could be real-valued. The DAE is trained under minimum MSE criterion using RMSprop optimizer. The learning rate is set to 0.001. After training, only the encoder part of DAE is retained for generating speaker-independent bottleneck features corresponding to MGCCs, for every frame. The feature extraction process is summarized as follows: Given a speech signal, we first divide it into frames (frame size = 25 ms and frame shift = 5 ms). Next, 34-dim MGCCs are extracted for each frame. Finally, the 17-dim bottleneck features corresponding to MGCCs of each frame are extracted by using DAE.

### 3.2 DNN for Feature Mapping

In this work, we used deep neural network (DNN) to capture the functional relationship between the $M/2$-dimensional bottleneck (input) features ($l_q$) and the $M$-dimensional MGCC (output) features ($s_q$) of the given target speaker data. The training samples $(s_{qi})_i^{N} \in X$ and $(l_{qi})_i^{N} \in X$ are the mel-cepstral coefficients and bottleneck features extracted from each frame of the target speaker’s speech data, respectively. The DNN model used in this paper is a four layer FFNN, and the (empirically arrived) final structure of the network is $(M/2)L \ 50N \ 50N \ ML$, where $L$ denotes a linear unit, and $N$ denotes a non-linear unit. The integer value indicates the number of units used in that layer. The non-linear units use sigmoid activation function. Prior to training, the input and output features are normalized to unit variance and zero mean. The weights of the network are adjusted using backpropagation learning algorithm to minimize the MSE for each pair of input-output features. The learning rate is 0.001, and the number of epochs is 25.

After training, a weight matrix is generated that represents the mapping function between input bottleneck features and output MGCCs. During conversion phase, the obtained weight matrix is used to transform bottleneck features ($l_r$) from any arbitrary source speaker $r$ to MGCCs ($s_q'$) of the desired target speaker (shown in Fig. 1b) on a frame-by-frame basis.

### 4 Experimental Evaluation

#### 4.1 Speech Corpus and Feature Extraction

For conducting the experiments we have considered the openslr multi-speaker databases from three Indian languages, namely, Telugu, Tamil and Malayalam. The openslr databases are available for free download at https://www.openslr.org/resources.php. For experiments, we chose 8 speakers (4 male and 4 female speakers) from Telugu language, 2 speakers (1 male and 1 female speakers) from Tamil language, and 2 speakers (1 male and 1 female speakers) from Malayalam language. Each speaker has 125 utterances for training and 30 utterances for testing. Data were recorded at 48 kHz, but we have downsampled to 16 kHz. The details of data considered for training and testing are given in Table 1. As shown in table, six out of the eight speakers (3 female and 3 male speakers) from Telugu dataset were considered for
training DAE and remaining speakers were used in VC experiments. The reason for choosing this database for DAE training is explained in the following section.

The WORLD vocoder [29] was used to extract speech parameters: $f_0$, aperiodicity (AP), and Spectral Envelope (SE). The frame length was 25 ms and the frame shift was 5 ms. The FFT length was set to 1024, so the resulting SE and AP were both 513-dimensional. 34-dimensional MGCCs plus log energy were derived from each spectral envelope. The dynamic features were not appended to the feature set. The proposed and existing VC models were trained using features corresponding to the target speaker alone. The trained models were then used to map the MGCCs of an arbitrary source speaker to the MGCCs of the target speaker. The transformed MGCCs were converted back to 513-dimensional SE. The $f_0$ of the source speaker was converted to that of the target speaker as in the baseline system (described in Sect. 2). The AP of source speaker was kept unmodified. Finally, all speech parameters were given as input to WORLD vocoder to synthesize the transformed utterance.

### 4.2 Training of Existing and Proposed CLVC Systems

In the baseline system, a GMM having 128 components is trained using 34-dimensional MGCCs from target speaker alone. The speaker-independent features are not required for a GMM-based system. The baseline system transforms voice of an arbitrary source speaker by replacing the source feature vectors with the mean vector of the Gaussian mixture component scoring the highest in GMM likelihood. In this work, a GMM model is built for every speaker from each language.

The training procedure of proposed VC system is different from that of the baseline system. Prior to training VC model, first a DAE is trained using speech utterances from 6 Telugu speakers to learn speaker-independent representations. The regional Indian languages considered are acoustically similar, to certain extent [31]. Hence, a common phoneset is derived by exploiting the acoustic similarities across the Indian languages [31]. In total, there are 39 phonemes in Tamil, 48 in Telugu, and 48 in Malayalam. It is observed that, 37 phones are common to all the languages. This intuitively shows that a DAE trained with data from one of these languages will be good enough to generate speaker-independent features for all the languages. Comparing Telugu and Malayalam, there are 47 phones in common. Comparing, Telugu and Tamil, there are 37 phones in common and 11 are unique to Telugu and 2 to Tamil. Similarly, comparing Malayalam and Tamil, there are 38 phones in common and 10 are unique to Malayalam and 1 is unique to Tamil. Considering the unique and common phones, either Malayalam or Telugu language is a better choice than Tamil language, for training DAE. Hence, we chose Telugu speaker data for training the DAE model. The encoder receives MGCCs computed from all the speakers and converts them to bottleneck features. The decoder reconstructs the input from the bottleneck features. The training procedure is terminated when there is no further improvement in terms of MSE for 15 epochs. The trained encoder is then utilized to extract bottleneck features for voice conversion. A DNN model which maps bottleneck features to MGCCs of a target speaker is built separately for all the considered speakers, except for those used during DAE training.

In addition, we have also considered the method proposed in [17] for comparison. This method uses a Deep Bidirectional Long Short-Term Memory Recurrent Neural Network (DBLSTM-RNN) for voice conversion. This approach requires PPGs of target speech. As in [17], PPGs are extracted using a SI-ASR system implemented using kaldi speech recognition toolkit [32] with openslr Telugu multispeaker corpus. Then, a DBLSTM-RNN is used to model the relationships between the PPGs and MGCCs of the target speech. The network has
Table 2  Average MCD scores obtained in case of intralingual voice conversion

| Source speaker | Target speaker | MCD (dB)       |
|----------------|----------------|----------------|
|                | Baseline       | Proposed       | DBLSTM-RNN |
| TeS (Female)   | TeF (Female)   | 6.226          | 3.513      | 4.462      |
| TeM (Male)     | TeF (Female)   | 7.511          | 3.861      | 4.717      |
| TaS (Female)   | TaF (Female)   | 6.221          | 3.782      | 4.533      |
| TaM (Male)     | TaF (Female)   | 7.506          | 4.089      | 4.966      |
| MaS (Female)   | MaF (Female)   | 6.237          | 3.572      | 4.509      |
| MaM (Male)     | MaF (Female)   | 7.518          | 3.943      | 4.776      |

TeS, TaS, and MaS indicates Telugu, Tamil, and Malayalam Female source speakers, respectively.

4 layers. The number of units in each layer is [48 128 128 34] respectively, where each hidden layer contains one forward LSTM layer and one backward LSTM layer. Back-propagation through time (BPTT) is used to train this model with a learning rate of $1.0 \times 10^{-6}$.

4.3 Results

4.3.1 Objective Evaluation

Mel-cepstral distortion (MCD) is used as an objective measure for evaluating the VC systems. MCD is calculated using the equation:

$$MCD(\text{dB}) = 10 \ln \frac{1}{10} \sqrt{\frac{2}{D_a} \sum_{d=1}^{D_a} (\hat{Y}_d - Y_d)^2}$$

where $D_a$ is the dimension of MGCCs, $Y_d$ and $\hat{Y}_d$ are the $d$th coefficients of corresponding target and converted MGCCs. A lower MCD value indicates that the predicted features are close to that of the target features. To compute MCD, we need reference speeches from target speaker corresponding to the converted speeches. Since the reference samples are generally not available for CLVC, we only report the objective evaluation results for intralingual VC. In the databases (Table 1), there is no overlap in the sentences uttered by both speakers from the same language. Therefore, to realize parallel data, we have recorded speech data from a native female voice talent for each language. The female voice talents act as source speakers for the purpose of objective evaluation. Each speaker has uttered 30 sentences in their native language. Among these, 15 sentences are from the test set of female speaker (Table 1) from the corresponding language. The remaining 15 are from the test set of male speaker. For example, the Tamil female voice talent has uttered total 30 sentences. Out of these, 15 sentences overlap with the test set of TaM and the remaining overlap with the test set of TaF. The goal is to evaluate the VC systems in the both cases of cross-gender and intra-gender conversion. The MCD scores are presented in Table 2.

For each source-target pair, only the 15 utterances from source speaker which have the corresponding reference speech from target speaker are converted. The target and converted feature sequences are time-aligned using dynamic time warping (DTW) algorithm. From the table, it can be seen that the proposed method outperforms baseline GMM-based CLVC system in intralingual VC. The MCD values obtained in cross-gender VC are slightly high compared to inter-gender VC, but are much less than those obtained with the GMM and
DBLSTM-RNN based CLVC systems. The results also indicate that the DAE bottleneck features are a good representation for speaker-independent information. Although DAE is trained using data from Telugu speakers alone, the encoder can perform speaker-independent encoding for languages that are acoustically closer. As MCD is not a representative indicator for perception, we further conducted subjective evaluations on speaker similarity and voice quality.

### 4.3.2 Subjective Evaluation

Performance of the CLVC systems is subjectively evaluated using two measures, namely, ABX preference test and Comparative Mean Opinion Score (CMOS). In preference tests, subjects were asked to listen to a pair of converted speech utterances and chose the one that is closer to natural speech in terms of similarity in voice. In CMOS test, subjects have to listen to two converted speech utterances (one from the proposed approach and another from one of the existing methods, played in random order) and then they have to rate the difference between the two samples on a 7-point scale ranging from much worse (−3) to much better (3). Subjects preference for proposed method over the baseline method and the opposite is indicated by a positive CMOS score and negative CMOS score, respectively. 15 listeners, including 5 Telugu, 5 Tamil and 5 Malayalam, participated in all the tests. The listening tests were conducted in the laboratory environment by playing the speech signals through headphones.

The results of ABX preference test and CMOS test comparing proposed and GMM-based systems for intra-lingual VC and cross-lingual VC are provided in Fig. 3. Similarly, the results comparing proposed and DBLSTM-RNN systems are shown in Fig. 4. To reduce
burden on listeners, we have considered only Malayalam Female as target and speakers from remaining languages as source, in cross-lingual VC. That is, the malayalam female speaker can now speak Tamil and Telugu. The trends in the results can be analyzed as follows: (i) The CMOS scores show that the proposed method provides a better voice quality compared to the GMM-based on intralingual VC, with a significance level of $p < 0.001$. The CMOS scores are further improved on cross-lingual VC. (ii) Although a trend seems to favor proposed method compared to baseline in speaker similarity, the difference is not significantly different. The listeners reported that there is muffledness and more distortion in the speech converted with baseline GMM-based system. (iii) The preference test results (Fig. 4) show that both DBLSTM-RNN and proposed systems perform equally well in terms of speaker similarity. (iv) The CMOS scores in Fig. 4 indicate that perceptually the proposed method is better than DBLSTM-RNN system on intralingual and cross-lingual VC. The improved perceptual quality with proposed method is mainly due to better acoustic feature mapping. The results also indicate that the proposed approach can render high-quality speaker conversion irrespective of languages. Hence, our system is capable of addressing the issues associated with unaligned data sets.

5 Summary and Conclusion

In this paper, we have presented a DAE-based CLVC approach for transforming a source speaker’s speech to sound as if it was uttered by a target speaker who is oblivious of the source language. The DAE is trained with data from multiple speakers to learn speaker-independent representations. Even though data from only one language is used to train DAE, the encoder performs robustly across acoustically closer languages. To build VC model for a given target
speaker, first the MGCCs are passed through the encoder to derive bottleneck features. Then, a DNN is trained to predict MGCCs of target speaker from bottleneck features. The proposed approach can map spectral features of any arbitrary source speaker onto a target speaker’s acoustic space. Hence, the proposed method can be considered as a “many-to-one mapping” method. The performance of the CLVC systems is evaluated using three acoustically similar Indian languages. The results of subjective evaluation confirm that both quality and target speaker similarity of converted speech from proposed CLVC system are much better than that of existing CLVC systems. In future, we plan to utilize the proposed CLVC technique to develop a polyglot SPSS system for Indian languages.

References

1. Stylianou Y et al (1998) Continuous probabilistic transform for voice conversion. IEEE Trans Speech Audio Process 6(2):131–142
2. Toda T et al (2007) Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory. IEEE Trans Audio Speech Lang Process 15(8):2222–2235
3. Takamichi S et al (2015) Modulation spectrum-constrained trajectory training algorithm for GMM-based voice conversion. In: Proceedings of ICASSP, South Brisbane, Queensland, Australia, pp 4859–4863
4. Yu J, Tao D, Wang M, Rui Y (2015) Learning to rank using user clicks and visual features for image retrieval. IEEE Trans Cybern 45(4):767–779
5. Hong C, Yu J, Wan J, Tao D, Wang M (2015) Multimodal deep autoencoder for human pose recovery. IEEE Trans Image Process 24(12):5659–5670
6. Yu Z, Yu J, Xiang G, Fan J (2018) Beyond bilinear: generalized multimodal factorized high-order pooling for visual question answering. IEEE Trans Neural Netw Learn Syst 29(2):5947–5959
7. Hong C, Yu J, Zhang J, Jin X, Lee K (2019) Multimodal face-pose estimation with multitask manifold deep learning. IEEE Trans Ind Inf 15(7):3952–3961
8. Zhang J, Yu J, Tao D (2018) Local deep-feature alignment for unsupervised dimension reduction. IEEE Trans Image Process 27(5):2420–2432
9. Yu J, Tan M, Zhang H, Tao D, Rui Y (2019) Hierarchical deep click feature prediction for fine-grained image recognition. IEEE Trans Pattern Anal Mach Intell. https://doi.org/10.1109/TPAMI.2019.2932058
10. Yu J, Zhu C, Zhang J, Huang Q, Tao D (2019) Spatial pyramid-enhanced netvlad with weighted triplet loss for place recognition. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2019.2908982
11. Desai S et al (2010) Spectral mapping using artificial neural networks for voice conversion. IEEE Trans Audio Speech Lang Process 18(5):954–964
12. Mohammadi SH, Kain A (2014) Voice conversion using deep neural networks with speaker-independent pre-training. In: Proceedings of SLT, pp 19–23
13. Saito Y, Takamichi S, Saruwatari H (2017) Voice conversion using input-to-output highway networks. IEICE Trans Inf Syst E100–D(8):1925–1928
14. Sun L et al (2015) Voice conversion using deep bidirectional long short-term memory based recurrent neural networks. In: Proceedings of ICASSP, South Brisbane, Queensland, Australia, pp 4869–4873
15. Chen LH et al (2014) Voice conversion using deep neural networks with layer-wise generative training. IEEE/ACM Trans Audio Speech Lang Process 22(12):1859–1872
16. Nakashika T et al (2015) Voice conversion using RNN pre-trained by recurrent temporal restricted Boltzmann machines. IEEE/ACM Trans Audio Speech Lang Process 23(3):580–587
17. Sun L, Wang H, Kang S, Li K, Meng HM (2016) Personalized, cross-lingual TTS using phonetic posteriorgrams. In: Proceedings of INTERSPEECH
18. Hsu C-C et al (2017) Voice conversion from unaligned corpora using variational autoencoding wasserstein generative adversarial networks. arXiv:1704.00849
19. Kameoka H et al (2018) StarGAN-VC: non-parallel many-to-many voice conversion with star generative adversarial networks. arXiv:1806.02169
20. Fang F et al (2018) High-quality nonparallel voice conversion based on cycle-consistent adversarial network. In: Proceedings of ICASSP, Alberta, Canada, pp 5279–5283
21. Abe M et al (1990) Cross-language voice conversion. In: Proceedings of ICASSP, Albuquerque, New Mexico, USA, pp 345–348
22. Sundermann D, Ney H, Hoge H (2003) VTLN-based cross-language voice conversion. In: IEEE workshop on automatic speech recognition and understanding, pp 676–681
23. Erro D, Moreno A (2007) Frame alignment method for crosslingual voice conversion. In: Proceedings of Interspeech 2007, Antwerp, Belgium, pp 1969–1972
24. Charlier M et al (2009) Cross-language voice conversion based on eigenvoices. In: Proceedings of INTERSPEECH, Brighton, UK, pp 1635–1638
25. Ramani B et al (2014) Cross-lingual voice conversion-based polyglot speech synthesizer for Indian languages. In: Proceedings of INTERSPEECH, Singapore, pp 775–779
26. Zhou Y et al (2019) Cross-lingual voice conversion with bilingual phonetic posteriorgram and average modeling. In: Proceedings of IEEE international conference on acoustics, speech and signal processing (ICASSP), Brighton, UK, pp 6790–6794
27. Hsu CC et al (2016) Voice conversion from non-parallel corpora using variational auto-encoder. In: Proceedings of Asia-Pacific signal and information processing association annual summit and conference (APSIPA), pp 1–6
28. Rabiner L, Juang B-H (1993) Fundamentals of speech recognition. Prentice-Hall, Englewood Cliffs, NJ
29. Morise M et al (2016) WORLD: a vocoder-based high-quality speech synthesis system for real-time applications. IEICE Trans Inf Syst E99–D(7):1877–1884
30. Hu Y, Ling Z (2018) Extracting spectral features using deep autoencoders with binary distributed hidden units for statistical parametric speech synthesis. IEEE/ACM Trans Audio Speech Lang Process 26(4):713–724
31. Ramani B et al (2013) A common attribute based unified HTS framework for speech synthesis in Indian languages. In: Proceedings of SSW, Barcelona, Spain, pp 291–296
32. Povey D et al (2011) The Kaldi speech recognition toolkit. In: IEEE Workshop on Automatic Speech Recognition and Understanding. IEEE Signal Processing Society

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