Voice conversion using deep neural networks

Nguyen, Quy Hy

2017

Nguyen, Q. H. (2017). Voice conversion using deep neural networks. Master's thesis, Nanyang Technological University, Singapore.

<http://hdl.handle.net/10356/72102>

<https://doi.org/10.32657/10356/72102>

VOICE CONVERSION USING DEEP NEURAL NETWORKS

A thesis submitted to the School of Computer Engineering
of the Nanyang Technological University

by

NGUYEN QUY HY

for the Degree of Master of Engineering

February 27, 2017

Abstract

This thesis focuses on techniques to improve the performance of voice conversion. Voice conversion modifies the recorded speech of a source speaker towards a given target speaker. The resultant speech is to sound like the target speaker with the language content unchanged. This technology has been applied to create personalized voice in text-to-speech or virtual avatar, speech-to-singing synthesis or spoofing attacks in speaker verification systems.

To perform voice conversion, the usual approach is to create a conversion functions which is applied on the source speaker's speech features such as timbre and prosodic features, to generate the corresponding target features. In this past decade, most of voice conversion researches had focused on spectral mapping, i.e. conversion of the features representing the timbre characteristics in a frame by frame manner.

In chapter 3, we investigate a comprehensive approach to train the conversion function using DNN which considers both timbre and prosodic features simultaneously. For better modelling, we have used high-dimension spectral features. However, this further worsen the ability to robustly train a DNN which typically requires large amount of training data. To overcome the issue of limited training data, we propose a new pretraining process using autoencoder. The experimental results show the proposed comprehensive framework with pretraining performs better than conventional voice conversion systems including the state-of-the-art GMM-based system.

The technique introduced in chapter 3 only learns a DNN system to convert between a pair of speaker. To reduce the need for parallel training data of new speaker pair, in chapter 4 we examine a novel DNN adaptation technology for voice conversion by including two bias vector representing both source and target speaker. By this configuration, new speaker pair conversion are archived. Our preliminary results show that conversion to new target speakers' voices could be achieved.

Acknowledgments

My deepest gratitude goes first to my supervisor, Dr. Chng Eng Siong, who has been guiding me through out my University education and giving significant help on the completion of this Master study.

I am also very grateful to my colleagues, Yvonne Lee Siu Wa and Tian Xiaohai, who has helped me tremendously throughout my Master study. Your enthusiasm and generosity make my journey in speech research not only fruitful, but also very enjoyable. And I want to thank all my colleagues in the speech group in NTU for all the help, discussions and suggestions during my study.

Contents

Abstract	i
Acknowledgments	ii
List of Figures	v
List of Tables	vi
Lists of Publications	vii
 1 Introduction	 1
1.1 Motivation	2
1.2 Contribution and Future Directions	3
1.3 Organization of Dissertation	3
 2 Introduction to Voice Conversion	 4
2.1 Voice Conversion Techniques	5
2.1.1 Spectral Transformation	6
2.1.1.1 Vector Quantization Based	6
2.1.1.2 GMM Based	6
2.1.1.3 Neural Network Based	8
2.1.2 Prosody Transformation	9
2.2 Summary	10
 3 Comprehensive DNN-based Voice Conversion Framework	 11
3.1 Introduction	11
3.2 Contributions	12
3.3 Framework	13
3.3.1 Spectrum Conversion	13

3.3.2	F0 Trajectory Conversion	17
3.3.3	Intensity Trajectory and Phone Duration Conversion	19
3.4	Experiment Setup	20
3.5	Evaluation Results	21
3.5.1	Spectrum Conversion	21
3.5.1.1	Objective evaluation	22
3.5.1.2	Subjective evaluation	24
3.5.2	Spectrum and F0 Conversion	25
3.5.2.1	Objective evaluation	26
3.5.2.2	Subjective evaluation	26
3.5.3	The Comprehensive Framework: Spectrum, F0 and Intensity and Duration Conversion	28
3.6	Conclusion	29
4	DNN Adaptation for Voice Conversion	31
4.1	DNN Adaptation	31
4.2	Contributions	32
4.3	System Overview	33
4.4	Conclusion	36
5	Conclusion	37

List of Figures

2.1	General structure of a typical voice conversion system	5
2.2	General structure of a Multilayer Perceptron (MLP) where neurons in each layer take inputs from the immediate previous layer and output to the immediate next layer. There is no backward connection nor inter-layer connection.	8
3.1	Comprehensive Voice Conversion Framework	14
3.2	Performing the forward-pass with the DNNs in different stages to demonstrate the effectiveness of autoencoder pretraining. A: spectrogram from source speech; B: spectrogram from target speech; C: output spectrogram from DNN with randomly initialized parameters; D: output spectrogram from DNN after autoencoder pretraining; E: output spectrogram from DNN after model training.	16
3.3	Re-sample spectral segment to create input vector for duration modeling	20
3.4	Outputs of F0 conversion from three methods: Mean-Var, DNN-Frame and DNN-Segment	27
3.5	DNN-based Voice Conversion	29
4.1	DNN architecture of VoiceProducer.	34
4.2	Training and runtime process of VoiceProducer.	35
4.3	User interface of VoiceProducer.	36

List of Tables

3.1	Comparison of LSD ratio of different spectral conversion methods	23
3.2	Results of average quality and similarity preference tests with 95% confidence intervals of different spectral conversion methods	25
3.3	Comparison of RMSE of different F0 conversion methods	26
3.4	Results of average similarity preference tests with 95% confidence intervals of different F0 conversion methods	27
3.5	Results of average similarity preference tests with 95% confidence intervals of the proposed intensity and duration conversion method	28

Lists of Publications

- **Hy Quy Nguyen**, Siu Wa Lee, Xiaohai Tian, Minghui Dong, Eng Siong Chng, "High quality voice conversion using prosodic and high-resolution spectral features", Multimedia Tools and Applications, Volume 75 Issue 9, May 2016 [1].
- Siu Wa Lee, **Hy Quy Nguyen**, Kong Aik Lee, Xiaohai Tian, Longting Xu, Eng Siong Chng, Minghui Dong, Haizhou Li, Voiceproducer: Many-To-Many Voice Conversion For Arbitrary Speakers, ICASSP (show and tell) Mar 2016 [2].
- Lei Meng, **Hy Quy Nguyen**, Xiaohai Tian, Zhiqi Shen, Eng Siong Chng, Frank Yunqing Guan, Chunyan Miao, Cyril Leung, "Towards Age-friendly E-commerce Through Crowd-improved Speech Recognition, Multimodal Search, and Personalized Speech Feedback", in Proceedings of the 1st International Conference on Crowd Science and Engineering (ICCSE16), July 2016 [3].
- Xiaohai Tian, Zhizheng Wu, Siu-Wa Lee, **Nguyen Quy Hy**, Minghui Dong, Eng Siong Chng, "System Fusion for High-Performance Voice Conversion", Interspeech 2015 [4].

Chapter 1

Introduction

Speech is arguably the most common and effective form of human communication. While human speech contains both language content and speaker identity, this research focuses on speaker identity, particularly transforming the speaker identity in recorded speech from a source to a target speaker. This field of research is called voice conversion and can open up a range of applications to improve human-machine interaction and speaker verification systems.

According to speech perception research [5, 6], speaker identity is characterized by the following factors,

- a) **Segmental factor (or timbre):** This is a type of acoustic characteristics, which relates to short-term features, such as spectrum, formants, and the shape of the glottal excitation pulse [7].
- b) **Supra-segmental factor (or prosody):** This is prosodic characteristics and is also a kind of acoustic characteristics. It involves the duration of phonemes, syllables and words; the evolution of fundamental frequency (intonation); stress; tone.
- c) **Linguistic factor:** This is relevant to the language content or the lexical message expressed by the speaker, and has the same information as presented in text format. Each speaker's identity is characterized by his/her own lexicon, and the grammatical choice of words and sentence structure the speaker tends to use in his/her conversations.

In speech research, given a fixed language content, the supra-segmental factor and segmental factor, also called *acoustic* factors, are considered the most relevant to speaker identity. Between the two factors, the segmental factor is believed to contain more individuality information [6], and hence gets much more attention in voice conversion research. In this regards, the objective of this research is to develop a novel voice conversion technique by applying conversion on both segmental and supra-segmental speech features and by addressing challenges in training voice conversion system with limited amount of training data.

1.1 Motivation

Voice conversion technology has many applications. First of all, it can be used to create personalized Text-to-speech (TTS) system. Current speech synthesis techniques require a large number of sample utterances from the intended speaker to create a similar voice with good speech quality; thus it is quite expensive to build a personalized TTS voice. Besides, acquiring a large number of sample utterances could be impossible in several situations, such as for people with speaking impairment or larynx disorders who want to retain their voices. In those cases, with the lower requirement in the amount of training data, voice conversion can be used to establish a mapping between a high quality TTS voice into a personalized voice. Integrated with facial animation techniques, voice conversion can be used to create virtual avatar, story teller, or assistant for e-commerce, e-learning, etc. Secondly, voice conversion has been used to hide true speaker identity and to impose certain speaker identity to attack speaker verification system [8]. Finally, techniques in voice conversion also have other applications in speech-to-singing synthesis [9] and speech enhancement [10, 11].

Several techniques have been used to estimate the mapping between source and target speaker's characteristics. Among them, deep neural networks (DNNs) with its ability to model complex non-linear functions yield as a very potential way to perform voice conversion. Despite promising results of DNN applications such as image recognition, machine translation and speech recognition, the performance of DNN in voice conversion is still limited due to the relatively small amount of training data. This motivates us to develop

techniques to improve training performance while maintaining the non-linearity advantage of using large DNN. Also, this work explores the capability of DNN in capturing suprasegmental information for prosodic features in voice conversion.

1.2 Contribution and Future Directions

In this thesis, we presented two contributions towards improving voice conversion techniques:

The first contribution focus on improving voice conversion quality by using DNN to predict both spectral and prosodic features for the target speaker. Specifically, our technique overcome the over-smoothing problem of low dimensional input features by using DNN which is capable to model with high dimensional input. Additionally, to alleviate the limited training data problem for DNN training, a pre-training technique using auto-encoder is applied. Finally, we introduce a comprehensive framework to generate both spectral and prosodic features to improve the quality of generated target speaker's speech. The work has been published in [1] Multimedia Tools and Applications journal and [3].

The second contribution introduced a novel idea of using DNN adaptation to remove the need of parallel data for new source and target speaker pairs during testing. This is achieved by including the source and target speaker identities as bias vectors into the input feature during training and conversion. This is a relatively well known technique in acoustic model adaptation of ASR. This work has been published in ICASSP2016 show and tell [2] with a demo to show one-many, many-one and many-many speaker pair conversion [4].

1.3 Organization of Dissertation

The thesis is organized as follows: Chapter 2 provides some background on voice conversion and reviews the existing state-of-the-art techniques. Chapter 3 proposes a DNN-based voice conversion framework and evaluation results on the proposed framework. Chapter 4 introduces a DNN adaptation framework to perform one-many, many-one and many-many speaker pair conversion. Finally, chapter 5 concludes the report and discusses the future research directions.

Chapter 2

Introduction to Voice Conversion

Voice conversion converts one speaker's voice to another's. Different from model adaptation approaches in text-to-speech, voice conversion operates directly on speech features using a linear or non-linear mapping. State-of-the-art voice conversion technique operates on frame level, and the primary module of voice conversion system is a conversion function whose task is to modify the source feature frames into target feature frames. The conversion function can be formulated as follows,

$$\mathbf{y} = \mathbf{F}(\mathbf{x}) \quad (2.1)$$

where $\mathbf{x}$ and $\mathbf{y}$ is the source and target feature vector, and $\mathbf{F}(\cdot)$ is the mapping from source to target characteristics.

In typical voice conversion techniques, two phases are involved: offline training and conversion. Offline training phase is to estimate the conversion function from recorded speech, and conversion phase applies the estimated function on testing source data to generate speech with target characteristics. The process of the two phases are demonstrated in Figure 2.1.

In the offline training phase, recorded speech of source and target speaker with the same language content, also known as parallel utterances, are used as inputs. Those training data are analyzed by a speech production model [12] to generate independent components such as spectrum, excitation and fundamental frequency (F0). A vocoder is usually used in this step to transform recorded speech signal into stable feature frames. Several vocoders such as MCEP in SPTK [13] or STRAIGHT [14] are commonly employed

Figure 2.1: General structure of a typical voice conversion system

in previous voice conversion systems. After that, for each parallel utterance, the source and target feature frames are paired up by frame alignment techniques such as dynamic time warping (DTW). Finally, the conversion function $\mathbf{F}(\cdot)$ is estimated from the aligned feature frames.

In the conversion phase, testing source speech is passed through the same speech analyzer module as in the training phase to obtain speech features. Those features are then modified using the conversion function estimated in training to generate target features $\hat{\mathbf{y}} = \mathbf{F}(\mathbf{x})$. Finally, the generated target features are input to a speech synthesizer module to reconstruct the resultant speech signal.

Each module faces its own challenges. This research focuses mainly on improving the pattern learning module and conversion function to enhance the performance of voice conversion system.

2.1 Voice Conversion Techniques

Most voice conversion techniques focus on spectral transformation and a few others work on prosody transformation. This section reviews some of the techniques that has been introduced for voice conversion. Essentially, all of them follow the structure in Figure 2.1 with some variations on the type of speech features and conversion functions.

2.1.1 Spectral Transformation

2.1.1.1 Vector Quantization Based

As voice conversion operates on feature vector, a classical technique to perform mapping from source to target would be vector quantization (VQ). VQ, particularly joint-vector VQ, has been used for spectral transformation in [15].

During the training phase, the time-aligned source x and target vector y are stacked into a joint-vector z . The joint vector is in turn used to build a codebook containing a set of joint-vector centroids. During the conversion phase, each source feature vector is paired up with one centroid in the codebook that has the nearest source component. The target component of that centroid is used as the converted feature vector corresponding to the input source vector. All the converted vectors are then concatenated to create the final converted feature.

When every joint source-target vector is used as a codebook centroid, VQ methods have the advantages of retaining all the spectral details because the converted feature vector is obtained directly from natural speech. However, due to the hard-clustering nature of the approach, VQ methods tend to create incoherence phenomenon across adjacent frames in converted feature [16, 17].

One variation of the VQ approach is sparse coding, where the each converted feature vector is not obtained directly from target vectors in the training set, but as a linear combination of a limited number (or sparse set) of target vector. In this approach, all the training joint vectors are kept in a dictionary, and the weights for linear combination is estimated dynamically for each input source vector at runtime. This approach has been successfully demonstrated with the application of Non-negative Matrix Factorization (NMF) algorithm in [18].

2.1.1.2 GMM Based

To avoid the incoherent problem from hard clustering, Gaussian mixture model (GMM) based methods were introduced in [16, 17, 19]. Among those, joint-density Gaussian mixture model (JD-GMM) [16, 17] is one of the most successful methods. In this method, GMM is estimated to model the joint distribution of source feature vectors $\mathbf{X}$ and target

feature vectors $\mathbf{Y}$. During training, expectation-minimization (EM) algorithm is used to maximize the joint distribution as below:

$$P(\mathbf{X}, \mathbf{Y}) = P(\mathbf{Z}) = \sum_{k=1}^K w_k \mathcal{N}(\mathbf{z} | \boldsymbol{\mu}_k^{(z)}, \boldsymbol{\Sigma}_k^{(z)}) \quad (2.2)$$

$$\mathbf{z} = \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix}, \quad \boldsymbol{\mu}_k^{(z)} = \begin{bmatrix} \boldsymbol{\mu}_k^{(x)} \\ \boldsymbol{\mu}_k^{(y)} \end{bmatrix}, \quad \boldsymbol{\Sigma}_k^{(z)} = \begin{bmatrix} \boldsymbol{\Sigma}_k^{(xx)} & \boldsymbol{\Sigma}_k^{(xy)} \\ \boldsymbol{\Sigma}_k^{(yz)} & \boldsymbol{\Sigma}_k^{(yy)} \end{bmatrix} \quad (2.3)$$

where $\mathbf{Z}$ is the joint paired feature vector sequence, K is the number of Gaussian components, w_k is the weight of the k -th component and $\mathcal{N}(\mathbf{z} | \boldsymbol{\mu}_k^{(z)}, \boldsymbol{\Sigma}_k^{(z)})$ is the Gaussian distribution with mean $\boldsymbol{\mu}_k^{(z)}$ and covariance matrix $\boldsymbol{\Sigma}_k^{(z)}$ of the k -th component.

During conversion, from each source vector $\mathbf{x}$, a converted vector $\hat{\mathbf{y}}$ is estimated from the trained Gaussian models using minimum mean square error criteria as follows:

$$\hat{\mathbf{y}} = F(\mathbf{x}) = \sum_{k=1}^K p_k(\mathbf{x}) [\boldsymbol{\mu}_k^{(y)} + \boldsymbol{\Sigma}_k^{(yx)} (\boldsymbol{\Sigma}_k^{(xx)})^{-1} (\mathbf{x} - \boldsymbol{\mu}_k^{(x)})] \quad (2.4)$$

$$p_k(\mathbf{x}) = \frac{w_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k^{(x)}, \boldsymbol{\Sigma}_k^{(xx)})}{\sum_{k=1}^K w_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k^{(x)}, \boldsymbol{\Sigma}_k^{(xx)})} \quad (2.5)$$

where $p_k(\mathbf{x})$ is the posterior probability of the source vector $\mathbf{x}$ generated from the k -th Gaussian component.

Essentially, this is similar to the joint-vector VQ approach introduced in Section 2.1.1.1, except that the hard clustering is replaced by the soft clustering nature of Gaussian models. This helps to create a smooth transition between adjacent frames in converted results, which better resembles features from natural speed.

However, this technique also has its own shortcomings. First of all, GMM techniques tend to work better with low-dimensional feature, which indicates a trade-off between highly detailed feature and good modeling performance. Secondly, due to the averaging nature of GMM, the trajectory of the resulting speech feature is usually over-smooth [18]. This problem has been partly solved by the maximum likelihood trajectory generation algorithm [16]. Moreover, global variance (GV) has been included in GMM-based conversion as a compensation for over-smooth result [20].

Figure 2.2: General structure of a Multilayer Perceptron (MLP) where neurons in each layer take inputs from the immediate previous layer and output to the immediate next layer. There is no backward connection nor inter-layer connection.

2.1.1.3 Neural Network Based

In the recent years, deep neural networks have been used to replace GMMs in modeling feature transformation with great success [21, 22]. A deep neural network is network of a large number of neuron, ranging from hundreds to several thousands. Each neuron in the network is a non-linear transformation from an input vector into a number as follows:

$$\hat{y} = F(\mathbf{x}) = \mathbf{G}(\mathbf{W} * \mathbf{x} + b) \quad (2.6)$$

where $\mathbf{W}$ is a weight vector, b is a bias scalar and $G(\cdot)$ is a non-linear function such as sigmoid, hyperbolic tangent, rectified linear, etc. To facilitate complex modeling, multiple neuron are connected in a layer-by-layer manner as demonstrated in Figure 2.2. This network architecture is known as Multilayer Perceptron (MLP) [23].

During training, parameters $\mathbf{W}$ and b of each neuron are estimated to optimize the error between the expected output y and the actual output $\hat{y}$. When there is multiple layers of neurons as in MLP, the error is propagated from the output layer back to all previous layers through an optimization process called back-propagation[24]. In particular, pairs of training input and output feature vectors are presented to the network. An estimated output are calculated from the training input and current weight and bias parameters. After that, an error is calculated by comparing the estimated output and

the actual training output and the parameters of the last neuron layer are updated accordingly. The error is then propagated back to previous neuron layers to update the parameters of those layers.

Comparing to GMM, DNN models can have a much larger number of parameters and can model a much more complex transformation from source to target speaker. Moreover, the back-propagation training algorithm enabling modeling directly on high-dimensional feature. [25] and [26] showed that using either low-dimensional feature and high-dimensional feature, DNN outperform GMM on converted speech quality.

2.1.2 Prosody Transformation

While spectral mapping seeks to change the timbre, prosody transformation aims at modifying prosodic features such as fundamental frequency (F0), intensity and duration.

Most of the work on prosody transformation focus on F0. The most common technique to convert F0 is normalizing the global mean and variance of F0 of source speaker into those of target speaker [19]: at runtime, the instantaneous F0 of target speaker in converted speech is calculated as a linear transformation of the input instantaneous F0 of source speaker as follow,

$$\hat{F0}_y = (F0_x - \mu_{F0_x}) * \frac{\sigma_{F0_y}}{\sigma_{F0_x}} + \mu_{F0_y} \quad (2.7)$$

where $F0_x$ is the instantaneous F0 of source speaker, μ_{F0_x} and μ_{F0_y} are the means of all training F0 from source and target speakers respectively, σ_{F0_x} and σ_{F0_y} are the variances of all training F0 from source and target speakers respectively and $\hat{F0}_y$ is the estimated instantaneous F0 of target speaker.

Thanks to the small number of parameters (i.e. global mean and variance), this technique can be applied with very small amount of training data. Moreover, as the statistics are calculated without the need to pair up source-target frames, the technique can be applied on both parallel and non-parallel training sets. On the other hands, few number of parameters limit the ability of the technique to model complex transformation in F0 trajectory between source and target speakers.

Some other methods on F0 transformation include vector quantization [27], partial least square regression [28] and DNN mapping [29]. Those methods estimate a larger

number of parameters in order to model more complex F0 transform. Among those, [29] operates on wavelet domain instead of the frequency domain directly, allowing certain control on the level of details of converted result.

Other prosody characteristics such as duration and intensity have been investigated in expressive speech synthesis and voice conversion. Representative works include state duration modification using decision tree[30], duration modification based on HMM model [31], duration embedded GMM-based conversion [32] and mean-variance transformation for phone or utterance duration [33]. Most of these methods use phone boundaries or phone identities from text labels. In [32], consecutive spectral frames are used for duration modeling. Studies have also suggested some ways to perform intensity conversion based on F0 to improve naturalness [34].

2.2 Summary

In this chapter, the concept and general structure of voice conversion have been introduced. State-of-the-art techniques on spectral conversion and prosody conversion have also been discussed, which are also the baselines and give background knowledge for our work in Chapter 3.

Chapter 3

Comprehensive DNN-based Voice Conversion Framework

The DNN model has been shown to be a very versatile system which can learn the mapping between input and output features. In this chapter, we examine the application of DNN for the voice conversion task in a comprehensive framework to model four target speaker's features, specifically the spectral features, F0, duration and intensity. Such a comprehensive framework to the best of our knowledge has not been studied. Our experimental results conducted on the CMU ARCTIC database validated the effectiveness of the proposed framework.

The chapter is organized as follows. Section 3.1 reviews existing works related to neural network based voice conversion. In Section 3.2, we point out the contributions of our work. In the Section 3.3, we present a detailed description of the proposed framework and give the , and in Section 3.4 we summarize our conclusions.

3.1 Introduction

Recently, DNN-based techniques have been well very well received in the speech research community [35, 36, 37, 38] for works related to acoustic model, language modelling as well as voice conversion. This is primarily due to the ability of DNN to learn non-linear mapping between source and target features with little restriction in the features' dimensions.

An early work of DNN for voice conversion [25] had performed spectral conversion by learning the formant transformation between source and target speaker using a low-

dimensional mel-cepstral feature. Although the DNN introduced was a simple feed forward network, it nevertheless showed improved performance over conventional GMM modelling framework for the task. In a similar work, [39] used a Deep Belief Nets (DBNs) model to extract the latent features of low dimensional source and target cepstrum coefficients to perform conversion. One of major issues in voice conversion is over-smoothing due to the use of low-dimensional features. Over-smoothing causes loss of speech details, and in turns leads to degrade in both quality and similarity of converted speech. Recently, high-dimensional spectral features [26] which can be used in the DNN framework achieved significant improvement to conversion quality.

Earlier voice conversion works had focused on the modelling of spectral features as it is the most representative of speech from a target speaker. However, the identity of a speaker is also influenced by the prosodic features, e.g, F0, duration and intensity. This has prompted recent works to include source to target conversion of F0 [29], duration modelling of phone or utterance using HMM and GMM [30, 31], state duration modification using decision tree [33] and duration embedded GMM-based conversion [32]. Lastly, intensity conversion based on F0 to improve naturalness [28] was also examined.

3.2 Contributions

To the best of our knowledge, the generation of the four features (spectral, intensity, D0, and duration) using a common set of input has not been examined in a common framework. Our novel contribution in this thesis is to demonstrate the use of DNN for this task.

- (i) Our DNN learns the conversion of spectral and prosodic features jointly.
- (ii) For the prediction of F0, we have exploited segmental information [26] as well as configuring our model to predict the derivative of F0 to emphasize F0's movement over time. This is slightly different to conventional approaches which models absolute values. Our experimental results show that this approach improves the DNN learning as well as the generation of F0 trajectories.
- (iii) As traditional DNN requires large amount of data, which we do not have in this task, we applied a pre-training scheme based on autoencoder to prevent overfitting.

The above proposed framework has been examined in details. The conversion performance delivered by various features was verified. In particular, the effectiveness of our autoencoder pretraining was confirmed and improved performance in objective measurement was achieved. The proposed framework showed the highest rating on speaker similarity over state of the art voice conversion methods.

3.3 Framework

The proposed framework is demonstrated in Fig. 3.1. In this work, the four features, spectral, F0 contour, intensity, and phone duration are modeled using feed-forward DNNs. As spectral features must be considered at the frame-level, while the prosodic features should be considered at segment-level and are of different nature to each other, individual DNN have been used for each input feature.

During training, the parallel speech features between all source and target utterances, namely - spectral features, F0 and duration, are first extracted. These two feature sequences are then aligned using Dynamic Time Warping to generate parallel features at the frame level. For the intensity and duration prosodic features, multi-frame of the source features are packed into appropriate input vectors to capture supra-segmental information. These prepared features are later presented to the DNN randomly for training. Each DNN is trained by the back-propagation algorithm with mean square error criteria.

During testing, the converted features from each DNN are combined to generate the target speech. Our system exploits the generated target speaker’s prosodic information of intensity and duration to modify the spectral features.

The details of the four DNN model in the framework are introduced below.

3.3.1 Spectrum Conversion

This section describes the DNN used to model the high dimensional source to target spectral feature transformation. The spectral feature used in this study is 512 in dimension. The input to the DNN is a 512×3 dimension feature make up by the 512 static feature and its delta and double delta feature. The output is the corresponding aligned

Figure 3.1: Comprehensive Voice Conversion Framework

target spectral feature frame. Our spectrum conversion process follows closely [26] with the following two modifications to improve performance.

Two-stage Alignment

To generate accurate frame alignment, a two-stage alignment [9] is employed in our system. For each pair of parallel source and target utterance, a phone recognizer is first employed to locate the phone boundaries at each utterance. The phone sequences between the source and target is then aligned. These aligned phones are then presented to a DTW to obtain the aligned source to target spectral feature at the frame level.

Autoencoder Pretraining

This sub-section describes the use of autoencoder training technique to pre-train the DNN. This is to reduce the overfitting problem that may occur in the DNN when used to model the high dimensional output spectral features. Overfitting can easily occur as the size of DNN can be extremely large - the number of nodes in the network can be in the range of hundreds of thousands, while the number of training features are only a fraction of the number of nodes. Researchers [22, 40, 41] have shown that pre-training techniques can improve the robustness of DNN training significantly under such conditions.

In this work, we employed the process of autoencoder training to pretrain our DNN. An auto-encoder as its name implies learn the mapping between its input to its input though a deep network. One of its use is to extract its middle hidden layer which is usually more narrow to extract embedded features that are typically more representative. By using all our available training data, we first pre-train the DNN by the auto-encoder training. The DNN at this stage now has learn the mapping of its input to itself. This is now very close to our desired task of mapping the input source feature to target. We will apply L1 sparsity constraint on the DNN weights [42]. Different to traditional auto-encoder structure, our DNN network structure is not narrow in the middle layer. Instead, our current network architecture has three layers with each layer having 3000 hidden nodes.

In our autoencoder pretraining, all network weights are first randomly initialized with uniform distribution and all biases are initialized at zero. The training criterion is the sum of squared error. After the pretraining error has converged, we remove the L1 norm constraint and use the trained network weights and biases as the initialization for the subsequent DNN training of the conversion function between source and target features.

Figure 3.2: Performing the forward-pass with the DNNs in different stages to demonstrate the effectiveness of autoencoder pretraining. A: spectrogram from source speech; B: spectrogram from target speech; C: output spectrogram from DNN with randomly initialized parameters; D: output spectrogram from DNN after autoencoder pretraining; E: output spectrogram from DNN after model training.

Different from the layer-wise autoencoder pretraining in [43], where the network is trained layer by layer. In our implementation, all parameters of autoencoder DNN are trained simultaneously. RBM pretraining is another popular pretraining method for DNN training. However, the optimization algorithm Contrastive Divergence [44] that is used for RBM pretraining is unsupervised and not targeted on reconstructing spectral parameters. Moreover, comparing to discriminative layer-wise pretraining, autoencoder pretraining does not require early termination, which is usually required in discriminative layer-wise pretraining to prevent overfitting problem on each hidden layer.

As mentioned, our DNN training process consists of three consecutive stages: initialization, pretraining and model training (fine-tuning). To demonstrate the effectiveness of the proposed pretraining, we examine the outputs of the model in all the three stages as shown in Fig. 3.2. Fig. 3.2A and 3.2B shows the source and target spectrograms respectively. The output spectrogram in Fig. 3.2C is generated by the randomly initialised DNN, and as expected show that the conversion to target fails utterly. The output spectrogram in Fig. 3.2D which is generated by our pre-trained DNN using autoencoder pretraining step a much more meaningful transformation, it resembles closely to the input as expected of an auto-encoder network. The output spectrogram in Fig. 3.2E is achieved after DNN-training between source and target pair using the pre-trained DNN model as initial system. The results show that the output spectrogram has become closer to the target spectrogram.

In this example, we use 40 epochs for autoencoder pretraining and 20 epochs for model training.

3.3.2 F0 Trajectory Conversion

It is well known that the F0 of an utterance generally depends on long context information [45]. To accomodate this into our modelling framework, we will performed a segment-level conversion to model this trajectory transformation between source and target as opposed to the shorter time-frame window considered by the DNN for spectral feature.

By segment-level, we define them as segments between voiced-unvoiced boundaries. Hence each segment is a continuous F0 measurement in voiced frames. To facilitate learning between variable length segments of source and target, the length of both source and

target segments are normalized into a predefined number L and passed to the DNN model for training. The length normalization allows F0 trajectory to be concisely captured to facilitate compact modeling. The normalized length will be chosen empirically from the distribution of segment length observed in training data and length normalization can be achieved by interpolation or sampling.

Given the length normalized source and target F0 feature vectors, we are now ready to train the DNN conversion function. Rather than modelling the desired output F0 values directly, we choose to model the output's F0 difference contour. Let $t_{i,j}$ be the output F0 features at feature element i and at frame index j . The output difference contour $t'_{i,j}$ is constructed by

$$t'_{i,j} = \begin{cases} 0, & \text{if } i = 1 \\ t_{i,j} - t_{i-1,j}, & \text{if } i > 1. \end{cases} \quad (3.1)$$

The reason to model the difference contour is motivated by [46], who reported that modeling relative pitch feature is more effective for singing voice synthesis and that smaller amount of training data is required for its training. Another way to understand this is to consider that multiple segments could have different absolute F0 levels while their F0 trajectories are similar. Hence, modeling the F0 contour allows these segments to share the model parameters and reduce the need of training data.

Once trained, our DNN will output the output F0 difference trajectory given the input voiced frame's segment F0. To generate the output F0 trajectory from its predicted difference trajectory, a reference mean level of the target F0 segment is necessary for the final F0 reconstruction. In this work, this target segmental mean level is calculated using the converted segment value from a frame-based F0 conversion system [29]. Hence, the reconstruction process for the output segments is achieved by

- (i) The target F0 trajectory is constructed from the output difference feature vector by first assuming that the first value is 0.

$$\hat{t}_{i,j} = \begin{cases} 0, & \text{if } i = 1 \\ \hat{t}_{i-1,j} + \hat{t}'_{i,j}, & \text{if } i > 1. \end{cases} \quad (3.2)$$

where $\hat{t}'_{i,j}$ is the i -th element in the j -th output segment from the DNN model.

- (ii) We then apply mean adjustment to each reconstructed trajectory

$$\hat{t}_{i,j}^* = \hat{t}_{i,j} - \frac{\sum_{i=1}^L \hat{t}_{i,j}}{L} + \hat{\mu}_j \quad (3.3)$$

where $\hat{\mu}_j$ is the predicted segmental mean level of the j -th segment and $\hat{t}_{i,j}^*$ is i -th element of the j -th the mean-adjusted F0 segment.

- (iii) The resultant segment is interpolated to the original length.

3.3.3 Intensity Trajectory and Phone Duration Conversion

For intensity, we perform a similar segmentation process as in F0 trajectory conversion, whereas intensity trajectories in voiced segments are extracted and then normalized in length. However, different to modelling F0, we choose to model intensity values directly between input and output feature vectors using DNN. Based on our experiments, similar performance was achieved when using either intensity ratio or the difference between consecutive frames as the input features. Once the model is trained, it will be used in the framework to scale the converted spectrum with the predicted intensity trajectory.

Given a source speaker utterance, the purpose of duration modelling it to allow the number of frames in the converted speech to be different to that of the source utterance. Without duration modification, the converted target features will have the same time alignment as the source speaker.

To model duration, our approach is to model the duration ratio between the source and target segment. The definition of segment here refers to a continuous spectral block associated with a phone, a syllable, etc. The duration modelling feature is constructed by sampling the input spectral frames from the desired segment. The output feature is simply a real number representing the ratio between the source and target segment. This is demonstrated in Fig. 3.3. In this example, each segment is constituted by three sampled frames. The number of sampled frames can be chosen empirically from the distribution of segment duration in the training data.

Once trained, the DNN predicts the duration ratio for the given spectral frames. This ratio is used to modify the converted feature's segment length by performing linear interpolation. In this work, we have only examined the performance of duration modeling with phone-based segment units.

Figure 3.3: Re-sample spectral segment to create input vector for duration modeling

3.4 Experiment Setup

To evaluate the performance of the proposed works, we conducted experiments on the CMU ARCTIC database [47]. four speaker pairs were used in our experiments

- BDL (male) to SLT (female)
- CLB (female) to RMS (male)
- CLB (female) to SLT(female)
- RMS (male) to BDL (male)

For each speaker pair, we choose 200 utterances for training and 60 non-overlapping utterances for testing. We used TANDEM-STRAIGHT [48] to extract the spectral feature with 512th-order, voiced-unvoiced flag (VUV) and F0 feature. These speech features were then used to create different input and output vectors for our conversion models.

The detailed descriptions of each input, output features and network architectures are described below:

- **Spectrum conversion:** We used 512th-order log spectral envelope and its delta and delta-delta coefficients and one dimension of VUV from source speaker as input feature vector and 512th-order log spectral envelope from target speaker as output feature vector. The DNN had three hidden layers; each hidden layer had 3000 nodes.
- **F0 trajectory conversion:** We extracted F0 input and output segments as described in Section 3.3.2 with the normalized segment length L of 55. The DNN had two hidden layers; each hidden layer had 500 nodes.
- **Intensity trajectory conversion:** Input and output vectors extracted as described in Section 3.3.3 with the normalized segment length of 55. The DNN architecture is the same as in F0 trajectory conversion stated above.
- **Phone duration conversion:** For each phone in source utterances, an input vector was created by concatenating five spectrum frames from the phone, and the output vector was the ratio between the current source phone duration and the corresponding target phone duration. The phone boundaries can be determined by any phone recognizer. In our work, we use the boundaries from CMU Sphinx recognizer [49].

All the DNNs in this work employed hyperbolic tangent activation in the hidden nodes and there was no non-linear activation function in any output nodes. In all the system below, aperiodicity feature was not converted. We use the same aperiodicity feature from the source speaker to reconstruct the converted speech.

3.5 Evaluation Results

3.5.1 Spectrum Conversion

We evaluate the performance of DNN-based spectral conversion using the following systems:

- **JD-GMM:** The spectral representation was 25-th order MCEP, and the training and conversion was done as described in Section 2.1.1.2. We used 64 Gaussian

components to build the system, which is empirically found in our preliminary experiments.

- **DNN-MCEP**: This system also used 25-th order MCEP as source and target feature. For DNN architecture and training parameters, we used the same settings as in [25]: two hidden layers with 50 nodes each layer, the learning rate was 0.01 and the momentum was 0.3.
- **DNN-SP256-DLP**: We built a system based on [26], in which the input vectors include log spectral frames with delta and delta-delta coefficients, log F0 with delta and delta-delta of the current frames and two context frames and one dimension of VUV. The DNN had two hidden layers with 1600 nodes each layer. We applied discriminative layer-wise pretraining (DLP) on this system.
- **DNN-SP512** (proposed): We used a DNN with the architecture described in Section 3.4. We did not perform any pretraining on this system.
- **DNN-SP512-DLP** (proposed): We used a DNN with the architecture described in Section 3.4 and applied discriminative layer-wise pretraining (DLP) on this system.
- **DNN-SP512-Autoencoder** (proposed): We used a DNN with the architecture described Section 3.4 and performed autoencoder pretraining as described in Section 3.3.1. The training data for the pretraining phase were 200 utterances from source speaker. This same set was also used in the model training stage.

3.5.1.1 Objective evaluation

We conducted objective evaluation to assess the proposed spectral conversion with autoencoder pretraining. Log spectral distortion (LSD) [50] was employed. The distortion of k -th order of log spectral pair is calculated as:

$$d(\mathbf{x}_k, \mathbf{y}_k) = \sum_{i=1}^M (\log(x_{k,i}) - \log(y_{k,i}))^2 \quad (3.4)$$

where M is the total number of frequency bin. A distortion ratio between converted-to-target distortion and the source-to-target distortion is defined as:

Table 3.1: Comparison of LSD ratio of different spectral conversion methods

Methods	LSD (%)
JD-GMM	87.93
DNN-MCEP	80.70
DNN-SP256-DLP	52.51
DNN-SP512	54.73
DNN-SP512-DLP	52.59
DNN-SP512-Autoencoder	51.31

$$\text{LSD} = \frac{\sum_{k=1}^K d(\hat{\mathbf{x}}_k, \mathbf{y}_k)}{\sum_{k=1}^K d(\mathbf{x}_k, \mathbf{y}_k)} * 100\% \quad (3.5)$$

where $\mathbf{x}_k$, $\mathbf{y}_k$ and $\hat{\mathbf{x}}_k$ denote the k -th frequency bin of the source, target and converted spectrum, respectively, and K denotes the number of frequency bins. A lower LSD indicates smaller distortion.

Table 3.1 presents the LSD for the baseline methods and our proposed methods. The average LSD over all evaluation pairs was reported. Comparing JD-GMM and DNN-MCEP with other methods, we first observed that the systems working on high dimensional representations of spectrum (i.e. DNN-SP256-DLP, DNN-SP512, DNN-SP512-Autoencoder) yield much lower distortion than those working on low dimensional representations (at least 25% lower).

Comparing among DNN-SP256-DLP, DNN-SP512 and DNN-SP512-DLP, DNN-SP512 and DNN-SP512-DLP achieve a slightly higher distortion than DNN-SP256-DLP, that is 54.73% and 52.59% to 52.51%, which might indicate the difficulty in achieving a good minimum in training DNN with high dimensional features. However, when employing our proposed autoencoder pretraining, DNN-SP512-Autoencoder converges to a better solution with a lower LSD, and has a slightly lower distortion than DNN-SP256-DLP, that is 51.31% to 52.51%. This indicates that the proposed autoencoder pretraining enables a better initialization for spectral conversion.

Note that discriminative layer-wise pretraining was applied in DNN-SP256-DLP; while in our proposed DNN-SP512-Autoencoder, autoencoder pretraining was implemented. Comparing the LSD performance of these two pre-trained methods, the combi-

nation of autoencoder pretraining and DNN-SP512 (DNN-SP512-Autoencoder) yields a lower LSD with a doubled resolution in frequency.

3.5.1.2 Subjective evaluation

We conducted listening tests to access speech quality and speaker similarity of the converted results. 10 subjects participated in all listening tests. In this subjective evaluation, we compared DNN-SP512-Autoencoder with DNN-SP256-DLP, DNN-MCEP and JD-GMM. We did not include DNN-SP512 nor DNN-SP512-DLP due to the highly similar structure with DNN-SP512-Autoencoder.

We first performed AB preference tests to access speech quality. 20 pairs were randomly selected from 80 paired samples. In each pair, A and B were the samples from the proposed method and a baseline method, respectively, presented to listeners in a random order. Each listener was asked to listen to both samples and decide which sample was better in term of quality.

We then conducted XAB tests to access the speaker similarity. Similar to the AB test, 20 pairs were randomly selected from the 80 paired samples. In each pair, X was the reference target sample, and A and B were the converted samples of comparison methods listed in the first column of Table 3.2, presented to listeners in a random order. The listeners were asked to listen to the sample X first, then A and B, and then decide which sample was closer to the reference target sample. For each pair of samples, X, A and B have the same language content.

The subjective evaluation results are presented in Table 3.2. Comparing to JD-GMM, our proposed method DNN-SP512-Autoencoder achieves much higher preference score in both quality and similarity test. Comparing to DNN-MCEP, DNN-SP512-Autoencoder also yields better results in both quality and similarity evaluations. Note that the spectral resolution in DNN-SP512-Autoencoder is much higher than JD-GMM and DNN-MCEP (In JD-GMM and DNN-MCEP, spectral envelope is represented by 25th-order MCEP). Spectral details are clearer in DNN-SP512-Autoencoder.

In the comparison between the state-of-the-art DNN-SP256-DLP and our proposed method DNN-SP512-Autoencoder, our proposed method achieves a slightly better subjective results in both quality and similarity. The results further confirms that DNN is

Table 3.2: Results of average quality and similarity preference tests with 95% confidence intervals of different spectral conversion methods

Methods	Preference score (%)	
	Quality test	Similarity test
JD-GMM	35.8 (± 6.7)	36.8 (± 8.3)
DNN-SP512-Autoencoder	64.2 (± 6.7)	63.2 (± 8.3)
DNN-MCEP	42.5 (± 3.8)	42.0 (± 6.6)
DNN-SP512-Autoencoder	57.5 (± 3.8)	58.0 (± 6.6)
DNN-SP256-DLP	46.6 (± 3.8)	44.0 (± 6.4)
DNN-SP512-Autoencoder	53.3 (± 3.8)	56.0 (± 6.4)

capable of modeling high-resolution spectra and autoencoder pretraining can effectively improve the performance of DNN training for voice conversion.

3.5.2 Spectrum and F0 Conversion

We compared our proposed segment-based F0 conversion system with two baseline systems:

- **Mean-Var:** F0 is converted using the conventional global mean-variance transformation from source F0 [51] as follows,

$$f_o = \mu_t + \frac{\sigma_t}{\sigma_s} * (f_s - \mu_s). \quad (3.6)$$

where f_s and f_o are the source F0 and the output converted F0, respectively. μ_s , σ_s , μ_t , σ_t are the global mean and standard deviation statistics of the F0s from source and target speaker, respectively.

- **DNN-Frame:** F0 is modeled using a frame-based approach with DNN [29]. In this system, the input vector contains 256th-order log spectral envelope (static, delta and delta-delta), F0 (static values within a 7-element window, delta and delta-delta) and VUV. The DNN has two hidden layers with 1600 nodes each.
- **DNN-Segment** (proposed): The F0 segment is modeled as described in Section 3.3.2. The DNN architecture is described in Section 3.4.

Table 3.3: Comparison of RMSE of different F0 conversion methods

Methods	RMSE (Hz)
Mean-Var	22.06
DNN-Frame	17.90
DNN-Segment	17.80

In all F0 conversion systems, the result from DNN-SP512-Autoencoder was used as spectrum feature for speech reconstruction.

3.5.2.1 Objective evaluation

We employed root mean square error (RMSE) as the objective measurement to compare our proposed methods with the baseline systems. The average RMSE over all evaluation pairs was reported. A lower RMSE indicates smaller distortion.

Table 3.3 presents the RMSE for the baseline methods and our proposed method. Comparing to Mean-Var, DNN-Segment achieves lower RMSE, that is 17.80 to 22.06, and comparing to DNN-Frame, DNN-Segment achieves a slightly lower RMSE, that is 17.80 to 17.90, which confirms the effectiveness of the segment-based trajectory modeling method.

3.5.2.2 Subjective evaluation

We performed further evaluation with subjective listening test. We conducted an XAB similarity preference test in this evaluation. The setup is similarly to the XAB test in Section 3.5.1.2. We instructed the listeners to choose the sample whose prosody sound more similar to the reference target sample.

The subjective evaluation results are presented in Table 3.4. The similarity tests further confirm that our proposed method DNN-Segment provides better F0 modeling than the conventional Mean-Var method by a significantly better preference score, that is 61.2 to 38.8. Our proposed segment-based method DNN-Segment also has comparable preference score with state-of-the-art F0 modeling using DNNs.

A demonstration of the results from F0 conversion method is showed in Fig. 3.4.

Table 3.4: Results of average similarity preference tests with 95% confidence intervals of different F0 conversion methods

Methods	Preference score (%)
	Similarity test
Mean-Var	38.8 (± 2.1)
DNN-Segment	61.2 (± 2.1)
DNN-Frame	48.8 (± 4.1)
DNN-Segment	51.2 (± 4.1)

Figure 3.4: Outputs of F0 conversion from three methods: Mean-Var, DNN-Frame and DNN-Segment

Table 3.5: Results of average similarity preference tests with 95% confidence intervals of the proposed intensity and duration conversion method

Methods	Preference score (%)
	Similarity test
Without intensity and duration	31.2 (± 4.1)
With intensity and duration	68.8 (± 4.1)

3.5.3 The Comprehensive Framework: Spectrum, F0 and Intensity and Duration Conversion

Finally, we combined our spectrum conversion and F0 conversion system with the proposed intensity and duration conversion described in Section 3.3.3. We compared the speaker similarity between the system with and without intensity and duration modification using listening tests. In this evaluation, we performed the evaluation on speaker pair CLB to SLT, where the intensity contour and time alignment between source and target are noticeably different.

The subjective evaluation results are presented in Table 3.5. The preference scores of systems with and without intensity and duration conversion are 68.8 and 31.2, respectively. This further suggests the effectiveness of our proposed system in converting prosodic features and the importance of having a comprehensive framework in voice conversion.

Some samples of this work are presented in the web link: <http://listeningtests.net/voiceconversion/hynq2015comprehensive>.

Our proposed voice conversion on various prosodic and spectral features is built by leveraging on the above merits of DNNs. Speaker characteristics are carried by various speech features. Besides spectrum, comprehensive voice conversion requires other features for transformation, for example, F0. Nevertheless, unlike spectrum, there are not many established works on F0 conversion. Some methods include transformation using vector quantization [13], partial least square regression [30] and DNN mapping [47]. Among these, [30] and [47] have focused on wavelet domain instead of on the frequency domain directly. The most popular method is the mean-variance global transformation, adjusting the average level and the range of source F0 values [34]. This method retains the general

Figure 3.5: DNN-based Voice Conversion

shape of the source F0 trajectory and ignores the detailed difference between the two F0 contours. It is a frame-level operation, while human manipulates F0 in a segmental manner with a scope which may be up to phone, syllable, word, phrase or sentence level. F0 modeling and generation becomes challenging in voice conversion, as the amount of training data is usually sparse with tens to a few hundreds of utterances only.

We focus on using a DNN to model the transformation from source to target speech features as demonstrated in Fig. 3.5. Most of the state-of-the-art DNN-based voice conversion methods are on spectral transformation. In those methods, spectral parameters of the source speech are extracted and put into an input vector for conversion. This input vector is then passed to a DNN-based conversion network, where the network performance is governed by the network architecture, node weights and biases and the associated activation functions. By feeding forward the input vector through the network, an output vector is generated which represents the converted speech parameters. These output parameters are finally used to construct the converted waveform. Conversion on other features, such as F0, can also be done similarly.

3.6 Conclusion

In this chapter, we have introduced a comprehensive framework for voice conversion, in which high resolution spectra features, F0, intensity and segment duration features are converted using DNNs. To overcome limited training data issues, we modified existing

autoencoder pre-training method and showed its usefulness to effectively initialize the DNN model parameters and improve spectral transformation modelling.

The resultant spectral conversion model with our proposed pretraining achieves comparable ratings in terms of similarity and quality as another state-of-the-art pretrained voice conversion system [26]. In our comprehensive system, we have also considered the use of the two prosodic features (intensity and duration) which were not typically applied for voice conversion. For these features, we introduced segment-level modeling. Our experimental results showed that both objective and subjective evaluations of the converted speech is more similar to the target speaker than existing state of the art approaches.

We have showed that it is possible to build a conversion framework to transform multiple features simultaneously. Nevertheless, many open questions remain regarding how better to jointly model the timbre and prosodic features. Lastly, to better understand the performance of intensity and duration conversion, future works to evaluate the perceptual effect of intensity and duration modification through more intensive psychoacoustic experiments is planned.

Chapter 4

DNN Adaptation for Voice Conversion

The conventional voice conversion technique discussed in the previous chapter builds an individual conversion model for each speaker pair. To have a conversion for each new speaker pair, parallel speech data from the two speakers are required, and training must be repeated. The training is usually computationally expensive and has to be done offline. In some cases, it may not be even possible to obtain sufficient training data. In this chapter, we will introduce the idea of using the source and target speaker’s identities as the additional input features to perform DNN adaptation.

The chapter is organized as follows. In Section 4.1, we first reviewed existing work related to DNN adaptation. In Section 4.2, then we introduced the contributions of this work. In Section 4.3, we presented a detailed description of the proposed framework, and in Section 4.4 we summarize our conclusions.

4.1 DNN Adaptation

Speaker adaptation attempt to optimize system performance by compensating the mismatch between speaker independent models and testing speakers based on a small amount of adaptation data. Due to efficiency and stable performance, DNN model adaptation has been an important research topic in several areas, such as automatic speech recognition (ASR) [52], speaker verification (SV) [53] and text-to-speech (TTS) [54, 55].

One way to realize the DNN model adaptation is to modify the target speaker features to match the given speaker independent models. The speaker features transformation

can be estimated in many ways. In [56], a technique was proposed to transform the target speaker features by normalize the vocal tract length. In [57, 58], feature modification was achieved by Maximum likelihood linear regression (MLLR) criterion in feature space. The effective of these feature transformation techniques for DNN model adaptation have been shown in [36, 52, 55, 59, 60].

An alternative way to realize the DNN model adaptation is to modify the speaker-independent DNN model towards one particular speaker. A number of speaker adaptation methods have been proposed. For example, in [61, 62], authors attempt to adapt the speaker-independent DNN model by adding additional layers. [63] proposed a retrained sub-set hidden units (RSHU) method to adapt the DNN model by update the weights of active hidden nodes. Similar technique, namely learning hidden unit contributions (LHUC), also proposed for DNN model adaptation [64].

Recently, the speaker codes and i-vector based DNN model adaptation has been proposed [53, 65, 66, 67, 68]. In such methods, the speakers code of i-vectors representing the speaker identities are incorporated as joint feature inputs to train the speaker independent DNN model. The speaker independent DNN model is capable to learn the transformation from training speaker features to a speaker-independent feature space. For a new target speaker, the model can be simply adapted by feeding the new speaker code to the DNN. As the speaker code is able to learn by a small mount of data, e.g. one or two sentences, this method can significantly reduce the requirement of adaptation data. Moreover, during adaptation process the weights of DNN model are not changed, hence the efficiency is improved.

4.2 Contributions

Inspired of the speaker codes based DNN adaptation, we propose a DNN adaptation based voice conversion framework, namely Voice Producer. The proposed DNN adaptation is performed by including additional input features representing the source and target speaker's identity. Specifically, we include both speakers' identity feature vectors into the input feature stream. By such a structure, the trained DNN learns the mapping of the input spectral features to that of the target speaker with knowledge of the two

speaker identities. During runtime, the desired source and target speakers' identities are extracted using a small number of utterances. These identity vectors are then used with the source speaker's input spectral feature for conversion. By our proposed approach, no DNN re-training is required for the conversion function estimation of new speaker pair.

The speaker's identity is captured using probabilistic linear discriminant analysis (PLDA) [69, 70] framework to measure the relation between the speaker's voice to a close set of reference speaker. Hence, the speaker's identity can be interpreted as the posterior probabilities of the set of reference speakers.

The approach and demonstrator was presented in the show and tell of ICASSP 2016 [2]. Our informal listening test showed that it is possible to produce conversion for different source and target speaker pairs.

4.3 System Overview

This section describes the Voice Producer DNN adaptation framework for voice conversion. The overall structure of the framework is illustrated in Figure 4.1. The novelty of the structure is that both source and speaker's identity vector into the input feature stream to facility the DNN adaptation. By such a structure, the input stream consists of the source spectral feature, source speaker identity vector and target speaker identity vector, and the output stream is the target spectral feature. The spectral features are analyzed using TANDEMSTRAIGHT [48] and the speaker identity vector is analyzed using PLDA [70].

To enable the VoiceProducer framework to learn the conversion function between different speaker pairs, we must train the system with many different speaker pairs. In this work, using the six reference speakers from CMU ARCTIC dataset: AWK, BDL, CLB, JMK, RMS, SLB, we generated 36 different speaker pairs. For each pair, we have 100 parallel utterances. These are jointly used to train the DNN model is illustrated in Figure 4.2 (Training), where Speaker A represents the source and Speaker B represents the target speaker respectively. To prevent the model from over-fitting certain pair, the order of utterances in the training data are randomized among all pairs. Similar to the DNN training in Chapter 3, DTW is used to align the spectral frames of the parallel

Figure 4.1: DNN architecture of VoiceProducer.

utterances, and auto-encoder-based pre-training is used to speed up the training process and improve the performance of the system.

The runtime process is illustrated in Figure 4.2 (Runtime). The source and target speakers' identities are first determined by passing each speaker's utterances through the speaker decomposition module which performs the PLDA process. For conversion, each source speaker's utterance is extracted and augmented with the speaker's identity to generate the target speaker's spectral feature. Note that the speaker's identity vector is able to be generated using any of the reference speaker's recordings (i.e. known to unknown speaker) or even from an arbitrary speaker (i.e. unknown to unknown speaker). In this context, known speakers refer to those that are in the training data of the DNN model, and unknown speakers refer to those that are not. This indicates that there are two advantages of the proposed system: (1) it does not require parallel training data from the source and target speakers but only a very small amount of speech sample to decompose the speakers' identities and (2) it does not require to re-train the model to facilitate new speaker pair. Figure 4.3 shows the user interface of our implemented application presented in the show and tell session of ICASSP2016.

Figure 4.2: Training and runtime process of VoiceProducer.

Figure 4.3: User interface of VoiceProducer.

4.4 Conclusion

This chapter presented the VoiceProducer approach for a DNN adaptation voice conversion system that can learn the transformation of multiple speaker pairs simultaneously. The system is applicable for the case when there is insufficient parallel training utterances from target speaker. Additionally, our empirical results showed that it is possible to modify the transformation of the model in a controlled manner by providing different source and target identity vectors, which allows conversion to voices that are not in the training data.

In our informal listening tests, the system performed well in the case of known to known speaker and known to unknown speaker, while did not perform very well for the case of unknown to unknown speaker. Since unknown speakers are represented as a combination of the reference speaker, we speculate the six reference speakers in the current system are too few to represent the speaker characteristic space. Some demonstration of the work is presented in the web link <http://www.listeningtests.net/voiceconversion/voiceproducer>.

Due to time constraint, experimental results measuring the quality of the voice conversion were not performed. We have highlighted this as future work to be done.

Chapter 5

Conclusion

In this thesis, we studied two methods to improve the voice conversion performance using DNN. In the first contribution, a comprehensive framework to generate both spectral and prosodic features by DNN is proposed. In the second contribution, DNN adaptation to new source/target speaker pairs are achieved by employing speaker identity vectors into the input feature space. The conclusion and future research directions are presented in this chapter.

Comprehensive DNN Framework for Voice Conversion

We conducted a study on using DNN to learn the transformations of both the high dimensional spectral features and three prosodic features, namely, F0, intensity and duration. Traditional voice conversion techniques only focus on spectral feature conversion. However, a speaker's speech is characterized by both spectral as well as prosodic features. Hence, techniques should be introduced to address conversion functions which simultaneously model the transformation and generate these features. Moreover, due to the model capability, low dimensional features usually employed in traditional GMM-based voice conversion methods. This is one of the reasons caused the over-smoothing effects of converted target speech. To overcome the smoothing problem when using GMM modelling with low dimensional feature, we use DNN to predict spectral at high dimension. To further alleviate the limited training data problem of DNN training, we incorporated the auto-encoder pre-training.

The experiment conducted on the CMU ARCTIC database shows that the proposed framework outperforms state-of-the-art GMM- and DNN-based voice conversion methods with low-dimensional feature as input in terms of similarity and quality significantly. Listening test also confirmed that the use of prosodic features can further improve the performance.

The work has been published in [1] Multimedia Tools and Applications journal.

DNN Adaptation for voice conversion

In Chapter 3, we have seen that DNN can be deployed to learn the voice conversion function. Although the amount of training data used to train the system is relatively small, about 200 parallel sentences, it may still be a challenge to get this amount of parallel training data from desired speaker pairs.

To reduce the need of parallel training data for new speaker pairs, we introduced a novel DNN adaptation technique in Chapter 4. In this preliminary work, we achieved adaptation by incorporating source and target speaker identities as input features to the DNN system. During DNN training, we still require multiple speaker pairs with sufficient amount of parallel training data. In this work, we trained a DNN using 36 source and target speaker pairs for the spectral feature conversion task. To achieve adaptation, we extract the source and target speaker identities and use them as bias vectors in the input features. Specifically, we concatenate the source and target speaker identity features with the source spectral features to learn the conversion function. By the presence of the two speaker identity vectors, the DNN not only learns the mapping between the source to target spectral features, it also learns the transformation between the source and target speaker pairs. During testing, our approach only requires a few utterances from new source and target speaker pair to generate the speaker identity vectors. The DNN is then applied on the source speaker's spectral features with the two speaker identity vectors to generate target spectral features. The conversion between the new unseen speaker pair is achieved by the DNN interpolating from its learned transformations of training speakers pairs. Informal listening test shows that conversion is somewhat achieved. This work has been published in ICASSP2016 show and tell [2] with a demo to show one-many, many-one and many-many speaker pair conversion.

Future Works

We list the following three possible future works:

- (i) Currently, we have only explored feedforward DNN for voice conversion. Due to the limited amount of training data, we had only include very limited context information to learn the transformation between the source and target speaker. This can be improved by using more advance neural network technique such as RNN or LSTM which naturally captures long term context information within their structure.
- (ii) In Chapter 4 , our focus was to demonstrate the capability of the proposed framework for DNN adaptation. Only informal listening tests were used to show its possibility. Hence, we propose more concrete experimental works to measure and compare its performance against other state of the art approaches.
- (iii) In Chapter 4 , we focus only on the spectral feature transformation. In the future, we also propose to include the application of prosodic features in the adaptation framework and measure its performance.

References

- [1] H. Q. Nguyen, S. W. Lee, X. Tian, M. Dong, and E. S. Chng, “High quality voice conversion using prosodic and high-resolution spectral features,” *Multimedia Tools and Applications*, 2015.
- [2] S. W. Lee, Q. H. Nguyen, K. A. Lee, X. Tian, L. Xu, E. S. Chng, and H. Li, “Voice-producer: Many-to-many voice conversion for arbitrary speakers,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Show and Tell*, IEEE, 2016.
- [3] L. Meng, Q. H. Nguyen, X. Tian, Z. Shen, E. S. Chng, F. Y. Guan, C. Miao, and C. Leung, “Towards age-friendly e-commerce through crowd-improved speech recognition, multimodal search, and personalized speech feedback,” in *Proceedings of the 1st International Conference on Crowd Science and Engineering (ICCSE’16)*, July 2016.
- [4] X. Tian, Z. Wu, S. W. Lee, Q. H. Nguyen, M. Dong, and E. S. Chng, “System fusion for high-performance voice conversion,” in *Proceedings of INTERSPEECH 2015*, pp. 2759–2763, ISCA, September 2015.
- [5] P. Belin, S. Fecteau, and C. Bedard, “Thinking the voice: neural correlates of voice perception,” *Trends in cognitive sciences*, vol. 8, no. 3, pp. 129–135, 2004.
- [6] T. Kinnunen and H. Li, “An overview of text-independent speaker recognition: From features to supervectors,” *Speech communication*, vol. 52, no. 1, pp. 12–40, 2010.
- [7] D. G. Childers and C. Lee, “Vocal quality factors: Analysis, synthesis, and perception,” *The Journal of the Acoustical Society of America*, vol. 90, no. 5, pp. 2394–2410, 1991.

REFERENCES

- [8] T. Kinnunen, Z.-Z. Wu, K. A. Lee, F. Sedlak, E. S. Chng, and H. Li, “Vulnerability of speaker verification systems against voice conversion spoofing attacks: The case of telephone speech,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4401–4404, IEEE, 2012.
- [9] S. W. Lee, Z. Wu, M. Dong, X. Tian, and H. Li, “A comparative study of spectral transformation techniques for singing voice synthesis,” in *Proc. INTERSPEECH*, no. September, pp. 2499–2503, 2014.
- [10] T. Toda, M. Nakagiri, and K. Shikano, “Statistical voice conversion techniques for body-conducted unvoiced speech enhancement,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 9, pp. 2505–2517, 2012.
- [11] D. Hironori, K. Nakamura, T. Tomoki, H. Saruwatari, and K. Shikano, “Esophageal speech enhancement based on statistical voice conversion with gaussian mixture models,” *IEICE TRANSACTIONS on Information and Systems*, vol. 93, no. 9, pp. 2472–2482, 2010.
- [12] L. Rabiner and R. Schafer, “Digital speech processing,” *The Froehlich/Kent Encyclopedia of Telecommunications*, vol. 6, pp. 237–258, 2011.
- [13] S. Imai, T. Kobayashi, K. Tokuda, T. Masuko, K. Koishida, S. Sako, and H. Zen, “Speech signal processing toolkit (sptk), version 3.3,” 2009.
- [14] H. Kawahara, “Straight, exploitation of the other aspect of vocoder: Perceptually isomorphic decomposition of speech sounds,” *Acoustical science and technology*, vol. 27, no. 6, pp. 349–353, 2006.
- [15] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, “Voice conversion through vector quantization,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 655–658, 1988.
- [16] T. Toda, A. W. Black, and K. Tokuda, “Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

REFERENCES

- [17] A. Kain and M. Macon, “Spectral voice conversion for text-to-speech synthesis,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 1, pp. 285–288 vol.1, May 1998.
- [18] Z. Wu, T. Virtanen, T. Kinnunen, E. S. Chng, and H. Li, “Exemplar-based voice conversion using non-negative spectrogram deconvolution,” in *Proc. 8th ISCA Speech Synthesis Workshop*, pp. 201–206, 2013.
- [19] Y. Stylianou, O. Cappé, and E. Moulines, “Continuous probabilistic transform for voice conversion,” *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 131–142, 1998.
- [20] H.-T. Hwang, Y. Tsao, H.-M. Wang, Y.-R. Wang, and S.-H. Chen, “Incorporating global variance in the training phase of GMM-based voice conversion,” in *Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pp. 1–6, 2013.
- [21] L.-H. Chen, Z.-H. Ling, L.-J. Liu, and L.-R. Dai, “Voice conversion using deep neural networks with layer-wise generative training,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 22, no. 12, pp. 1859–1872, 2014.
- [22] Z.-H. Ling, S.-Y. Kang, H. Zen, A. Senior, M. Schuster, X.-J. Qian, H. Meng, and L. Deng, “Deep learning for acoustic modeling in parametric speech generation,” *IEEE Signal Processing Magazine*, no. April, pp. 35–52, 2015.
- [23] K. Hornik, M. Stinchcombe, and H. White, “Multilayer feedforward networks are universal approximators,” *Neural networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [24] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, pp. 533–536, 1986.
- [25] S. Desai, E. Raghavendra, B. Yegnanarayana, A. Black, and K. Prahallad, “Voice conversion using artificial neural networks,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3893–3896, 2009.

REFERENCES

- [26] F.-L. Xie, Y. Qian, Y. Fan, F. K. Soong, and H. Li, “Sequence error (SE) minimization training of neural network for voice conversion,” in *Proc. INTERSPEECH*, no. September, pp. 2283–2287, 2014.
- [27] E. E. Helander and J. Nurminen, “A novel method for prosody prediction in voice conversion,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4, pp. 509–512, 2007.
- [28] G. Sanchez, H. Silen, J. Nurminen, and M. Gabbouj, “Hierarchical modeling of F0 contours for voice conversion,” in *Proc. INTERSPEECH*, no. September, pp. 2318–2321, 2014.
- [29] F.-L. Xie, Y. Qian, F. K. Soong, and H. Li, “Pitch transformation in neural network based voice conversion,” in *Proc. The 9th International Symposium on Chinese Spoken Language Processing (ISCSLP)*, pp. 197–200, IEEE, September 2014.
- [30] T. Yoshimura, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, “Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis,” in *Proc. EUROSPEECH*, pp. 2347–2350, 1999.
- [31] C. H. Wu, C. C. Hsia, T. H. Liu, and J. F. Wang, “Voice conversion using duration-embedded Bi-HMMs for expressive speech synthesis,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 14, no. 4, pp. 1109–1116, 2006.
- [32] K. Yutani, Y. Uto, Y. Nankaku, T. Toda, and K. Tokuda, “Simultaneous conversion of duration and spectrum based on statistical models including time-sequence matching,” in *Proc. INTERSPEECH*, no. 3, pp. 1072–1075, 2008.
- [33] R. Srikanth, “Duration modelling in voice conversion using artificial neural networks,” in *Proc. The Annual International Conference on Systems, Signals and Image Processing*, pp. 556–559, 2012.
- [34] A. Sorin, S. Shechtman, and V. Pollet, “Coherent modification of pitch and energy for expressive prosody implantation,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4914–4918, 2015.

REFERENCES

- [35] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, “Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.
- [36] J. Pan, C. Liu, Z. Wang, Y. Hu, and H. Jiang, “Investigation of deep neural networks (dnn) for large vocabulary continuous speech recognition: Why dnn surpasses gmms in acoustic modeling,” in *Chinese Spoken Language Processing (ISCSLP), 2012 8th International Symposium on*, pp. 301–305, IEEE, 2012.
- [37] R. Collobert and J. Weston, “A unified architecture for natural language processing: Deep neural networks with multitask learning,” in *Proceedings of the 25th international conference on Machine learning*, pp. 160–167, ACM, 2008.
- [38] L.-J. Liu, L.-H. Chen, Z.-H. Ling, and L.-R. Dai, “Spectral conversion using deep neural networks trained with multi-source speakers,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 4849–4853, 2015.
- [39] T. Nakashika and R. Takashima, “Voice conversion in high-order eigen space using deep belief nets,” in *Proc. INTERSPEECH*, no. August, pp. 369–372, 2013.
- [40] D. Erhan, P.-A. Manzagol, Y. Bengio, S. Bengio, and P. Vincent, “The difficulty of training deep architectures and the effect of unsupervised pre-training,” in *AISTATS*, vol. 5, pp. 153–160, 2009.
- [41] G. E. Dahl, D. Yu, L. Deng, and A. Acero, “Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.
- [42] Q. V. Le, A. Coates, B. Prochnow, and A. Y. Ng, “On optimization methods for deep learning,” in *Proc. The 28th International Conference on Machine Learning (ICML)*, pp. 265–272, 2011.

- [43] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P. Manzagol, “Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion,” *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, December 2010.
- [44] G. Hinton, “Training products of experts by minimizing contrastive divergence,” *Neural computation*, vol. 14, no. 8, pp. 1771–1800, 2002.
- [45] G. A. Meyer, *The Semantics of Stress and Pitch in English*. The Faculty Association, Utah State University, 1961.
- [46] S. W. Lee, S. T. Ang, M. Dong, and H. Li, “Generalized F0 modelling with absolute and relative pitch features for singing voice synthesis,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 429–432, 2012.
- [47] J. Kominek and A. W. Black, “CMU ARCTIC databases for speech synthesis,” tech. rep., 2003.
- [48] H. Kawahara, M. Morise, T. Takahashi, R. Nisimura, T. Irino, and H. Banno, “TANDEM-STRAIGHT: a temporally stable power spectral representation for periodic signals and applications to interference-free spectrum, F0, and aperiodicity estimation,” in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 3933–3936, 2008.
- [49] P. Lamere, P. Kwok, W. Walker, E. B. Gouvêa, R. Singh, B. Raj, and P. Wolf, “Design of the CMU Sphinx-4 decoder,” in *Proc. EUROSPEECH*, pp. 1181–1184, 2003.
- [50] H. Ye and S. Young, “High quality voice morphing,” in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, pp. I–9–12, 2004.
- [51] Y. Stylianou, O. Cappé, and E. Moulines, “Continuous probabilistic transform for voice conversion,” *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 2, pp. 131–142, 1998.

REFERENCES

- [52] F. Seide, G. Li, X. Chen, and D. Yu, “Feature engineering in context-dependent deep neural networks for conversational speech transcription,” in *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, pp. 24–29, IEEE, 2011.
- [53] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, “Speaker adaptation of neural network acoustic models using i-vectors.,” in *ASRU*, pp. 55–59, 2013.
- [54] Y. Fan, Y. Qian, F. K. Soong, and L. He, “Multi-speaker modeling and speaker adaptation for dnn-based tts synthesis,” in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pp. 4475–4479, IEEE, 2015.
- [55] Z. Wu, P. Swietojanski, C. Veaux, S. Renals, and S. King, “A study of speaker adaptation for dnn-based speech synthesis.,” in *INTERSPEECH*, pp. 879–883, 2015.
- [56] E. Eide and H. Gish, “A parametric approach to vocal tract length normalization,” in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 1, pp. 346–348, IEEE, 1996.
- [57] C. J. Leggetter and P. C. Woodland, “Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models,” *Computer Speech & Language*, vol. 9, no. 2, pp. 171–185, 1995.
- [58] M. J. Gales, “Maximum likelihood linear transformations for hmm-based speech recognition,” *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.
- [59] S. H. K. Parthasarathi, B. Hoffmeister, S. Matsoukas, A. Mandal, N. Strom, and S. Garimella, “fmllr based feature-space speaker adaptation of dnn acoustic models.,” in *INTERSPEECH*, pp. 3630–3634, 2015.
- [60] H. Huang and K. C. Sim, “An investigation of augmenting speaker representations to improve speaker normalisation for dnn-based speech recognition,” in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pp. 4610–4613, IEEE, 2015.

- [61] J. Neto, L. Almeida, M. Hochberg, C. Martins, L. Nunes, S. Renals, and T. Robinson, “Speaker-adaptation for hybrid hmm-ann continuous speech recognition system,” 1995.
- [62] R. Gemello, F. Mana, S. Scanzio, P. Laface, and R. De Mori, “Linear hidden transformations for adaptation of hybrid ann/hmm models,” *Speech Communication*, vol. 49, no. 10, pp. 827–835, 2007.
- [63] J. Stadermann and G. Rigoll, “Two-stage speaker adaptation of hybrid tied-posterior acoustic models,” in *Acoustics, Speech, and Signal Processing, 2005. Proceedings.(ICASSP’05). IEEE International Conference on*, vol. 1, pp. I–977, IEEE, 2005.
- [64] P. Swietojanski and S. Renals, “Learning hidden unit contributions for unsupervised speaker adaptation of neural network acoustic models,” in *Spoken Language Technology Workshop (SLT), 2014 IEEE*, pp. 171–176, IEEE, 2014.
- [65] P. Karanasou, Y. Wang, M. J. Gales, and P. C. Woodland, “Adaptation of deep neural network acoustic models using factorised i-vectors,” in *Interspeech*, pp. 2180–2184, 2014.
- [66] S. Xue, O. Abdel-Hamid, H. Jiang, L. Dai, and Q. Liu, “Fast adaptation of deep neural network based on discriminant codes for speech recognition,” *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 12, pp. 1713–1725, 2014.
- [67] S. Xue, H. Jiang, L. Dai, and Q. Liu, “Unsupervised speaker adaptation of deep neural network based on the combination of speaker codes and singular value decomposition for speech recognition,” in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, pp. 4555–4559, IEEE, 2015.
- [68] V. Gupta, P. Kenny, P. Ouellet, and T. Stafylakis, “I-vector-based speaker adaptation of deep neural networks for french broadcast audio transcription,” in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pp. 6334–6338, IEEE, 2014.

REFERENCES

- [69] S. Ioffe, “Probabilistic linear discriminant analysis,” in *European Conference on Computer Vision*, pp. 531–542, Springer, 2006.
- [70] L. Burget, O. Plchot, S. Cumani, O. Glembek, P. Matějka, and N. Brümmer, “Discriminatively trained probabilistic linear discriminant analysis for speaker verification,” in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pp. 4832–4835, IEEE, 2011.