Voice Conversion in High-order Eigen Space Using Deep Belief Nets

Toru Nakashika[1], *Ryoichi Takashima*[1], *Tetsuya Takiguchi*[2], *Yasuo Ariki*[2]

[1]Graduate School of System Informatics, Kobe University, 1-1 Rokkodai, Kobe, Japan
[2]Organization of Advanced Science and Technology, Kobe University, 1-1 Rokkodai, Kobe, Japan

nakashika@me.cs.scitec.kobe-u.ac.jp, takashima@me.cs.scitec.kobe-u.ac.jp
takigu@kobe-u.ac.jp, ariki@kobe-u.ac.jp

Abstract

This paper presents a voice conversion technique using Deep Belief Nets (DBNs) to build high-order eigen spaces of the source/target speakers, where it is easier to convert the source speech to the target speech than in the traditional cepstrum space. DBNs have a deep architecture that automatically discovers abstractions to maximally express the original input features. If we train the DBNs using only the speech of an individual speaker, it can be considered that there is less phonological information and relatively more speaker individuality in the output features at the highest layer. Training the DBNs for a source speaker and a target speaker, we can then connect and convert the speaker individuality abstractions using Neural Networks (NNs). The converted abstraction of the source speaker is then brought back to the cepstrum space using an inverse process of the DBNs of the target speaker. We conducted speaker-voice conversion experiments and confirmed the efficacy of our method with respect to subjective and objective criteria, comparing it with the conventional Gaussian Mixture Model-based method.

Index Terms: voice conversion, deep learning, deep belief nets

1. Introduction

Voice conversion (VC) is a technique for changing specific information in the speech of a source speaker to that of a target speaker, while retaining the other information in the utterance such as its linguistic information. The VC techniques have been applied to various tasks, such as speech enhancement [1], emotion conversion [2], speaking assistance [3], and other applications [4, 5]. Most of the related works in voice conversion focus on the conversion of spectrum features, and we conform to that.

Many statistical approaches to VC have been studied so far [6, 7]. Among these approaches, the GMM-based mapping method [8] is widely used, and a number of improvements have been proposed. Toda et al. [9] introduced dynamic features and the global variance (GV) of the converted spectra over a time sequence. Helnder et al. [10] proposed transforms based on Partial Least Squares (PLS) in order to prevent the over-fitting problem of standard multivariate regression. There have also been approaches that does not require parallel data by using a GMM adaptation technique [11], eigen-voice GMM [13, 14] or probabilistic integration model [12]. Other approaches based on statistical approaches has been proposed; Jian et al. [15] used canonical correlation analysis for the VC, and Takashima et al. [16] proposed a VC technique using exemplar-based NMF.

However, most of the conventional works, including GMM-based approaches, relied on "shallow" voice conversion, in which a source speech was converted in the original feature space directly or in the shallow architecture with a few hidden layers. To capture the characteristics of the speech more precisely, it is necessary to have more hidden layers in the stack. An important method has been proposed by Desai et al. [17] based on Neural Networks (NNs). The NN-based approach has another advantage in addition to having multiple hidden layers. In the GMM-based approaches, the conversion is achieved so as to maximize the conditional probability calculated from a joint probability of source speech and target speech, which is trained beforehand. On the other hand, NN-based approaches directly train the conditional probability which converts the feature vector of a source speaker to that of a target speaker. It is often reported that such a discriminative approach performs better than a generative approach, such as GMM, in speech recognition and synthesis as well as in VC [18, 19]. Furthermore, the shape of the vocal tract is generally non-linear and compatible with NNs, whose conversion function is also non-linear. For these reasons, NN-based approaches achieve relatively high performance [17].

Meanwhile, Hinton et el. introduced an effective training algorithm of Deep Belief Nets (DBNs) in 2006 [20], and the use of DBNs rapidly spread in the field of signal processing with great success. DBNs and related models have been used, for example, for hand-written character recognition [21], 3-D object recognition [22], machine transliteration [23], and speech recognition tasks [24]. DBNs are probabilistic generative models that are composed of multiple layers of stochastic latent variables, and have a greedy layer-wise unsupervised learning algorithm. Since DBNs stack self-discovering extractors of abstractions (called Restricted Boltzmann Machines; RBMs) in a deep architecture, they can capture the fundamental bases to express the input vector at the highest layer.

In this paper, we propose a voice conversion technique using a combination of speaker-dependent DBNs and concatenating NNs. In our approach, we first train two exclusive DBNs for source and target speakers to obtain the deep networks that capture abstractions for each speech. Since the training data for the source speaker DBNs, for instance, include various phonemes of the speaker, the DBNs try to capture the abstractions to maximally express the training data that have abundant speaker individuality information and less phonological information. Therefore, we can expect that it is easier to convert the feature vectors in these speaker-individuality-emphasized high-order spaces than the original cepstrum space. At this point, we employ NNs to connect the highest layers of the DBNs. The input source signal is converted through the trained NNs in the high-order space, and brought back to the cepstrum space using the inverse process (reconstruction) of the target DBNs.

This paper presents the following: in Section 2, we explain our proposed voice conversion method. We show our setup and

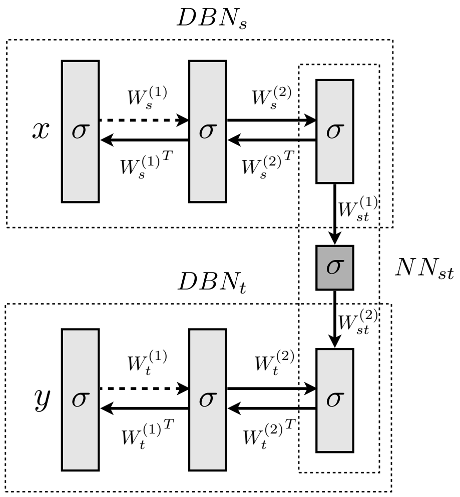

Figure 1: Our proposed voice conversion architecture, combined with two different DBNs and concatenating NNs. A source feature vector x is fed to DBN_x, NN_{xy}, and DBN_y in order, and then converted to a target vector y. This figure shows an example of architectures with two hidden layers in the DBNs and with one hidden layer in the NNs. σ indicates a standard sigmoid function, i.e. $\sigma(x) = 1./(1 + \exp(-x))$.

experimental results in Section 3, and Section 4 is our conclusion.

2. Methodology

2.1. Voice conversion using DBNs and NNs

Fig. 1 shows a flow of our proposed method, where an input vector (cepstrum) of the source speaker is converted to that of a target speaker in the high-order space by using DBNs and NNs. We prepare different DBNs for source speech and target speech (DBN_s and DBN_t, respectively) so as to capture the speaker-individuality information. All the DBNs are trained using the corresponding speaker's training data. As shown in Fig. 1, DBNs stack multiple layers (L layers) and share the weights bottom-up and top-down[1]. Given weight parameter matrices $\boldsymbol{W}_s^{(l)}$ and $\boldsymbol{W}_t^{(l)} (l = 1, 2, \ldots, L)$ for DBN_s and DBN_t, respectively, bottom-up conversion functions $\zeta_s(\boldsymbol{z})$ and $\zeta_t(\boldsymbol{z})$ can be represented by:

$$
\begin{aligned}
\zeta_i(\boldsymbol{z}) &= (\zeta_i^{(1)} \circ \zeta_i^{(2)} \circ \cdots \circ \zeta_i^{(L)})(\boldsymbol{z}) && (1)\\
&= \bigodot_{l=1}^{L} \zeta_i^{(l)}(\boldsymbol{z}) && (2)\\
\zeta_i^{(l)}(\boldsymbol{z}) &= \sigma(\boldsymbol{W}_i^{(l)}\boldsymbol{z}), \quad i \in \{s, t\} && (3)
\end{aligned}
$$

where $\bigodot_{l=1}^{L}$ denotes composition of L functions. For instance, $\bigodot_{l=1}^{2} \zeta_s^{(l)}(\boldsymbol{z}) = \sigma(\boldsymbol{W}_s^{(2)}\sigma(\boldsymbol{W}_s^{(1)}\boldsymbol{z}))$.

Similarly, given a high-order feature vector at the highest layer, a top-down conversion function $\zeta_i^{-1}(\boldsymbol{z})$ that brings it back to the original (cepstrum) space is given by:

$$
\zeta_i^{-1}(\boldsymbol{z}) = \bigodot_{l=1}^{L} \sigma(\boldsymbol{W}_i^{(L-l+1)^T}\boldsymbol{z}). \qquad (4)
$$

In our approach, the compact-represented input vector calculated by Eq. (1) is converted into the high-order target space using $(I + 1)$ layers perceptron NN_{st} (in Fig. 1). Once the weight parameters $\boldsymbol{W}_{st}^{(l)} (l = 1, 2, \ldots, I)$ of NN_{st} are estimated beforehand, an input vector can be converted to:

$$
\eta_{st}(\boldsymbol{z}) = \bigodot_{l=1}^{I} \sigma(\boldsymbol{W}_{st}^{(i)}\boldsymbol{z}) \qquad (5)
$$

Summarizing the above, a conversion function of our method from a source speech $\boldsymbol{x}$ to a target speech $\boldsymbol{y}$ is given by:

$$
\begin{aligned}
\boldsymbol{y} &= \zeta_t^{-1}(\eta_{st}(\zeta_s(\boldsymbol{x}))) && (6)\\
&= \bigodot_{l=1}^{2L+I} \sigma(\boldsymbol{W}^{(l)}\boldsymbol{x}) && (7)
\end{aligned}
$$

where $\boldsymbol{W}^{(l)}$ denotes an element of a set of weight parameters $\boldsymbol{W}$, where

$$
\begin{aligned}
\boldsymbol{W} &= \{\boldsymbol{W}^{(l)}\}_{l=1}^{2L+1} && (8)\\
&= \{\boldsymbol{W}_s^{(1)}, \cdots, \boldsymbol{W}_s^{(L)}, \boldsymbol{W}_{st}^{(1)}, \cdots, && (9)\\
&\qquad \boldsymbol{W}_{st}^{(I)}, \boldsymbol{W}_t^{(L)^T}, \cdots, \boldsymbol{W}_t^{(1)^T}\}. && (10)
\end{aligned}
$$

As Eq. (7) indicates, our conversion method is based on the composite function of multiple different non-linear functions. On the other hand, a conventional GMM-based approach with M Gaussian mixtures converts the source features $\boldsymbol{x}$ as

$$
\begin{aligned}
\boldsymbol{y} &= \sum_{m=1}^{M} P(m|\boldsymbol{x})\boldsymbol{E}_m^{(y)} && (11)\\
P(m|\boldsymbol{x}) &= \frac{w_m \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_m^{(x)}, \boldsymbol{\Sigma}_m^{(xx)})}{\sum_{m=1}^{M} w_m \mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}_m^{(x)}, \boldsymbol{\Sigma}_m^{(xx)})} && (12)\\
\boldsymbol{E}_m^{(y)} &= \boldsymbol{\mu}_m^{(y)} + \boldsymbol{\Sigma}_m^{(yx)}\boldsymbol{\Sigma}_m^{(xx)-1}(\boldsymbol{x} - \boldsymbol{\mu}_m^{(x)}) && (13)
\end{aligned}
$$

showing it to be an additive model of non-linear functions. Therefore, it is expected that our compositive model has a richer expression than the conventional GMM-based method.

2.2. Training the networks

DBNs have an architecture that stacks multiple Restricted Boltzmann Machines (RBMs) which compose a visible layer and a hidden layer. In the training, parameters of DBNs are determined layer-by-layer: from the lowest layer of the RBM to the highest. For each RBM, there are no connections among visible units or hidden units, but bidirectional connections between visible units and hidden units. In the literature of RBMs, the joint probability $p(\boldsymbol{v}, \boldsymbol{h})$ of binary-valued visible units $\boldsymbol{v} = [v_1, \cdots, v_I]^T, v_i \in \{0, 1\}$ and binary-valued hidden units $\boldsymbol{h} = [h_1, \cdots, h_J]^T, h_j \in \{0, 1\}$ is defined as:

$$
\begin{aligned}
p(\boldsymbol{v}, \boldsymbol{h}) &= \frac{1}{Z}\exp(-E(\boldsymbol{v}, \boldsymbol{h})) && (14)\\
E(\boldsymbol{v}, \boldsymbol{h}) &= -\boldsymbol{b}^T\boldsymbol{v} - \boldsymbol{c}^T\boldsymbol{h} - \boldsymbol{v}^T\boldsymbol{W}\boldsymbol{h} && (15)\\
Z &= \sum_{\boldsymbol{v}, \boldsymbol{h}} \exp(-E(\boldsymbol{v}, \boldsymbol{h})) && (16)
\end{aligned}
$$

[1] Technically, each stack is not a bidirectional model except for the highest layer; however, the architecture is approximately regarded as being a bidirectional model in this paper.

where, $\boldsymbol{W} \in \mathbb{R}^{I \times J}$, $\boldsymbol{b} \in \mathbb{R}^{I \times 1}$, and $\boldsymbol{c} \in \mathbb{R}^{J \times 1}$ are the weight-parameter matrix between visible units and hidden units, a bias vector of visible units, and a bias vector of hidden units, respectively.

Since there are no connections between visible units or between hidden units, the conditional probabilities $p(\boldsymbol{h}|\boldsymbol{v})$ and $p(\boldsymbol{v}|\boldsymbol{h})$ form simple equations as follows:

$$
\begin{aligned}
p(h_j = 1|\boldsymbol{v}) &= \sigma(c_j + \boldsymbol{v}^T \boldsymbol{W}_{:j}) && (17) \\
p(v_i = 1|\boldsymbol{h}) &= \sigma(b_i + \boldsymbol{h}^T \boldsymbol{W}_{i:}^T) && (18)
\end{aligned}
$$

where $\boldsymbol{W}_{:j}$ and $\boldsymbol{W}_{i:}$ denote the j-th column vector and the i-th row vector, respectively. Eqs. (17) and (18) show that each layer has a non-linear activation of sigmoid function, in accord with Fig. 1 and Eqs. (3) and (4).

For parameter estimation, the log likelihood of visible units is used as an evaluation function. Differentiating partially with respect to each parameter, we obtain:

$$
\begin{aligned}
\frac{\partial \log p(\boldsymbol{v})}{\partial w_{ij}} &= \langle v_i h_j \rangle_{data} - \langle v_i h_j \rangle_{model} && (19) \\
\frac{\partial \log p(\boldsymbol{v})}{\partial b_i} &= \langle v_i \rangle_{data} - \langle v_i \rangle_{model} && (20) \\
\frac{\partial \log p(\boldsymbol{v})}{\partial c_j} &= \langle h_j \rangle_{data} - \langle h_j \rangle_{model} && (21)
\end{aligned}
$$

where, $\langle \cdot \rangle_{data}$ and $\langle \cdot \rangle_{model}$ indicate expectations of input data and the inner model, respectively. However, it is generally hard to compute the second term. Typically, expectation of the reconstructed data $\langle \cdot \rangle_{recon}$ computed by Eqs. (17) and (18) is alternatively used [20]. Using Eqs. (19), (20) and (21), each parameter can be updated by stochastic gradient descent.

In the training of DBNs, the hidden units of the current stack are regarded as visible units in the next layer. In other words, the hidden units computed as a conditional probability $p(\boldsymbol{h}|\boldsymbol{v})$ in Eq. (17) are fed to the following RBMs, and trained in the similar way. This procedure is repeated layer-by-layer until the highest layer is reached.

After training two DBNs for source and target speakers, we train a converting-in-high-order-space NN_{st} using parallel speeches $\{\boldsymbol{x}_n, \boldsymbol{y}_n\}_{n=1}^N$ of source/target speakers. Weight parameters of NN_{st} are estimated so as to minimize the error between the output $\eta_{st}(\zeta_s(\boldsymbol{x}_n))$ and the target vector $\zeta_t(\boldsymbol{y}_n)$. Finally, each parameter of the whole networks (DBN_s, DBN_t and NN_{st}) is fine-tuned by back-propagation using the raw parallel data.

2.3. Pre-processing

The above-mentioned DBNs (or RBMs) as shown in Eqs. (15) and (18) are modeled under the assumption that each visible unit is binary. Therefore, real-valued data, such as cepstrum features, do not suit the training of DBNs. There is an approach that supports a real-valued input by modeling in which each visible unit is sampled from a Gaussian distribution [20]; however, we took another approach based on soft-binarization. The input vectors (both for the source speaker and the target speaker) are first normalized to have zero mean and unit variance for each dimension, and then binarized using an element-wise sigmoid function $\sigma_\alpha(\boldsymbol{x})$ as:

$$\boldsymbol{x} \leftarrow \sigma_\alpha(\boldsymbol{x}) = 1./(1 + \exp(-\alpha \boldsymbol{x})) \tag{22}$$

where, α ($= 2$, in this paper) indicates a gain parameter.

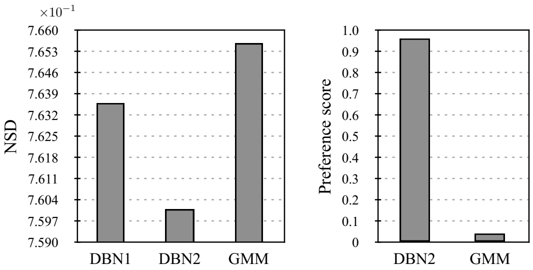

Figure 2: Normalized spectrum distortion calculated from converted speech using each method (left), and preference score to see auditory measure (right).

3. Experiments

3.1. Setup

We conducted voice conversion experiments using the ATR Japanese speech database [25], comparing our method with the conventional GMM-based approach. From this database, we selected and used a male speaker (MMY) for the source, and a female speaker (FTK) for the target.

For the training and validation set, we resampled the acoustic signals to 8 kHz from their original 20 kHz, extracted STRAIGHT parameters, and used D cepstrum coefficients ($D = 40$ except for the energy features, in this paper) computed from the STRAIGHT parameters. The parallel data of the source/target speakers processed by Dynamic Programming were created from 216 word speeches in the dataset, and used for the training of DBNs and NNs (and GMM for the conventional method). Note that the parallel data was prepared for NNs (and GMM), and two DBNs were trained independently. The number of the training data (frames) was 189,992 (about 63 min.). We let the learning rate and the number of epochs in the gradient decent-based training of DBNs be 0.05 and 50, respectively. We tested two different architectures of DBNs for reference: (DBN1) L=1,I=1, and deeper (DBN2) L=2,I=1. The numbers of each node from input $\boldsymbol{x}$ to output $\boldsymbol{y}$ in Fig. 1 were [40 80 80 40] for DBN1, and [40 120 30 30 120 40] for DBN2. For the conventional method, we used a GMM with 64 mixtures and a diagonal covariance matrix. For the validation, 15 sentences (about 52 sec.) were arbitrarily selected from the database.

We converted only spectrum features using DBNs (or GMM), and pitches using a traditional linear conversion with mean and standard deviation. Energy features (0-order cepstrum coefficients) of the source signal were used for the target signal synthesis without change.

3.2. Results and discussion

Fig. 2 summarizes the experimental results, showing the comparison of our method with the conventional GMM method with respect to objective and subjective metrics. For the objective metric, we used normalized spectrum distortion (NSD), calculated by:

$$NSD = \sqrt{\frac{\|S^Y - \hat{S^X}\|^2}{\|S^Y - S^X\|^2}} \tag{23}$$

where, S^X, S^Y, and $\hat{S^X}$ denote STRAIGHT spectra (a matrix of spectra by time sequence) of the source speaker, spectra of the target speaker, and the converted spectra using each

Table 1: Smoothness of spectra in time axis ($\times 10^{-2}$).

DBN2	GMM	Target
7.80881	10.3204	6.05998

method. The smaller the value of NSD is, the closer the converted spectra is to the target spectra. As shown in Fig. 2 (left side), our approach (DBN1 and DBN2) outperformed the conventional GMM-based method. When we compare within our methods, the deeper architecture (DBN2) achieved better performance than the shallower architecture (DBN1). The reason for the improvement can be considered to the result of the fact that our deep conversion system (DBN2) could capture and convert the abstractions of speaker individualities better than other methods.

We also carried out preference tests related to the naturalness of the converted speech. For the evaluation, a paired comparison test was carried out, where each subject (in total there were 9 subjects) listened to pairs of speech converted by the two methods (DBN2 and GMM)[2] and selected which sample sounded more natural. The results of the preference tests are shown in the right-hand portion of Fig. 2. As shown in the figure, our approach performed much better than the conventional method in the subjective criteria, compared with the case of the objective evaluation. This might be because DBNs produced auditory superior converted speech compared to that produced by GMM. In order to examine this, we compared the methods with their smoothness as shown in Table 1, calculated by the norm of the spectral gradient in the time axis. The smaller the value of the smoothness is, the smoother the converted spectrum is. As shown in Table 1, the converted spectra using our approach is smoother than that of GMM, being close to the target spectra. The spectra converted by GMM are somewhat jagged, and hence it negatively affected the auditory evaluation.

4. Conclusion

In this paper, we presented a voice conversion technique using Deep Belief Nets (DBNs) for generation of the high-order eigen space of the speaker, where it captures the abstractions of speaker individuality. Our experimental results showed the efficacy of the proposed method, in comparison to a conventional GMM-based approach. Future work will include the use of Deep Boltzmann Machines (which have a deep bidirectional model), instead of DBNs, to improve the synthesis accuracy.

5. References

[1] A. Kain and M. W. Macon, "Spectral voice conversion for text-to-speech synthesis," Proc. ICASSP, vol. 1, pp. 285—288, 1998.

[2] C. Veaux and X. Robet, "Intonation conversion from neutral to expressive speech," in Proc. INTERSPEECH, pp. 2765—2768, 2011.

[3] K. Nakamura, T. Toda, H. Saruwatari, and K. Shikano, "Speaking-aid systems using GMM- based voice conversion for electrolaryngeal speech," Speech Communication, Vol. 54, No. 1, pp. 134—146, 2012.

[4] L. Deng, A. Acero, L. Jiang, J. Droppo, and X. Huang, "High-performance robust speech recognition using stereo training data," Proc. ICASSP, pp. 301—304, 2001.

[5] A. Kunikoshi, Y. Qiao, N. Minematsu, and K. Hirose, "Speech generation from hand gestures based on space mapping," Proc. INTERSPEECH, pp. 308—311, 2009.

[6] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Vice conversion through vector quantization," in Proc. ICASSP, pp. 655—658, 1988.

[7] H. Valbret, E. Moulines and J. P. Tubach, "Voice transformation using PSOLA technique," Speech Communication, Vol. 11, No. 2-3, pp. 175—187, 1992.

[8] Y. Stylianou, O. Cappe, and E. Moulines, "Continuous probabilistic transform for voice conversion," IEEE Trans. Speech and Audio Processing, Vol. 6, No. 2, pp. 131—142, 1998.

[9] T. Toda, A. Black, and K. Tokuda, "Voice conversion based on maximum likelihood estimation of spectral parameter trajectory," IEEE Trans. Audio, Speech, Lang. Process., Vol. 15, No. 8, pp. 2222—2235, 2007.

[10] E. Helander, T. Virtanen, J. Nurminen, and M. Gabbouj, "Voice conversion using partial least squares regression," IEEE Trans. Audo, Speech, Lang. Process., Vol. 18, No. 5, pp. 912—921, 2010.

[11] C. H. Lee and C. H. Wu, "Map-based adaptation for speech conversion using adaptation data selection and non-parallel training," in Proc. INTERSPEECH, pp. 2254—2257, 2006.

[12] D. Saito, S. Watanabe, A. Nakamura, N. Minematsu, "Voice conversion based on probabilistic integration of joint density model and speaker model," in Proc. Acoustic Society of Japan, pp. 335–338, 2010.

[13] T. Toda, Y. Ohtani, and K. Shikano, "Eigenvoice conversion based on Gaussian mixture model," in Proc. INTERSPEECH, pp. 2446—2449, 2006.

[14] D. Saito, K. Yamamoto, N. Minematsu, and K. Hirose, "One-to-many voice conversion based on tensor representation of speaker space," in Proc. INTERSPEECH, pp. 653—656, 2011.

[15] Z. H. Jian and Z. Yang, "Voice conversion using canonical correlation analysis based on gaussian mixture model," SNPD, Vol. 1, pp. 210–215, 2007.

[16] R. Takashima, T. Takiguchi, Y. Ariki, "Exemplar-based voice conversion in noisy environment," SLT, pp.313–317, 2012.

[17] S. Desai, E. V. Raghavendra, B. Yegnanarayana, A. W. Black, and K. Prahallad, "Voice conversion using artificial neural networks," in Proc. ICASSP, pp. 3893—3896, 2009.

[18] Y. J. Wu, H. Kawai, J. Ni, and R. H. Wang, "Minimum segmentation error based discriminative training for speech synthesis application," in Proc. ICASSP 04, vol. 1, pp. 629-32, 2004.

[19] E. McDermott, T. Hazen, J. L. Roux, A. Nakamura, and S. Katagiri, "Discriminative training for large vocabulary speech recognition using minimum classification error," IEEE Transactions on Speech and Audio Processing, vol. 15, no. 1, pp. 203-223, 2007.

[20] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," Neural Computation, vol. 18, pp. 1527—1554, 2006.

[21] G. E. Hinton, S. Osindero, and Y. Teh, "A fast learning algorithm for deep belief nets," Neural Computation, vol. 18, pp. 1527—1554, 2006.

[22] V. Nair and G. Hinton, "3-d object recognition with deep belief nets.," in To appear in Advances in Neural Information Processing Systems 22, 2009.

[23] T Deselaers, S. Hasan, O. Bender, and H. Ney, "A deep learning approach to machine transliteration," in Proc. EACL Workshop on Statistical Machine Translation, 2009, pp. 233—241.

[24] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic Modeling using Deep Belief Networks," IEEE Trans. on Audio, Speech, and Language Procesing, vol. 20, no. 1, pp. 14-22, 2012.

[25] A. Kurematsu, K. Takeda, Y. Sagisaka, S. Katagiri, H. Kuwabara, and K.Shikano, "ATR Japanese speech database as a tool of speech recognition and synthesis," Speech Communication, vol. 9, pp. 357—363, 1990.

[2] We tested only the two methods for the subjective evaluation, since we got the best results with DBN2 of our approach in the objective evaluation (NSD).