Non-native speech conversion with consistency-aware recursive network and generative adversarial network

Keisuke Oyamada* , Hirokazu Kameoka† , Takuhiro KANEKO† ,
Hiroyasu ANDO* ‡, Kaoru HIRAMATSU† , Kunio KASHINO†
* Division of Policy and Planning Sciences, Faculty of Engineering, Information and Systems , University of Tsukuba, Japan
E-mail: s1720584@s.tsukuba.ac.jp
† NTT Communication Science Laboratories, NTT Corporation, Japan
‡ Center for Artificial Intelligent Research, University of Tsukuba, Japan

***Abstract*—This paper deals with the problem of automatically correcting the pronunciation of non-native speakers. Since the pronunciation characteristics of non-native speakers depend heavily on the context (such as words), conversion rules for correcting pronunciation should be learned from a sequence of features rather than a single-frame feature. For the online conversion of local sequences of features, we construct a neural network (NN) that takes a sequence of features as an input/output, generates a sequence of features in a segment-by-segment fashion and guarantees the consistency of the generated features within overlapped segments. Futhermore, we apply a recently proposed generative adversarial network (GAN)-based postfilter to the generated feature sequence with the aim of synthesizing natural-sounding speech. Through subjective and quantitative evaluations, we confirmed the superiority of our proposed method over a conventional NN approach in terms of conversion quality.**

I. Introduction

This paper deals with the problem of automatically correcting the pronunciation of non-native speakers. We are particularly interested in developing a real-time (or low-latency) pronunciation-conversion system, which converts non-native speakers into intelligible native-like speech in an online manner. The problem of converting non/para-linguistic speech information while preserving linguistic information is called voice conversion (VC). Since pronunciation quality is in a broad sense part of a speaker's identity, the pronunciation-conversion problem can be seen as a special class of VC problem.

Many VC methods have been proposed during the last two decades [1]. One successful VC framework involves statistical methods based on Gaussian mixture models (GMMs) [2], [3]. Recently, neural network (NN)-based methods using restricted Boltzmann machines (RBMs) [4] and deep neural networks (DNNs) [5], [6], and exemplar-based methods using non-negative matrix factorization (NMF) [7], [8] have been proposed with notable success. These methods, with some exceptions, typically consist of training a mapping function between the acoustic features of the source and target speech using parallel data, i.e., a pair of time-aligned feature sequences of source and target speech. At test time, the feature sequence of the input speech is converted using the trained mapping function. One common problem as regards these methods is that the converted features tend to be oversmoothed and that the oversmoothing of feature sequences causes perceptual quality degradation. This is caused by a side effect of assuming a particular form of similarity metric (e.g., mean squared error (MSE)) or distribution (e.g., Gaussian) for the parameter training of the acoustic model so that the generated feature sequence that on average fits the training target example is considered optimal. Post-filtering methods based on global variance [3], [9] or the modulation spectrum [10] have been proposed with the aim of reconstructing the original spectro-temporal fine texture. However, these methods rely on heuristics since it is usually difficult to model the exact probability density of the features of real speech. Recently, a learning-based postfiltering method using a generative adversarial network (GAN) [11] was proposed [12]. A GAN offers a framework for training a generator in such a way that it can deceive a discriminator that tries to distinguish real data from samples drawn from the generator. The GAN postfiltering method allows us to add a high-fidelity spectro-temporal texture to a feature sequence obtained with parametric speech synthesis or VC so that the synthesized speech becomes as indistinguishable as possible from real speech. Furthermore, some attempts have recently been made to directly apply the GAN framework to VC tasks [13].

Since the aim of conventional VC methods (GMM-based [3] and DNN-based [6] methods) is to convert the speaker identity of source speech, it is not trivial that these methods also work successfully on a pronunciation-conversion task. Hence, we conducted a preliminary experiment to see how these methods behave on a pronunciation-conversion task by using English utterances spoken by a non-native Indian speaker and a native American speaker as the source and target speech, respectively. This experiment revealed that while the perceived speaker identity was successfully converted, the pronunciation quality

did not improve as much as we expected. We believe that this is because these methods are designed to focus only on a local segment of parallel feature sequences to determine a conversion rule. When it comes to a pronunciation-conversion task, we must consider the contextual or temporal dependencies of the conversion rules since the way speakers pronounce each phoneme may differ depending on the preceding, current, and succeeding contexts. For example, consider a speaker who pronounces *"thing"* as *"/ ting /"*. In this case, it is difficult for a VC system to decide whether to convert *"/ t /"* to *"/ th /"* or leave it as is unless it knows that the spoken word is *"thing"*. A typical way of modeling long-term dependencies would be to use recurrent NNs (RNNs) [4] including long short-term memory (LSTM) networks [14]. However, RNNs are not well suited to parallel implementations, thus, the conversion process becomes computationally demanding. In [15] Toda et al. proposed a GMM-based VC method, which takes as its input a vector (or its dimensionally compressed version) formed by concatenating acoustic features within a segment of consecutive frames. This feature is called a "segmental feature". Inspired by this work, in this paper we propose an NN-based VC system that takes segmental features with a sufficiently long segment as both the input and output of the NN as a reasonable way of modeling long-term dependencies. In addition, it allows us to perform a segment-by-segment conversion in an online manner. Here, when consecutive segments do not overlap, a generated feature sequence may become discontinuous at the boundary of the segments. If we let each segment overlap, we must somehow take an average of the feature sequences within the overlapping segments. This may cause oversmoothing unless the feature sequences within the overlapping segments are perfectly consistent. To address this, we further propose a recursive network that guarantees the consistency of the feature sequences within the overlapping segments. In addition, we apply the GAN-based postfiltering method proposed in [12] to a feature sequence converted using the proposed recursive network.

II. Related work

Mohammadi et al. proposed an NN-based VC system using a DNN with a bottleneck structure [6]. This system takes single-frame mel-cepstral features as the input and output of the DNN and performs frame-by-frame conversion. To efficiently train the DNN parameters, it uses a stacked auto-encoder (AE), constructed by stacking single layer AEs so that the hidden layer of one AE becomes the input layer of another AE. The output of each AE is given as

$$\hat{x} = h_\theta(x), \tag{1}$$

where x is the input, $\hat{x}$ is the output and h_θ is a non-liner function parameterized by θ. The AE parameter θ is typically trained by minimizing the divergence between x and $\hat{x}$, e.g., a squared error

$$L(\theta) = \frac{1}{2}||x - \hat{x}||^2. \tag{2}$$

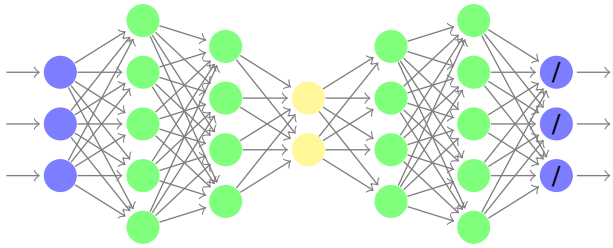

Fig. 1. **Stacked auto-encoder [6]**

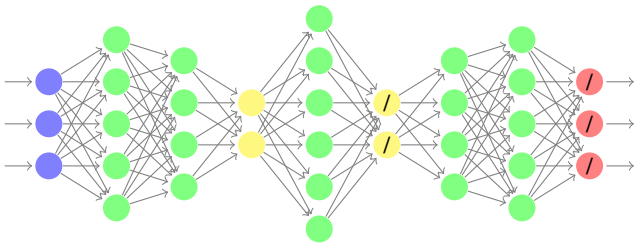

Fig. 2. **DNN structure in [6]**

Since the aim of AEs is to obtain a compact representation of the input vector x by reducing redundant information underlying x, we usually do not want h_θ to become an identity mapping. Therefore, the dimension of the hidden layer is often set smaller than the input dimension. After stacking the trained AEs and inserting an additional hidden (conversion) layer in the middle of the stacked AE, all the parameters can be trained efficiently via global fine-tuning.

While using low-dimensional features such as single-frame mel-cepstral features as the input and output features is advantageous for efficient parameter training, it causes difficulties as regards the long-term dependencies of conversion rules and ensuring the local and global continuity of a generated feature sequence.

III. Consistency-aware recursive network

A. Basic idea

As a reasonable way of modeling a conversion rule with long-term dependencies, we propose an NN-based VC system that takes segmental features with a sufficiently long segment as both the input and output of the NN and makes it possible to perform a segment-by-segment conversion in an online manner. We further propose a recursive network that guarantees the consistency of the feature sequences within the overlapping segments.

If there is no overlap between one segment and its neighboring segments, generated feature sequences can become discontinuous at the segment boundaries. Therefore, it would be reasonable to let consecutive segments overlap each other. The center figure of Fig. 3 shows a case where there are 10 frames within each segment with a 5-frame overlap, whereas the right figure shows a case without an overlap. To guarantee the continuity of the generated feature sequence as a whole, we

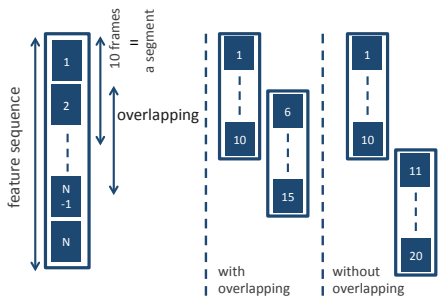

Fig. 3. **Example of frame segmentation.** The left figure shows a feature sequence extracted from speech. The center figure shows the first two segmental features extracted from a 10-frame long feature sequence with a 5-frame overlap. The right figure shows a case with no overlap.

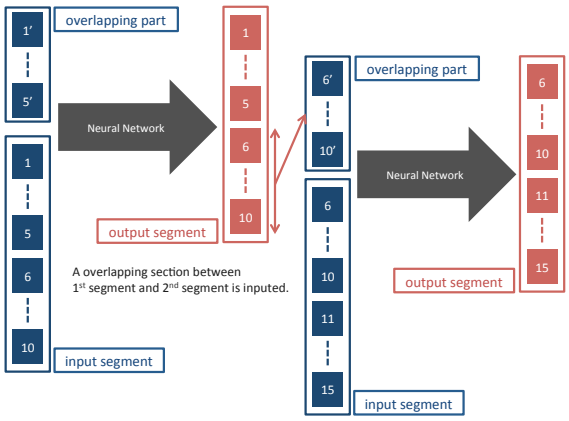

Fig. 4. **Structure of consistency-aware recursive network.** There are 10 frames in a segment and 5 overlapped frames. The blue segments are the segment features for the input and the red segments are for the output. The number in each frame indicates the index of the feature.

must ensure that the generated features within the overlapping segment of two consecutive segments are consistent. To this end, as shown in the center figure of Fig. 4, we take a segmental feature of source speech as an input to the NN along with the part of the segmental feature of the target speech corresponding to the overlapping segment. In this way, we can let the NN learn to convert a segmental feature of source speech so that the converted feature becomes as close as possible to both the segmental feature of the target speech and the part of the output the NN has just generated at the previous segment. Fig. 3 shows an example where the 6th to 10th frames are shared by the first and second segments. When converting the segmental feature at the second segment, the acoustic features corresponding to the 6th to 10th frames the NN has generated at the first segment are fed into the NN in addition to the source segmental feature of the current segment. The segmental features are generated recursively so the approach is suitable for online processing.

B. Training

Let $f_1^x, \ldots, f_N$ and $f_1^y, \ldots, f_N^y$ be the time-aligned acoustic feature sequences of source and target speech, respectively, where N is the number of frames. We divide these sequences

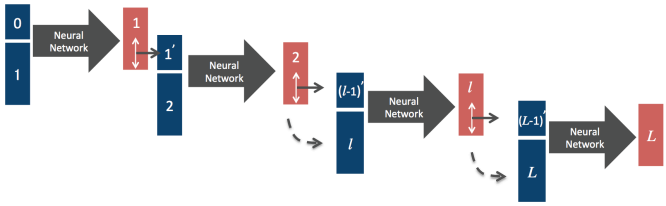

Fig. 5. **Overview of consistency-aware recursive network.** The whole network has a staircase shape. The outputs go to the inputs.

into L segments and define $s_l^x = ({f_{N_l^0}^x}^\mathsf{T}, \ldots, {f_{N_l^1}^x}^\mathsf{T})^\mathsf{T}$ and $s_l^y = ({f_{N_l^0}^y}^\mathsf{T}, \ldots, {f_{N_l^1}^y}^\mathsf{T})^\mathsf{T}$ as the segmental feature at segment l where N_l^0 and N_l^1 denote the indices of the first and last frames in the l-th segment. We use $\mathcal{N}_{l-1}$ to denote the set of frames within the segment shared by segment $l-1$ and segment l. We hereafter call it a *subsegment* in segment $l-1$. We consider an NN that takes s_l^x and $[s_{l-1}^y]_{\mathcal{N}_{l-1}}$ as inputs and generates $\hat{s}_l^y$ at segment l where $[s_{l-1}^y]_{\mathcal{N}_{l-1}}$ denotes the part of the target segmental feature s_{l-1}^y corresponding to the subsegment $\mathcal{N}_{l-1}$:

$$\hat{s}_l^y = g_\theta(s_l^x,\ [s_{l-1}^y]_{\mathcal{N}_{l-1}}), \tag{3}$$

where θ denotes the parameter set of the NN. Since the goal is to find θ such that $s_l^y \simeq \hat{s}_l^y$, through parameter training this NN is forced to learn a conversion rule that makes the output $\hat{s}_l^y$ as close as possible to the second argument of g_θ within the subsegment $\mathcal{N}_{l-1}$ as well as to the target segmental feature s_l^y within the entire segment.

As a learning criterion $L(\theta)$, we use the squared error between $\hat{s}_l^y$ and s_l^y:

$$L(\theta) = \sum_l D_{\mathrm{EU}}(s_l^y|\hat{s}_l^y), \tag{4}$$

$$D_{\mathrm{EU}}(x|y) = \frac{1}{2}\|x-y\|_2^2. \tag{5}$$

Note that we can also use the Kullback-Leibler (KL) divergence or the Itakura-Saito divergence when we use a magnitude/power spectrum as the acoustic feature in place of D_{EU}.

As with the method in [6], we can use a stacked AE to efficiently train θ. The output $\hat{s}_l^x$ of the AE is given by

$$\hat{s}_l^x = h_\phi(s_l^x) \tag{6}$$

or

$$\hat{s}_l^x = h_\phi(s_l^x,\ [s_{l-1}^y]_{\mathcal{N}_{l-1}}). \tag{7}$$

We search for ϕ such that $\hat{s}_l^x \simeq s_l^x$ in a layerwise fashion. After stacking the trained AEs and inserting an additional hidden (conversion) layer in the middle of the stacked AE, all the parameters can be trained efficiently via global fine-tuning.

C. Conversion

Once θ is trained, the segmental feature $\hat{s}^y$ of the converted speech can be computed recursively via

$$\hat{s}_l^y = g_\theta(s_l^x, [\hat{s}_{l-1}^y]_{\mathcal{N}_{l-1}})\ (l = 1, \ldots, L), \tag{8}$$

where $[\hat{s}^y_{l-1}]_{\mathcal{N}_{l-1}}$ is a part corresponding to the subsegment $\mathcal{N}_{l-1}$ of the segmental feature $\hat{s}^y_{l-1}$ the NN has just generated at segment $l-1$. Owing to the nature of the proposed NN, we expect that $\hat{s}^x_l$ and $\hat{s}^y_{l-1}$ will become similar within the subsegment $\mathcal{N}_{l-1}$. Hence, we can constitute an entire feature sequence by taking their average or median segment-by-segment. After a feature sequence is obtained, we can convert it back to a time-domain signal either by using a vocoder or a phase reconstruction algorithm (when using a magnitude/power spectrum as the acoustic feature).

D. Acoustic feature

As the acoustic feature, we use a subset of a vocal tract spectral feature (e.g. mel-cepstral coefficients), a magnitude (or power) spectrum, a fundamental frequency (F_0), and an aperiodicity measure. The vocal tract spectral feature, the F_0, and the aperiodicity measure can be obtained by using STRAIGHT [16] or WORLD [17] followed by mel-cepstral analysis. A magnitude (or power) spectrum can be extracted via a short-time Fourier transform (STFT) or constant-Q transform (CQT). We can convert these features back into a time-domain signal either by using the STRAIGHT/WORLD/mel-cepstrum vocoder or a phase reconstruction algorithm [18], [19].

IV. GAN-based postfiltering

Since the proposed method uses an explicit form of loss function to optimize the conversion rule parameter, acoustic feature sequences generated by the converter will be over-smoothed, resulting in buzzy-sounding speech, as found with conventional statistical parametric speech synthesis or VC methods. To restore the spectro-temporal details and fill in the gap between the converted and real speech, we propose applying GAN-based postfiltering [12] to the converted speech to make it as indistinguishable as possible from real speech.

A GAN [11] offers a framework for estimating a random generator through the adversarial training of generator and discriminator networks, where the goal is to learn a generator distribution $P_G(x)$ that matches the true data distribution $P_{\mathrm{Data}}(x)$. The generator G maps noise variables $z \sim P_{\mathrm{Noise}}(z)$ to the data space $x = G(z)$ whereas the discriminator D assigns probability $p = D(x)$ when x is sampled from $P_{\mathrm{Data}}(x)$ and assigns probability $1-p$ when x is sampled from $P_G(x)$. D and G play a two-player minimax game using the following objective function:

$$\min_G \max_D \mathbb{E}_{x\sim P_{\mathrm{Data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim P_{\mathrm{Noise}}(z)}[\log(1 - D(G(z)))]. \quad (9)$$

Maximizing this objective with respect to D encourages D to find a binary classifier that provides the best possible discrimination between the real and generated data whereas minimizing this with respect to G encourages G to fit $P_G(x)$ to $P_{\mathrm{Data}}(x)$. Both G and D can be trained using back-propagation.

With the GAN-based postfiltering method [12], three modifications are made to the regular GAN architecture, namely the adoption of conditional [20], [21], residual [22] and convolutional [23] networks. We use a conditional GAN (CGAN) [20], [21], which is an extension of a GAN, where both G and D receive the additional data y as an input:

$$\min_G \max_D \mathbb{E}_{x,y\sim P_{\mathrm{Data}}(x,y)}[\log D(x,y)] + \mathbb{E}_{z\sim P_{\mathrm{Noise}}(z),y\sim P_y(y)}[\log(1 - D(G(z,y),y))]. \quad (10)$$

If we let y be an acoustic feature sequence of synthesized speech, $G(z,y)$ can be regarded as the postfilter. Thus, we can express this as a residual network

$$G(z,y) = y + R(z,y), \quad (11)$$

where R represents a residual spectro-temporal texture [22]. Based on the observation that a spectral texture is typically structured in both time and frequency directions, we use a convolutional architecture to determine the structure with a reasonably small number of parameters. In particular, we design G as a fully convolutional network (FCN) [23] that allows input segments to take an arbitrary length. Fig. 6 shows the network configuration of the proposed postfilter.

V. Experimental evaluation

A. Experiment setting

For experimental evaluations, we used English utterances spoken by a male non-native Indian speaker and a male native American speaker as the parallel data. Each set of data lasted for a total time of about one hour. 10% and 90% of the parallel data were used as the test and training data, respectively. The sampling frequency was 16 kHz and the frame shift was 5 ms. We used 25-dimensional mel-cepstral coefficients as the acoustic feature. The mel-cepstrum was extracted from the vocal tract spectra obtained with the STRAIGHT analysis. The proposed recursive NN was fully connected in all layers. As the activate functions we used a sigmoid function at the middle layer and an identity mapping at the output layer. Experiments were conducted with various settings for the number of frames in each subsegment and each segment. Table. I summarizes the number of units constructing the proposed NN and [6] in this experiment. In Table. I, "seg X" indicates that the number of frames in a segment is X and, "over Y" indicates that the number of overlapped frames in consecutive segments is Y.

The mel-cepstrum was normalized as its average is 0 and standard deviation is 1. Adam [24] was used as an optimization algorithm. We set $\alpha = 0.001$ for pre-training and $\alpha = 0.0001$ for fine-tuning. The conventional method in [6] was pre-trained by stacking the denoising auto-encoder [25] and the contractive auto-encoder [26]. The proposed NN was pre-trained by stacking the auto-encoder using dropout [27]. The numbers of epochs for pre-training and fine-tuning are 100 and 300, respectively. We used the model for which the test error was minimum.

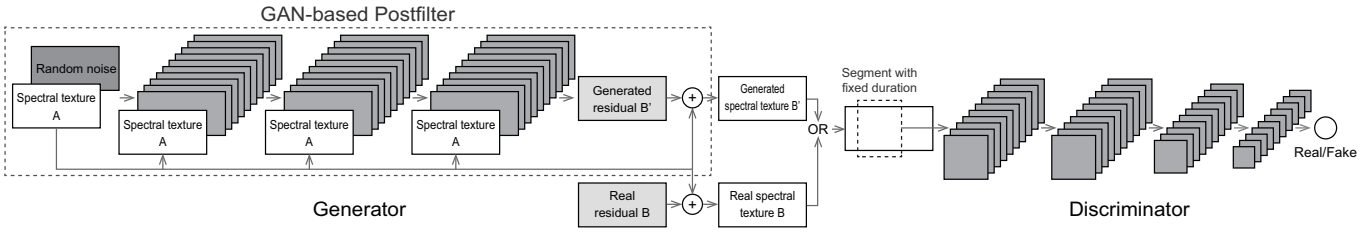

Fig. 6. **Network architecture for GAN-based postfiltering**

TABLE I
The number of units in each method. NUMBERS IN THE FIRST ROW SHOW THE INDEX OF HIDDEN LAYERS. THE FIRST COLUMN PROVIDES INFORMATION ABOUT EACH MODEL.

	input	1	2	3	4
[6]	25	100	40	15	150
seg10 over5	375	800	480	290	170
seg8 over4	300	640	380	230	140
seg6 over3	225	480	290	170	100
seg4 over2	150	320	190	110	70

5	6	7	8	9	output
15	40	100			25
680	170	290	480	800	250
540	140	230	380	640	200
400	100	170	290	480	150
300	70	110	190	320	100

Fig. 7. **Comparison of [6] and proposed NN.** Each line shows the time series transition of the 10th mel-cepstrum. The blue line corresponds to mel-cepstrum of native speech, the green line corresponds to that converted by the method in [6], and the red line corresponds to that converted by the proposed NN.

B. Objective Evaluations

1) Mel-cepstrum distortion: First, we evaluated each model by mel-cepstrum distortion. As for the data used in this experiment, the mel-cepstrum distortion between the source speech and the target speech was 8.69 dB. Therefore, this value was the upper limit of the distortion in this experiment. The mel-cepstrum distortion of features converted by [6] was 6.28 dB. Table. II shows, the mel-cepstrum distortions of features converted by the proposed NN.

TABLE II
Mel-cepstrum distortion for consistency-aware recursive networks.

seg10 over5	seg8 over4	seg6 over3	seg4 over2
6.17	6.27	6.20	6.25

In this experiment, the distortion with regards to the method in [6] and the proposed NN are lower than the upper limit (8.69 dB). Moreover, the distortions of features converted by the proposed NN with any number of frames in the segment and any number of overlapped frames are smaller than those with the method in [6]. When there are 10 frames in the segment and 5 overlapping frames, the distortion was the smallest among the four cases.

C. Evaluation about oversmoothing

In general, a converted spectrum envelope sequence tends to be oversmoothed. In this subsection, we show that the feature sequence can be recovered by the GAN-based post filter. Fig. 7 shows the feature sequences converted by the proposed NN and the method reported in [6]. As shown in Fig. 7, smoothing of the feature is observed in the both methods. Fig. 8 shows the converted features by the method in [6] and the proposed NN with GAN-based postfilter. It is clearly observed that the GAN-based postfilter improves the oversmoothing problem.

VI. SUBJECTIVE EVALUATIONS

In this section, the quality and similarity of the converted speech is evaluated subjectively.

A. Quality

The quality of speech converted by each method was evaluated by MOS. There were 5 subjects, and 30 speech utterances of 2 to 5 seconds in length extracted from the test data were used. Subjects were asked to evaluate 7 random samples for each method.

Table. III shows the results of a quality evaluation. In Table. III, 'only' indicates speech converted by each method without a postfilter, 'VS' indicates that converted with a postfilter by Variance Scaling [9], and 'GAN' indicates that converted with a GAN-based postfilter. The values in parentheses indicate the standard deviation. VS was introduced as a general postfilter method. There was no clear difference between the MOS values of speech converted with the method in [6] and with

Fig. 8. **Comparison of [6] and proposed NN with GAN-based postfilter.** Each line shows the time series transition of the 10th mel-cepstrum. The blue line corresponds to the mel-cepstrum of native speech, the green line corresponds to that converted by the method in [6], and the red line corresponds to that converted by the proposed NN with the GAN-based postfilter.

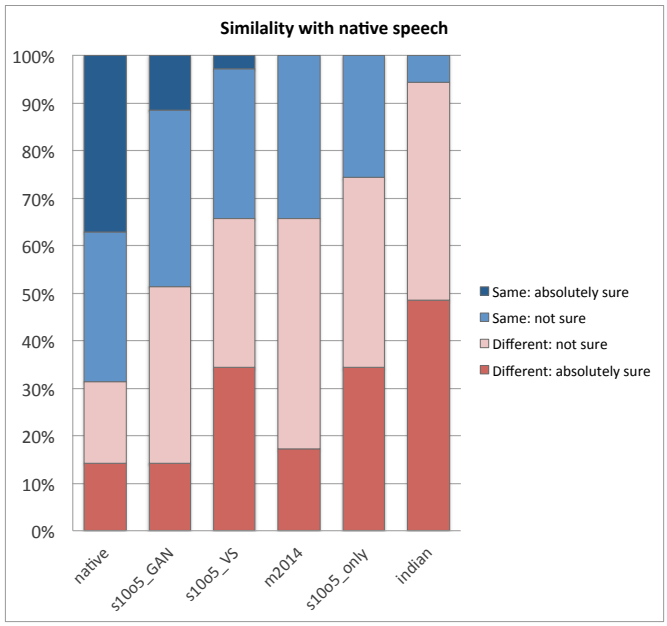

Fig. 9. **Subjective evaluation on similarity with native speech.** Each method is arranged in descending order of similarity.

the proposed NN. On the other hand, the quality of speech converted by the proposed NN with a GAN-based postfilter was better than that converted with [6]. This result confirms that the GAN-based postfilter improves the naturalness of converted speech. In addition, it is found that the MOS value of speech when the VS postfilter was applied was greater than that with GAN. Namely, although the speech with VS had a more mechanical sound than with GAN, the speech was less noisy with VS than with GAN. The difference between the MOS values of VS and GAN can be explained by the fact that the speech was evaluated in particular with respect to the amount of noise.

TABLE III
Subjective evaluation of quality

Methods	[6]	seg10 over5		
Post filter	only	only	GAN	VS
MOS	2.86 (0.68)	2.74 (0.91)	3.43 (1.05)	4.3 (0.72)

B. Similarity

The similarity of the converted speech was evaluated according to the evaluation method described in Voice Conversion Challenge 2016 [28]. With the evaluation method, the similarity was evaluated for pairs consisting of source speech and source speech, target speech and target speech, source speech and target speech, source speech and converted speech, target speech and converted speech. In the evaluation of the source speech and converted speech, a preferable evaluation was 'different'. In this research, the source speech was English spoken by Indians and the target speech was English spoken by native speakers.

Fig. 9 shows the results of evaluating similarity with respect to native speech and converted speech. The blue areas in the bars indicate the rate of evaluations where speech converted by the method is similar to native speaker speech. Therefore, it is desirable for a bar to have a large blue area. The conversion methods are shown before the underscore of each label, and the postfilter methods are shown after the underscore. "m2014" represents the method in [6] and "s10o5" represents the proposed NN in which the number of frames in a segment is 10 and the number of overlapped frames is 5. In Fig. 9, each method is arranged in descending order of similarity. Comparing the method described in [6] and the proposed NN with the GAN-based postfilter, the speech converted by the latter approach is more similar to native speech than that obtained with the former method.

Fig. 10 shows the result of evaluating similarity with respect to Indian speech and converted speech. The red areas in the bars indicate the rate of evaluations where speech converted by the method is dissimilar to Indian speech. Therefore, it is desirable for a bar to have a large red area. In Fig. 10, each method is arranged in ascending order of similarity. A comparison of the method in [6] and the proposed NN with a GAN-based postfilter shows that the speech converted by the latter approach provided a result that was less similar to Indian speech.

VII. Conclusions

In this research, we considered the problem of automatically modifying the pronunciation of non-native speakers. We introduced a segment feature as the input and output of an NN to convert a feature sequence segment-by-segment in real-time. Furthermore, to guarantee the continuity of features, we have proposed an NN architecture such that the converted features in overlapping sections of segments do not contradict

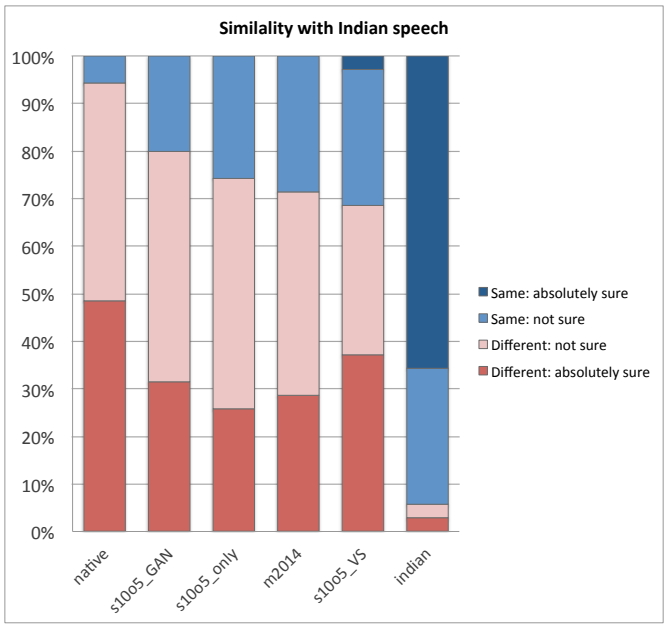

Fig. 10. **Subjective evaluation on similarity with Indian speech.** Each method is arranged in ascending order of similarity.

each other. We have verified the effectiveness of the proposed method by comparing the converted speech in subjective and objective evaluation experiments.

ACKNOWLEDGMENT

This research was conducted with the grant of JSPS Grant 26730100.

REFERENCES

[1] M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara, "Voice conversion through vector quantization," Proc. ICASSP, pp.655–658, 1998.
[2] Y. Stylianou, "Applying the harmonic plus noise model in concatenative speech synthesis," IEEE Trans. ASLP, vol.9, no.1, pp.21–29, 2001.
[3] T. Toda, A.W. Black, and K. Tokuda, "Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory," IEEE Trans. ASLP, vol.15, no.8, pp.2222–2235, 2007.
[4] T. Nakashika, T. Takiguchi, and Y. Ariki, "Voice conversion based on speaker-dependent restricted boltzmann machines," IEICE Trans. Info. Syst., vol.E67-D, no.6, pp.1403–1410, 2014.
[5] 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/ACM Trans. ASLP, vol.22, no.12, pp.1859–1872, 2014.
[6] S.H. Mohammadi and A. Kain, "Voice conversion using deep neural networks with speaker-independent pre-training," Proc. SLT, pp.19–23, 2014.
[7] R. Takashima, T. Takiguchi, and Y. Ariki, "Exemplar-based voice conversion using sparse representation in noisy environments," IEICE Trans. Info. Syst., vol.E96-A, no.10, pp.1946–1953, 2013.
[8] Z. Wu, T. Virtanen, E. Chng, and H. Li, "Exemplar-based sparse representation with residual compensation for voice conversion," IEEE/ACM Trans. ASLP, vol.22, no.10, pp.1506–1521, 2014.
[9] H. Silén, E. Helander, J. Nurminen, and M. Gabbouj, "Ways to implement global variance in statistical speech synthesis," Proc. Interspeech, pp.1436–1439, 2012.
[10] Z. Wu, T. Virtanen, E. Chng, and H. Li, "Post-filters to modify the modulation spectrum for statistical parametric speech synthesis," IEEE/ACM Trans. ASLP, vol.24, no.4, pp.757–767, 2016.
[11] I.J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," Adv. NIPS, 2014.
[12] T. Kaneko, H. Kameoka, N. Hojo, Y. Ijima, K. Hiramatsu, and K. Kashino, "Generative adversarial network-based postfilter for statistical parametric speech synthesis," Proc. ICASSP, 2017 (to appear).
[13] Y. Saito, S. Takamichi, and H. Saruwatari, "Evaluation of dnn-based voice conversion deceiving anti-spoofing verification," Technical Report of IEICE, 2017. in Japanese.
[14] L. Sun, S. Kang, K. Li, and H. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," Proc. ICASSP, pp.4869–4873, 2015.
[15] T. Toda, T. Muramatsu, and H. Banno, "Implementation of computationally efficient real-time voice conversion," Proc. Interspeech, pp.94–97, 2012.
[16] 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," Proc. ICASSP, pp.3933–3936, 2008.
[17] M. Morise, F. Yokomori, and K. Ozawa, "WORLD: A vocoder-based high-quality speech synthesis system for real-time applications," IEICE Trans. Inf. Syst., vol.E99-D, no.7, pp.1877–1884, 2016.
[18] D.W. Griffin and J.S. Lim, "Signal estimation from modified short-time Fourier transform," IEEE Trans. on Acoust., Speech and Sig. Proc., vol.32, no.2, pp.236–243, 1984.
[19] T. Nakamura and H. Kameoka, "Fast signal reconstruction from magnitude spectrogram of continuous wavelet transform based on spectrogram consistency," Proc. DAFx, pp.129–135, 2014.
[20] M. Mirza and S. Osindero, "Conditional generative adversarial nets," arXiv preprint arXiv:1411.1784, 2014.
[21] E. Denton, S. Chintala, A. Szlam, and R. Fergus, "Deep generative image models using a laplacian pyramid of adversarial networks," Adv. NIPS, pp.1486–1494, 2015.
[22] L. Sun, S. Kang, K. Li, and H. Meng, "Deep residual learning for image recognition," Proc. CVPR, pp.770–778, 2016.
[23] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," Proc. CVPR, pp.3431–3440, 2015.
[24] D.P. Kingma and M. Welling, "Adam: A method for stochastic optimization," Proc. ICLR, 2015.
[25] P. Vincent, H. Larochelle, I. Lajoie, Y. Bengio, and P.-A. Manzagol, "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion," Journal of Machine Learning Research, vol.11, no.Dec, pp.3371–3408, 2010.
[26] S. Rifai, P. Vincent, X. Muller, X. Glorot, and Y. Bengio, "Contractive auto-encoders: Explicit invariance during feature extraction," Proceedings of the 28th international conference on machine learning (ICML-11), pp.833–840, 2011.
[27] N. Srivastava, G.E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.," Journal of Mach. Learn. Res., vol.15, no.1, pp.1929–1958, 2014.
[28] T. Toda, L. Chen, D. Saito, F. Villavicencio, M. Wester, Z. Wu, and J. Yamagishi, "The voice conversion challenge 2016," Proc. Interspeech, 2016.