Non-Parallel Voice Conversion Using CycleGAN-VC3, Random CNN and Baidu API

ChenYiyun, HeLanxin, LinXin, ZhengSiyuan

***Abstract*—We reproduced a non-parallel voice conversion (VC) method CYCLEGAN-VC3 and compared it with VC based on TTS. In order to show the State-of-the-art technology, we calls Baidu's intelligent voice API to realize the speaker conversion as the second method. As results, Baidu API got the highest MOS rate among all, and CycleGAN-VC3 is regarded as understandable but performs bad at similarity. As the third method, CNN got the worst grade.**

***Keywords:* Voice conversion, Cycle-GAN, CNN, Baidu API.**

I. INTRODUCTION

A. Basic Concepts of Speech Conversion System

Speech signals convey a wide range of information. Among them, the meaning of the message being uttered is of prime importance. However, secondary information such as speaker identity also plays an important part in oral communication. Voice conversion techniques attempt to transform the speech signals uttered by a given speaker(source), so that it sounds as if it was uttered by another speaker (target).VC is useful in many applications, such as customizing audio book and cloning of voices of historical persons. Since VC technology involves identity conversion, it can also be used to protect the privacy of the individual in social media and sensitive interviews, for instance.

An overview of a typical VC system is presented in Figure 1 (Erro et al., 2010a). In the training phase, the VC system is presented with a set of utterances recorded from the source and target speakers (the training utterances). [1] The speech analysis and mapping feature computation steps encode the speech waveform signal into a representation that allows modification of speech properties. Source and target speakers' speech segments are aligned (with respect to time) such that segments with similar phonetic content are associated with each other. The mapping or conversion function is trained on these aligned mapping features. In the conversion phase, after computing the mapping features from a new source speaker utterance, the features are converted using the trained conversion function. [2] The speech features are computed from the converted features which are then used to synthesize the converted utterance waveform.

*Asignment of group members:

ZhengSiyuan.Author,Contact,information(2018302120069@whu.edu.cn), contribution to the project.

LinXin. Author, Contact information(2017301200184@whu.edu.cn), contribution to the project..

HeLanxin. Author, Contact information, (2016301200291@whu.edu.cn), contribution to the project.

ChenYiyun.Author, Contact information, (2018302120012@whu.edu.cn), contribution to the project.

There are various ways to categorize VC methods. One factor is based on the language that source and target speakers speak. A second factor is whether they are text-dependent or text-independent. Text-dependent approaches require word or phonetic transcriptions along with the recordings. These approaches may require parallel sentences recorded from both source and target speakers. A third factor is whether they require parallel or non-parallel recordings during their training phase. Parallel recordings are defined as utterances that have the same linguistic content, and only vary in the aspect that needs to be mapped (speaker identity, in the VC case). The following discussion is based on the third classification factor.

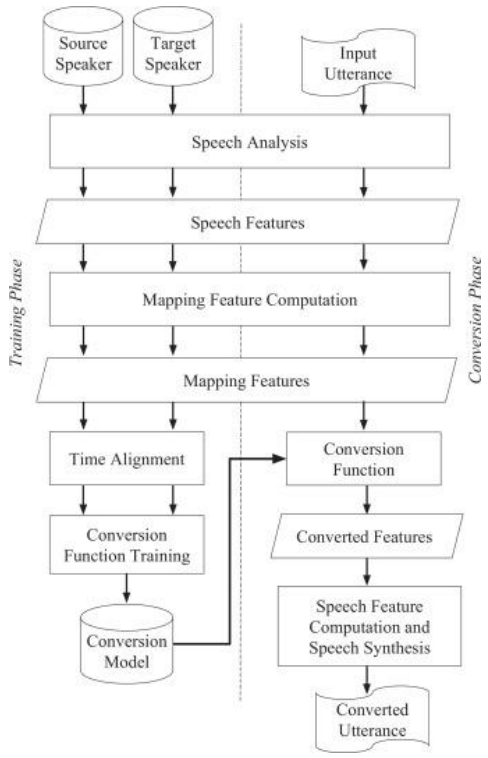

Figure1.Training and conversion phases of a typical VC system.

B. State of Research

1) Parallel VC

Many VC methods (including the above-mentioned) are categorized as parallel VC, among them, there are three typical types: statistical methods, such as a Gaussian mixture model (GMM). Neural network (NN)-based methods, such as a restricted Boltzmann machine (RBM) , feed forward NN, recurrent NN (RNN), and convolutional NN (CNN), and

generative adversarial networks (GANs). exemplar-based methods, such as non-negative matrix factorization (NMF).

2) Non-parallel VC

Many VC methods including those mentioned above typically use temporally aligned parallel data of source and target speech as training data. If perfectly aligned parallel data are available, obtaining the mapping function becomes relatively simple; however, collecting such data can be a painstaking process in real application scenarios. Even though we could collect such data, we need to perform automatic time alignment, which may occasionally fail. This can be problematic since misalignment involved in parallel data can cause speech-quality degradation; thus, careful pre-screening and manual correction may be require. These facts motivated us to consider a VC problem that is free from parallel data. [3]

However, non-parallel VC is quite challenging and is inferior to parallel VC in terms of quality due to the disadvantages of the training conditions. To alleviate these severe conditions, several studies have incorporated an extra module (e.g., an automatic speech recognition (ASR) module or extra data (e.g., parallel utterance pairs among reference speakers. Although these additional modules or data are helpful for training, preparing them imposes other costs and thus limits application. To avoid such additional costs, recent studies have examined the use of probabilistic NNs (e.g., an RBN and variational autoencoders (VAEs), which embed the acoustic features into common low-dimensional space with the supervision of speaker identification. It is noteworthy that they are free from extra data, modules, and time alignment procedures. [5] However, one limitation is that they need to approximate data distribution explicitly (e.g., Gaussian is typically used), which tends to cause over-smoothing through statistical averaging.

To avoid such a requirement and achieve non-parallel VC using only acoustic data, variational auto encoder-based methods and GAN-based methods have been proposed. Among them, CycleGAN-VC3 has garnered attention and has been widely used as benchmark method in several studies. CycleGAN-VC3 uses PatchGAN as its discriminator and its training objective is two-step adversarial losses. As for generator, CycleGAN-VC3 introduces a 2-1-2D CNN that uses 2D CNNs in upsampling and downsampling blocks and uses 1D CNNs in residual blocks. It also incorporates time-frequency adaptive normalization (TFAN). [6]

C. Comparative Experiments

Many domestic companies provide the API (application programming interface) of speech technology, because its speech recognition and synthesis model has been well optimized in the long-term practice. Our group calls Baidu's intelligent voice API to realize the speaker conversion as a comparison. Through the API, we can accomplish the task of non-restricted voice conversion based on TTS. The results are used as comparison to show the State-of-the-art technology.

In this paper, we compared VC based on TTS and VC based on cycle-consistent adversarial networks (CycleGan-VC3). And it is structured as follows. The Section 2 describes the fundamental theoretical aspects of voice conversion. In Section 3, the deep-learning method CycleGan and its improvements are discussed. The Section 4, the experimental evaluations are presented. Finally, the summarization of this paper will be given in Section 5.

II. THE BASIC THEORY OF VOICE CONVERSION

A. Feature Extraction

In voice conversion, a speech signal is first analyzed into a time-synchronized acoustic features, namely, MCEP as the spectral feature.The main idea of MFCC is to transform the signal from time domain to frequency domain and to map the transformed signal from H ertz to Mel-scale due to the fact that 1 kHz is a threshold of human's hearing ability. Human ears are less sensitive to sound with frequency above that threshold. The calculation of MFCCs includes the following steps:

- Pre-emphasis filtering
- Take the absolute value of the short time Fourier transformation using windowing
- Warp to auditory frequency scale (Mel-scale)
- Take the discrete cosine transformation of the log-auditory-spectrum
- Return the first q MFCCs

B. MelGAN Vocoder

Current approaches to mel-spectogram inversion can be categorized into three distinct families: pure signal processing techniques, autoregressive and non-autoregressive neural networks. The main issue with pure signal processing methods is that the mapping from intermediate features to audio usually introduces noticeable artifacts. And auto-regressive models are usually not suited for real-time applications. In addition, the large size of the model makes non autoregressive models impractical for applications with a constrained memory budget. So we use MelGAN, a non-autoregressive feed-forward convolutional architecture to perform audio waveform generation in a GAN setup. Figure 2 shows the overall architecture. [7]

1) Generator

The generator is a fully convolutional feed-forward network with mel-spectrograms as input and raw waveform x as output. Since the mel-spectrogram (used for all experiments) is at a 256 X lower temporal resolution, we use a stack of transposed convolutional layers to up sample the input sequence. Each transposed convolutional layer is followed by a stack of residual blocks with dilated convolutions.

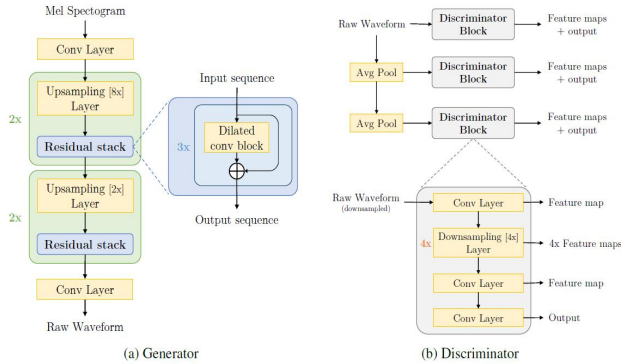

Figure 2.MelGAN model architecture.

2) Discriminator

The vocoder adopt a multi-scale architecture with 3 discriminators (D1;D2;D3) that have identical network structure but operate on different audio scales. D1 operates on the scale of raw audio, whereas D2; D3 operate on raw audio downsampled by a factor of 2 and 4 respectively. The downsampling is performed using strided average pooling with kernel size 4. Multiple discriminators at different scales are motivated from the fact that audio has structure at different levels. This structure has an inductive bias that each discriminator learns features for different frequency range of the audio. [8] For example, the discriminator operating on downsampled audio, does not have access to high frequency component, hence, it is biased to learn discriminative features based on low frequency components only.

C. The Basic Framework of Voice Conversion System

Generally speaking, the speech conversion system consists of two stages, training and conversion. In the training phase, first select a segment of the target speaker for training, which can be the same sentence or not. Next, use the features as input data to train the speech conversion model.

In the conversion phase, a segment of speech from an arbitrary input source speaker is also analyzed by speech analysis and feature extraction, and then be converted by the model we got. The converted speech is still in the form of feature parameters, which needs to be generated by speech synthesis.

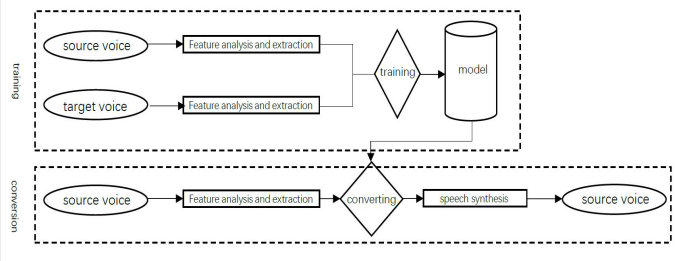

Figure 3.The basic framework of the voice conversion system

III. CycleGAN-VC3

A. Introduction

As an alternative, non-parallel VCs that do not require parallel corpus for training have recently received attention. In terms of data collection costs, non-parallel VC is attractive. However, due to the lack of clear supervision, its learning is challenging. To solve this problem, some studies have used language information. Although this additional supervision can improve performance, it requires auxiliary data or modules to extract language information.

In order to avoid this requirement and use only acoustic data to implement non-parallel VC, methods based on variational autoencoders and methods based on GAN have been proposed. Among them, CycleGAN-VC and its variants (CycleGAN-VC2 and StarGAN-VCs) have attracted people's attention and have been widely used as benchmark methods in some studies. However, due to their ambiguity about the effectiveness of mel-spectrogram conversion, they are usually used for half-spectrum peak conversion even when the comparison method uses mel-spectrogram as the conversion target.

These facts prompted us to study the applicability of CycleGAN-VC and CycleGAN-VC2 in mass spectrum conversion. Through initial experiments, we found that when CycleGAN-VC/VC2 is directly applied to the mass spectrum, the time-frequency structure that should be preserved during the conversion will be damaged.

In order to solve this problem, CycleGAN-VC3 was proposed, which is an improvement of CycleGAN-VC2, which combines time-frequency adaptive normalization (TFAN). TFAN is inspired by spatial adaptive (de)normalization (SPADE), which was originally proposed for semantic image synthesis. We modified SPADE to apply it to 1D and 2D time-frequency features. Using TFAN, we can adjust the scale and deviation of the conversion feature, while reflecting the time-frequency structure of the source Mel spectrogram.

B. Conventional CycleGAN-VC/VC2

1) Training objectives

CycleGAN-VC / VC2 aims to learn the mapping $G_{X\to Y}$, which converts the source acoustic feature $x \in X$ into the target acoustic feature $y \in Y$ without using a parallel corpus. Inspired by CycleGAN, which was originally proposed for unpaired image-to-image conversion, CycleGAN-VC/VC2 uses adversarial loss, cycle consistency loss, and identity mapping loss to learn the mapping. In addition, CycleGAN-VC2 uses a second adversarial loss to improve the details of reconstructed features.

Adversarial loss：In order to ensure that the transformed feature $G_{X\to Y}(x)$ is in target Y, use the adversarial loss $L_{adv}^{X\to Y}$ as follows:

$$L_{adv}^{X\to Y} = E_{y\sim P_Y}\left[\log D_Y(y)\right] + E_{x\sim P_X}\left[\log(1-D_Y(D_{X-Y}(x)))\right] \quad (1)$$

Among them, the discriminator D_Y tries to distinguish the composite $G_{X\to Y}(x)$ from the real number y by maximizing the loss, and the $G_{X\to Y}$ tries to synthesize D_Y that can deceive $G_{X\to Y}(x)$ by minimizing the loss. Similarly, inverse mapping $G_{Y\to X}$ and discriminator D_X use $L_{adv}^{Y\to X}$ for adversarial training.

Cycle-consistency loss： In order to maintain the composition in the conversion, the cycle consistency loss L_{cyc} is used as follows:

$$L_{cyc} = E_{x\sim P_X}\left[\left\|G_{Y\to X}(G_{X\to Y}(x))-x\right\|_1\right], + E_{y\sim P_Y}\left[\left\|G_{X\to Y}(G_{Y\to X}(y))-y\right\|_1\right] \quad (2)$$

This loss is used with the hyper-parameter λ_{cyc} that controls its relative importance. The loss helps $G_{X\to Y}$ and $G_{Y\to X}$ to identify false pairs within the cycle consistency constraint.

Identity-mapping loss： In order to save the input, an identification mapping loss L_{id} is used, as shown below:

$$L_{id} = E_{y\sim P_Y}\left[\left\|G_{X\to Y}(y)-y\right\|_1\right], + E_{x\sim P_X}\left[\left\|G_{Y\to X}-x\right\|_1\right] \quad (3)$$

This loss is used with the hyperparameter λ_{id} that controls its relative importance.

Second adversarial loss： In CycleGAN-VC2, in order to reduce the statistical average caused by L1 loss (formula 2), an additional discriminator D'_X is introduced, and a second counter-loss $L_{adv2}^{X\to Y\to X}$ is applied to the features after cyclic conversion, as follows Shown:

$$L_{adv2}^{X\to Y} = E_{x\sim P_X}\left[\log D'_X(x)\right] + E_{x\sim P_X}\left[\log(1-D'_X(D_{Y\to X}(G_{X\to Y}(x))))\right], \quad (4)$$

Similarly, introduce the discriminator D'_Y and apply $L_{adv2}^{Y\to X\to Y}$ to the reverse mapping.

2) Generator architectures

CycleGAN-VC uses a one-dimensional CNN generator to capture the overall relationship and feature direction while preserving the temporal structure. In particular, the network is composed of down-sampling, residual and up-sampling blocks to effectively capture a wide range of time relationships, and a gated linear unit (GLU) is used as an activation to adaptively learn the order and hierarchy.

However, research on CycleGAN-VC2 shows that the one-dimensional CNN in the down-sampling and up-sampling blocks affects the structure that should be kept in the conversion. To alleviate this situation, CycleGAN-VC2 introduces 2-1-2D CNN, which uses 2D CNN in the up-sampling and down-sampling blocks, and 1D CNN in the remaining blocks. The former is used to extract the time-frequency structure while retaining the original structure. The latter is used to perform dynamic changes.

3) Discriminator architectures

CycleGAN-VC uses a 2D CNN discriminator to discriminate data based on 2D spectral texture. In particular, it uses FullGAN with a fully connected layer as the last layer to distinguish data based on the overall input structure. However, in FullGAN, many parameters need to be learned, which leads to learning difficulties. To alleviate this situation, CycleGAN-VC2 introduces PatchGAN that uses convolution in the last layer. This reduces the parameters and stabilizes the GAN training.

C. TFAN: Time-frequency adaptive normalization

CycleGAN-VC and CycleGAN-VC2 were originally designed for Mel-Cepstrum conversion, and their effectiveness in Mel-Cepstrum conversion has not been fully tested. We checked their effectiveness through experience and found that they damage the time-frequency structure that should be retained in the conversion, as shown in Figure 1.

Based on this discovery, we designed TFAN to extend the instance normalization (IN) to adjust the ratio and deviation of the conversion features, while reflecting the source information (ie x) in a time and frequency manner. In particular, we designed TFAN for 1D and 2D time-frequency features for use in 2-1-2D CNN (Section 2.2). Figure 2 illustrates the architecture of TFAN. Given a feature f, TFAN normalizes it in a channel manner similar to IN, and then uses scale $\gamma(x)$ and deviation $\beta(x)$ to modulate the normalized feature element-wise, which is to use CNN from x Calculated:

$$f' = \gamma(x)\frac{f-\mu(f)}{\sigma(f)} + \beta(x), \quad (5)$$

Where f' is the output characteristic, $\mu(f)$ and $\sigma(f)$ are the channel average and standard deviation of f, respectively.

In IN, the ratio β and deviation γ that are independent of x are applied in a channel manner, while in TFAN, the ratio and deviation calculated based on x (ie $\beta(x)$ and $\gamma(x)$) are applied in an elemental manner. These differences allow TFAN to adjust the scale and deviation of f, while reflecting x in time and frequency. [9]

IV. EXPERIMENTS

We conducted experiments to evaluate our fimethod on a parallel-data-free VC task. We used our own dataset, which was recorded by two native Chinese speakers. For each speaker, 32 utterances (approximately 5 min, which is relatively low for VC) and 35 utterances were used for training and evaluation, respectively. The recordings were downsampled to 16 kHz. Following the study of Mel-GAN, which we used as a vocoder in our experiments, we extracted an 24-dimensional log mel-spectrogram with a window length of 1024 and hop length of 256 samples.

A. CycleGAN-VC3

We used CycleGAN-VCs for mel-spectrogram conversion and synthesized waveforms using the pretrained MelGAN vocoder. We did not alter the parameters of the vocoder such that we could focus on the evaluation of melspectrogram conversion; however, fine-tuning them for each speaker is a possible means for improvement.

As the acoustic feature is changed from mel-cepstrum to mel-spectrogram, the feature dimension increased from 35 to 80. However, the generators of CycleGAN-VCs are fully convolutional; therefore, they can be used without modifying the network architecture. Regarding the discriminators, we used the same network architecture as those for mel-cepstrum conversion, except that in CycleGANVC2/ VC3, the kernel size in the second-last convolutional layer was doubled in the frequency direction.The training settings were similar to those used in CycleGAN-VC/VC2 for mel-cepstrum conversion. For preprocessing, we normalized the mel-spectrograms using the mean and variance of the training data. We used the least square GAN as the GAN objective. We trained the networks for 500k iterations using the Adam optimizer with a batch size of 1. A training sample consisted of randomly cropped 64 frames (approximately 0.75 s). The learning rates were set to 0.0002 for the generators and 0.0001 for the discriminators with momentum terms β_1 and β_2 of 0.5 and 0.999, respectively. λ_{cyc} and λ_{id} were set to 10 and 5, respectively, and $\mathcal{L}_{id}$ was used only for the first $10k$ iterations. Note that similar to the original CycleGAN-VC/VC2, we did not use extra data, modules, or time alignment procedures for training.

B. Baidu Api

The main program can be divided into three parts. First of all, after establishing a TCP connection and getting permission, send a piece of audio to the address provided by Baidu by POST, and the server will return the recognition result which is encapsulated in JSON format. In the second part, the recognized text information with some parameters (to select different timbre, speech speed and tone) is sent to the API of speech synthesis in the same way. After that, the return synthetic speech is obtained and saved. Finally, we encapsulate this function with pyqt5 to complete a UI for easier operation.

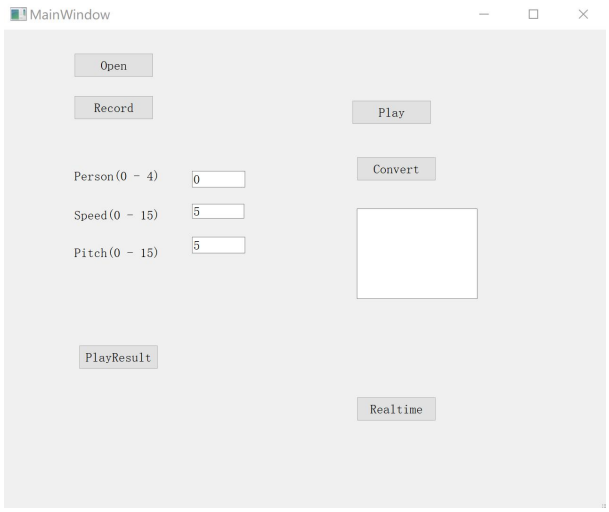

Figure 4 Baidu API UI

C. Random CNN

The raw audio is converted to a spectrogram via Short Time Fourier Transform. Spectrogram is a 2D representation of a 1D signal so it can be treated (almost) as an image. In fact, is better to think of spectrogram as of 1xT image with F channels.

This method uses the image processing method of computer vision for reference. The spectrum of audio file is divided into two parts: content and style. Input two audio files and output one audio file. We use convolutional neural network to realize this conversion process. The ultimate goal of network training is to make the style of the output audio file the same as one audio file, and the content the same as another input audio file. Finally, we can get the result of speaker style conversion through the trained network.

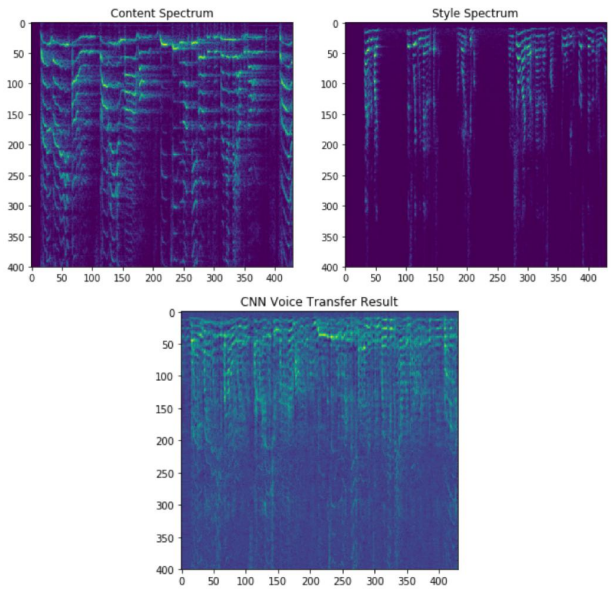

Figure 5 CNN Results

D. Results

We conducted listening tests to evaluate the performance of converted speech. We evaluated the naturalness and speaker similarity of the converted samples. We compared our method

with the baseline of the VCC 2016, which is a GMM-based method using parallel. We also made a comparison between 2 methods that we have already reproduced with Baidu API，that is CycleGANs and Random CNN. To measure naturalness, we conducted a mean opinion score (MOS) test. As a reference, we used original and synthesized-and-analyzed (upper bound of our method) speeches of target speakers. To measure speaker similarity, we used the same/different paradigm. There were nine participants who were well-educated Chinese speakers. We evaluated on two subsets: intra-gender VC and inter-gender VC.

We show the MOS for naturalness in Fig.8. The results indicate that most of the methods we used significantly outperformed the baseline. We show the similarity to a source speaker and to a target speaker in Fig. 7. The results indicate that our method was slightly inferior to the baseline in inter-gender VC but superior in intra-gender 2 VC. Overall, our method is comparable to the baseline. This is noteworthy since our method is trained under disadvantageous conditions with half the amount of and non-parallel data.

In order to reasonably evaluate the quality of the work done by our group, we chose the results of other people's work to compare with our own. In general, baidu API is very good, as a voice conversion model, it is better than most neural network training methods. The effect of cycle Gan is inferior to that of the source code author. On the one hand, our own data set is relatively small, and the training time and computing power are limited. On the other hand, we judge that there may be some differences between Chinese and English voice timbres, and our own data results are Chinese. The example given by the author of random CNN method is quite good, which can clearly hear the speaker's timbre being converted, but the result of our operation has no obvious conversion effect. Because this method has no training data set, we speculate that random CNN method is not universal, but for some specific situations. Compared with the results of other authors, there is still a certain gap in the MOS and ABX indicators of the network model we trained. To get better results, we need to further optimize the model and algorithm.

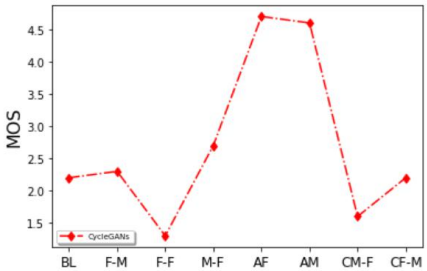

Figure 6 MOS

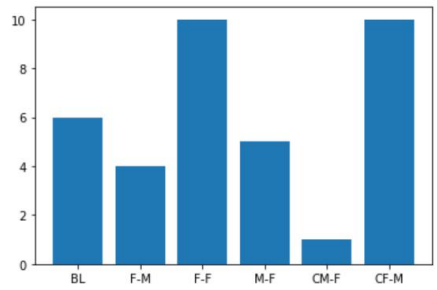

Figure 7 ABX results

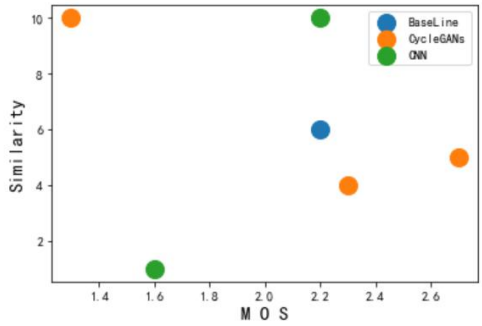

Figure 8 Consolidated results.

REFERENCES

[1] Mohammadi S H, Kain A. An Overview of Voice Conversion Systems[J]. Speech Communication, 2017, 88:65-82..

[2] W.- Stylianou Y, Cappé O, Moulines E. Continuous probabilistic transform for voice conversion[J]. IEEE Transactions on Speech & Audio Processing, 1998, 6(2):131-142.

[3] Chen L H, Ling Z H, Liu L J, et al. Voice conversion using deep neural networkswith layer-wise generative training[J]. IEEE/ACM Transactions on Audio Speech &Language Processing, 2014, 22(12):1859-1872.

[4] Takashima R, Takiguchi T, Ariki Y. Exemplar-Based Voice Conversion Using Sparse Representation in Noisy Environments[J]. Ieice Transactions on Fundamentals of Electronics Communications & Computer Sciences, 2013, E96.A(10):1946-1953.

[5] Takashima R, Takiguchi T, Ariki Y. Exemplar-Based Voice Conversion Using Sparse Representation in Noisy Environments[J]. Ieice Transactions on Fundamentals of Electronics Communications & Computer Sciences, 2013, E96.A(10):1946-1953.

[6] T. Kaneko and H. Kameoka. Parallel-Data-Free Voice Conversion Using Cycle-Consistent Adversarial Networks. arXiv:1711.11293, Nov. 2017 (EUSIPCO, 2018).Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces(Translation Journals style),” *IEEE Transl. J. Magn.Jpn.*, vol. 2, Aug. 1987, pp. 740–741 [*Dig. 9th Annu. Conf. Magnetics* Japan, 1982, p. 301].

[7] E Xu N, Tang Y, Bao J, et al. Voice conversion based on Gaussian processes by coherent and asymmetric training with limited training data [J]. Speech Communication, 2014,

[8] E Xu N, Tang Y, Bao J, et al. Voice conversion based on Gaussian processes by coherent and asymmetric training with limited training data [J]. Speech Communication, 2014,

[9] Tomoki Toda, Ling-Hui Chen, Daisuke Saito, Fernando Villavicencio, Mirjam Wester, Zhizheng Wu, and Junichi Yamagishi. The Voice Conversion Challenge 2016. The Annual Conference of the International Speech Communication Association (INTERSPEECH), 2016.J. U. Duncombe, “Infrared navigation—Part I: An assessment of feasibility (Periodical style),” *IEEE Trans. Electron Devices*, vol. ED-11, pp. 34–39, Jan. 1959.