K-CONVERTER: AN UNSUPERVISED SINGING VOICE CONVERSION SYSTEM

Ying Zhang, Peng Yang, Jinba Xiao, Ye Bai, Hao Che, Xiaorui Wang

Kwai, Beijing, P.R. China

ABSTRACT

Singing voice conversion (SVC) converts a singer's voice to another one's voice while preserving the linguistic content. Recently, some SVC systems rely on supervised phonetic features extracted from pre-trained automatic speech recognition (ASR) models, increasing system complexity. Some end-to-end SVC systems use adversarial training, which causes instability during optimization. To address these issues, we present K-Converter, a simple system to disentangle the timbre, pitch, and content information without any manual supervision or adversarial training. First, low quefrencies of mel-frequency cepstral coefficients (MFCC), which remove the glottal excitation mainly, are used as input representations. And the pitch-shift augmentation is used for further disentangling the pitch. Second, an encoder network is carefully designed to construct an information bottleneck, which learns to break up the pitch and timbre information of the source. Third, the content consistency loss is introduced to keep the content consistent between encoder outputs of source utterances and reconstructed ones. Experimental results show that our proposed system performs well in both speech naturalness and timbre similarity, with better robustness to comparisons.

Index Terms— singing voice conversion, MFCC, temporal down-sampling, content consistency loss

1. INTRODUCTION

Singing voice conversion (SVC) aims to replace the voice of a given template song using another one's voice. Lyrics, tempo, and pitch are all critical for the musical expression and should be retained after conversion. SVC can aid the ability to free oneself from some of limitations of one's voice, enable a single singer to record a creative song by mimicking another voice.

Classical SVC methods are based on concatenative [1] or statistical methods [2, 3]. These methods need parallel data, i.e., different singers should perform the same song. As we known, it is hard and expensive to collect such parallel data.

Recently, several deep learning methods have achieved SVC with non-parallel data. They can be divided into two categories: the one is based on supervised phonetic features. Speaker-agnostic linguistic information are extracted by pre-trained ASR models in [4, 5, 6]. The input features of SVC systems have already disentangled the timbre and pitch information. However, the word accuracy of converted songs highly depends on the performance of the front-end ASR system. And a well-trained ASR system also requires a great number of labeled data. The other is based on unsupervised methods which don't employ any linguistic features. The adversarial training is first introduced for disentangling the timbre information in [7]. Subsequently, this method is advanced by applying an additional adversarial network to separate the pitch information in [8]. Although they have made progress in unsupervised SVC, it is widely acknowledged that the generative adversarial network is hard to train, and its convergence property is fragile. Other researchers adopt the variational autoencoder (VAE) to disentangle linguistic features from speech [9, 10]. Whereas outputs of VAE models are prone to over-smoothing, which leads to speech-quality degradation [11].

This paper proposes a simple and effective SVC system called K-Converter without any manual supervision or adversarial training. The K-Converter mainly contains a content encoder and a decoder. The content encoder is used to disentangle speaker-agnostic and pitch-invariant information by the following improvements. First, to relieve the decoupling stress of the content encoder, only the first 20-dimension MFCCs of 80-dimension MFCCs are used. As the source-filter theory describes, speech sounds are the responses of a sound source and a vocal tract filter [12]. And formant frequencies formulated by the vocal tract filter, which affect the phonation, can be separated by the cepstral analysis with low coefficients [13]. Second, the content encoder is designed with a small dimension setting and temporal downsampling. The small dimension setting is believed to constrain the information flow [6, 14]. The temporal downsampling is helpful to break up the pitch information without harming the linguistic encoding since the pitch is highly frame-dependent while each phone lasts for several frames. Third, a content consistency loss is introduced to keep content consistent between the source and the reconstructed utterance. Similar ideas are conducted in [5, 14]. The decoder is applied to predict speaker-dependent mel-scale spectrograms condition on the encoder outputs, target speaker embeddings, and the pitch extracted from the source utterance. Experimental results evidence our K-Converter can significantly improve the quality of the converted songs and simplify the model training.

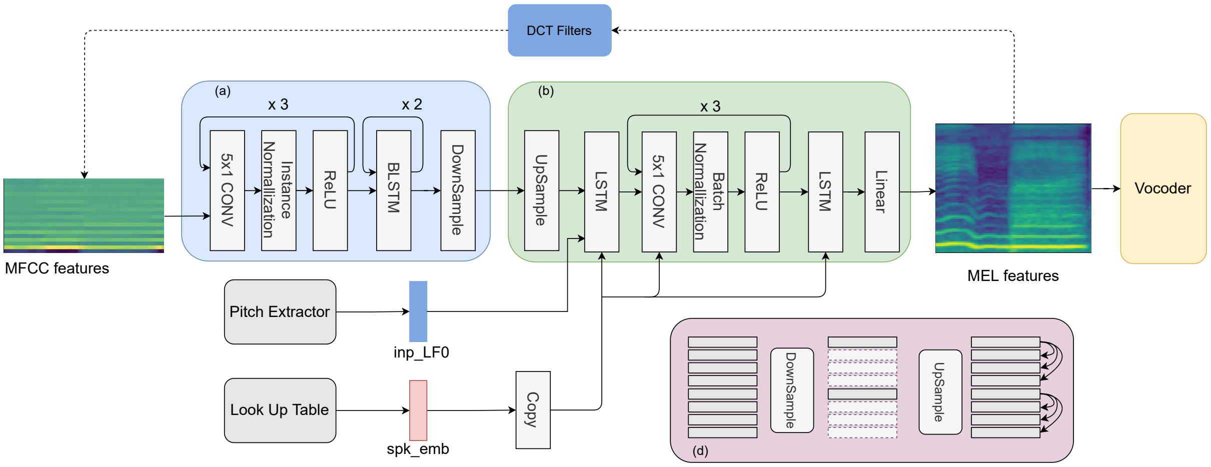

Fig. 1. The schematic architecture of our proposed K-Converter. CONV means convolution. BLSTM means bi-directional LSTM. The inp_LF0 is the pitch extracted from source speech. The spk_emb is learned during training and stored in a Look Up Table. The architecture contains (a) The content encoder. (b) The decoder. (c) Vocoder. We use the Parallel WaveGan to reconstruct waveform from mel-scale spectrograms. (d) describes the downsampling and upsampling stage along the temporal axis. Here we use an up/downsampling rate of 4 as an example. The real up/down-sampling rate is 8. Removed features are represented as dashed lines and will be up-sampled with nearest-neighbor interpolation. DCT filters represent discrete cosine transform filters.

2. METHOD

The architecture of our proposed K-Converter system is depicted in Fig. 1. It mainly contains a content encoder and decoder module. An additional pitch extractor and Look Up Table (LUT) extract fundamental frequency and speaker embedding vectors, respectively. The decoder module uses the input pieces of information to reconstruct the acoustic features. And the final vocoder reconstructs waveforms using the predicted acoustic features. There are two losses to be optimized in each training step. The first is the self-reconstruction loss, which is applied to samples of each singer separately. The second is the content consistency loss, which measures the difference between the encoder output of the source utterance and the reconstructed one's.

2.1. The Content Encoder

The content encoder is enforced to extract linguistic content information from input utterances with several deliberately designed strategies.

First, speech disentanglement is considered while selecting input representations. 20-dimension MFCCs are chosen as the input representations to prevent the content encoder from learning the pitch information. According to the source-filter and cepstral analysis theory, the low coefficients of the cepstrum mainly represent the vocal tract, which formulates the formant frequencies and affects the linguistic information. The high coefficients mainly represent the glottal excitation, which affects the fundamental frequency and the harmonics [12, 13]. Therefore, there is less pitch information in the 20-dimension MFCCs compared to the 80-dimension mel-scale spectrograms. Besides, pitch-shift augmentation is conducted to the training data. In detail, MFCCs fed to the content encoder are extracted from augmented utterances whose pitch is shifted up or down randomly by [-12, +12] semitones. However, the target mel-scale spectrograms and the condition pitch are extracted from original utterances.

Second, the content encoder is designed with a small dimension setting and carefully tuned temporal down-sampling to construct the information bottleneck. As we know, the length of the acoustic feature is much longer than its corresponding phonetic sequence. However, the pitch contour is highly time-dependent. Consequently, the temporally down-sampled encoder output is expected to represent the linguistic information while limiting the pitch information flowing through the encoder.

Third, the content consistency loss is introduced to keep content encoder outputs encoded from source and reconstructed utterances. Details of the content consistency loss will be described in section 2.3.

The content encoder comprises three 5x1 convolutional layers, each followed by instance normalization and ReLU activation and two bidirectional long short-term memory (BLSTM) layers. Instance normalization has been demonstrated useful to remove the global information such as speaker information [15]. The cell dimensions of the last BLSTM are 32. The forward and backward outputs of the BLSTM are downsampled by 8. In detail, for the forward outputs, the time steps $\{0, 8, 16, \cdots\}$ are kept, and for the backward outputs, the time steps $\{7, 15, 23, \cdots\}$ are kept. The small dimension setting and the temporal down-sampling operation construct an information bottleneck, limiting the information passing through the encoder.

Formally, let E be the encoder module, $\mathbf{X}$ be the mel-scale spectrograms. Let DCT be the process which transforms mel-scale spectrograms to expected MFCCs in this paper. The output of the encoder module $\mathbf{e}$ can be expressed as

$$\mathbf{e} = E(DCT(\mathbf{X})) \tag{1}$$

Note that the temporal down-sampling operation is omitted for briefness.

2.2. Decoder

The decoder predicts 80-dimension mel-scale spectrograms using the encoder output, speaker embedding, and pitch information. The encoder output is up-sampled by nearest-neighbor interpolation to restore to the original temporal resolution. The speaker embedding is retrieved from the LUT with a one-hot speaker id and then up-sampled to the actual temporal resolution by repeating. The speaker embedding is fed to multiple layers of the decoder to strengthen its influence. The log-domain fundamental frequency after linear interpolation is provided to the decoder.

The decoder contains an LSTM layer, three 5x1 convolutional layers, each followed by batch normalization and ReLU, and an LSTM layer. Outputs of the final LSTM layer are projected to dimension 80 with a linear layer.

Outputs $\hat{\mathbf{X}}$ of the decoder module D are calculated as follows:

$$\hat{\mathbf{X}} = D(\mathbf{e}, \mathbf{s}, \mathbf{p}) \tag{2}$$

Where $\mathbf{s}$ denotes the speaker embedding vector, $\mathbf{p}$ denotes the log-domain fundamental frequency after linear interpolation. Since parallel data is unavailable, $\mathbf{e}$, $\mathbf{s}$, $\mathbf{p}$ all correspond to the same utterance in the training stage. And the temporal up-sampling operation is also omitted.

In the inference stage, the source pitch is scaled to the vocal range of the target speaker before it is sent to the decoder module, as follows:

$$p = p_{src} \times \frac{M_{tp}}{M_{sp}} \tag{3}$$

Where p_{src} is the pitch of source utterance, M_{tp} and M_{sp} are the mean value of all target speaker's utterances and the input source utterances separately.

2.3. Loss Function

There are two losses to be minimized in our method.

(1) Self-reconstruction loss L_{recon}. Model parameters are optimized to minimize the L1 loss between input mel-scale spectrograms $\mathbf{X}$ and reconstructed ones $\hat{\mathbf{X}}$,

$$L_{recon} = \mathbb{E}[\|\mathbf{X} - \hat{\mathbf{X}}\|_1] \tag{4}$$

(2) Content consistency loss $L_{consist}$. The content information extracted from reconstructed mel-scale spectrograms $\hat{\mathbf{X}}$, should be the same to the input mel-scale spectrograms $\mathbf{X}$ [14].

$$L_{consist} = \mathbb{E}[\|E(DCT(\mathbf{X})) - E(DCT(\hat{\mathbf{X}}))\|_1] \tag{5}$$

At each training step, the reconstructed mel-scale spectrograms are transformed to MFCCs and sent to the content encoder again to optimize the content consistency loss $L_{consist}$. Note that the DCT is just a linear transformation in implementation, so it is differential.

The total loss L_{total} can be calculated as:

$$L_{total} = L_{recon} + \lambda L_{consist} \tag{6}$$

where λ is a weight factor.

3. EXPERIMENTS

3.1. Experimental Setup

Our experiments are implemented with an internal Mandarin singing corpus, and the NUS-48E dataset [16]. The Mandarin dataset contains 6 female singers and 6 male singers. And each singer has 0.5 hours of audio data on average. And all of the NUS-48E dataset is used for training. For evaluation, unseen utterances from 2 female and 2 male singers are chosen for test.[1] All songs are sampled to 24 kHz.

F0 features are extracted by the WORLD vocoder [17]. Before sending to the decoder, the linear interpolation is conducted. Mel-scale spectrograms are extracted with a 50 milliseconds window length, a 12.5 milliseconds hop length, a 2048 STFT window size, and 80 bins. We use 80 DCT-filters [18] to transform the mel- spectrograms into MFCCs, and only the first 20-dimension MFCCs (MFCC20) are used.

The vocoder used in this work is a pre-trained multi-speaker Parallel WaveGAN model [19] which only takes 80-dimension mel-scale spectrograms as input. The multi-length generative adversarial network proposed by [20] is also adopted to improve the quality of generated audio samples.

We compare our model with two baseline systems. For a fair comparison, the basic neural network architectures are the same, and only the key disentanglement methods are compared: (1) BASE1, our implementation follows [15], the Kullback-Leibler divergence is used for content disentanglement, and the 8x temporal down-sampling is also conducted. (2) BASE2, adversarial training modules are used to enforce encoder outputs to be speaker-agnostic and pitch-invariant [7, 8]. Architectures of the singer classification network and pitch regression network are the same as those in [8]. The singer classification loss and pitch regression loss are added adversarially to the reconstruction loss to train the entire model. Since the confusion net is conducted on the frame level in [7, 8], no temporal down-sampling is made in the BASE2 model.

[1] samples can be found in https://vcdemo-1.github.io/KConverter

Table 1. Evaluation of baselines and our proposed system. Reconst. means reconstruction.

System		Naturalness	Similarity	NCC
Ground Truth		4.50	—	—
Reconst.	BASE1	4.18	4.35	0.888
	BASE2	4.27	4.28	0.858
	Ours	4.35	4.40	0.895
Conversion	BASE1	3.41	3.23	0.856
	BASE2	3.57	3.18	0.839
	Ours	**3.88**	**3.56**	**0.882**

Table 2. Ablation of input feature representations. MEL80 means the 80-dimension mel-scale spectrograms, +Aug means MFCC20 with pitch-shift augmentation.

Feature Representation	Naturalness	Similarity	NCC
MEL80	3.70	2.68	0.850
MFCC20	3.65	3.38	0.866
+Aug	**3.88**	**3.56**	**0.882**

All conversion models are trained for 250k steps with a batch size of 32. We use the Adam optimizer [21]. For the weight factor in Equation 4, we set $\lambda = 0$ in the early 20k steps and then $\lambda = 1$ for better model convergence. As for the training of the BASE2, back-translation and mixup are employed as [7, 8] after the first 50k steps. Weight factors of the singer classification loss and the pitch regression loss are set to 0.01 and 0.05, respectively.

3.2. Comparison with baseline methods

Subjective hearing tests are conducted by 10 music experts. Two metrics are conducted to measure the models: (1) Mean Opinion Score (MOS) of the naturalness, used to judge an integrated assessment of intonation, rhythm, melody, clarity, and expression. (2) MOS of the timbre similarity between the converted samples and the target timbres. Both metrics are scaled between 1-5.

The normalized cross-correlation (NCC) scores are also calculated to compare source and converted songs' pitch. The score ranges from 0 to 1. The higher the score is, the better the output pitch matches the input pitch.

Table 1 shows the evaluation results between baseline systems (BASE1 and BASE2) and our proposed system (Ours). Scores of reconstruction are higher than conversion since three models all benefit from the reconstruction loss. The BASE2 system gets the lowest similarity and NCC results, exposing its maintaining pitch and timbre shortage. Note that there is a significant degradation for the baseline systems when source speech with singing skills switched, such as converting true false sound or changing the singing key signature. Still, our system does not display such a degradation owing to its better performance in pitch disentanglement. These also demonstrate the robustness of our system.

Table 3. Ablation of the temporal down-sampling rate **N**.

N	Naturalness	Similarity	NCC
1	2.89	3.47	0.854
8	**3.88**	**3.56**	**0.882**
32	2.34	2.98	0.856

Table 4. Ablation of the content consistency loss.

	Naturalness	Similarity	NCC
w/o	3.63	3.51	0.862
w	3.88	3.56	0.882

3.3. Ablations

The system is trained on MFCC20 with pitch augmentation, the temporal downsampling rate is 8, and the content consistency loss is conducted unless otherwise specified.

Table 2 summarises the ablation of different feature representations. Though the model trained on the MEL80 features obtains a competitive score in the naturalness test, it fails in cross-gender conversion. The MEL80 representations have sufficient timbre and pitch information, which may flow to the decoder and harm the conversion. Results show that the model trained on MFCC20 performs better than the MEL80 in both MOS and NCC results. And the intonation and similarity can be further improved in +Aug.

Three temporal down-sampling rates, 1, 8, and 32, are compared. Results shown in Table 3 indicate the temporal down-sampling rate of 8 is more suitable. When the down-sampling rate is set to 1, some mumbling cases are observed. We deduce that the whole temporal resolution setting causes the encoder outputs containing the source pitch and timbre information to a certain extent, harmful to conversion. However, some mispronunciation cases are observed when the down-sampling rate is set to 32, especially in fast-tempo songs. It means the temporal down-sampling rate is too wide for the system to retain content information.

Ablation results of the content consistency loss are shown in Table 4. Without the $Lconsist$, the system can also make within and cross-gender conversions. But the system with the $Lconsist$ performs better since it minimizes the mumbling and out-of-key problems, which can be found in that without $Lconsist$ loss. It indicates the limitation between source and reconstructed ones are useful for content consistency.

4. CONCLUSION

In this paper, a novel unsupervised singing voice conversion method named K-Converter is proposed. The content consistency loss, the temporal down-sampling, and low dimensional MFCC features assist feature disentanglement. Experiments indicate our method outperforms the state-of-art unsupervised singing voice conversion methods. In future work, we will continue to improve the quality of converted audio and attempt to extend our approach to speech-to-sing tasks.

5. REFERENCES

[1] Fernando Villavicencio and Jordi Bonada, "Applying voice conversion to concatenative singing-voice synthesis," in *InterSpeech*. ISCA, 2010, pp. 2162–2165.

[2] Kazuhiro Kobayashi, Tomoki Toda, Graham Neubig, Sakriani Sakti, and Satoshi Nakamura, "Statistical singing voice conversion with direct waveform modification based on the spectrum differential," in *InterSpeech*. ISCA, 2014, pp. 2514–2518.

[3] Kazuhiro Kobayashi, Tomoki Toda, Graham Neubig, Sakriani Sakti, and Satoshi Nakamura, "Statistical singing voice conversion based on direct waveform modification with global variance," in *InterSpeech*. ISCA, 2015, pp. 2754–2758.

[4] Xin Chen, Wei Chu, Jinxi Guo, and Ning Xu, "Singing voice conversion with non-parallel data," in *MIPR*. IEEE, 2019, pp. 292–296.

[5] Adam Polyak, Lior Wolf, Yossi Adi, and Yaniv Taigman, "Unsupervised cross-domain singing voice conversion," in *InterSpeech*. ISCA, 2020, pp. 2583–2587.

[6] Zhonghao Li, Benlai Tang, Xiang Yin, Yuan Wan, Ling Xu, Chen Shen, and Zejun Ma, "Ppg-based singing voice conversion with adversarial representation learning," in *ICASSP*. IEEE, 2021, pp. 7073–7077.

[7] Eliya Nachmani and Lior Wolf, "Unsupervised singing voice conversion," in *InterSpeech*. ISCA, 2019, pp. 2583–2587.

[8] Chengqi Deng, Chengzhu Yu, Heng Lu, Chao Weng, and Dong Yu, "Pitchnet: Unsupervised singing voice conversion with pitch adversarial network," in *ICASSP*. IEEE, 2020, pp. 7749–7753.

[9] Yin-Jyun Luo, Chin-Cheng Hsu, Kat Agres, and Dorien Herremans, "Singing voice conversion with disentangled representations of singer and vocal technique using variational autoencoders," in *ICASSP*. IEEE, 2020, pp. 3277–3281.

[10] Junchen Lu, Kun Zhou, Berrak Sisman, and Haizhou Li, "Vaw-gan for singing voice conversion with non-parallel training data," in *APSIPA*. IEEE, 2020, pp. 514–519.

[11] Takuhiro Kaneko and Hirokazu Kameoka, "Parallel-data-free voice conversion using cycle-consistent adversarial networks," *arXiv preprint arXiv:1711.11293*, 2017.

[12] Gunnar Fant, *Acoustic theory of speech production*, Number 2. Walter de Gruyter, 1970.

[13] Taejun Bak, Jae-Sung Bae, Hanbin Bae, Young-Ik Kim, and Hoon-Young Cho, "Fastpitchformant: Source-filter based decomposed modeling for speech synthesis," in *InterSpeech*. ISCA, 2021, pp. 116–120.

[14] Kaizhi Qian, Yang Zhang, Shiyu Chang, Xuesong Yang, and Mark Hasegawa-Johnson, "Autovc: Zero-shot voice style transfer with only autoencoder loss," in *ICML*. PMLR, 2019, pp. 5210–5219.

[15] Ju-chieh Chou, Cheng-chieh Yeh, and Hung-yi Lee, "One-shot voice conversion by separating speaker and content representations with instance normalization," in *InterSpeech*. ISCA, 2019, pp. 664–668.

[16] Zhiyan Duan, Haotian Fang, Bo Li, Khe Chai Sim, and Ye Wang, "The nus sung and spoken lyrics corpus: A quantitative comparison of singing and speech," in *APSIPA*. IEEE, 2013, pp. 1–9.

[17] Masanori Morise, Fumiya Yokomori, and Kenji Ozawa, "World: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[18] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto, "librosa: Audio and music signal analysis in python," in *Proceedings of the 14th python in science conference*. Citeseer, 2015, vol. 8, pp. 18–25.

[19] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim, "Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram," in *ICASSP*. IEEE, 2020, pp. 6199–6203.

[20] Jiawei Chen, Xu Tan, Jian Luan, Tao Qin, and Tie-Yan Liu, "Hifisinger: Towards high-fidelity neural singing voice synthesis," *arXiv preprint arXiv:2009.01776*, 2020.

[21] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," in *ICLR*, 2015.