The HCCL-CUHK System for the Voice Conversion Challenge 2018

Songxiang Liu[1], *Lifa Sun*[1,2], *Xixin Wu*[1], *Xunying Liu*[1] *and Helen Meng*[1]

[1]Human-Computer Communications Laboratory
Department of Systems Engineering and Engineering Management,
The Chinese University of Hong Kong, Shatin, N.T., Hong Kong SAR, China
[2]SpeechX Limited, Shenzhen, China

{sxliu, lfsun, wuxx, xyliu, hmmeng}@se.cuhk.edu.hk

Abstract

This paper presents the HCCL-CUHK system for the Voice Conversion Challenge 2018 (the VCC 2018), which is mainly characterized by doing Voice Conversion (VC) with non-parallel training data using Phonetic PosteriorGrams (PPGs). We propose to use Short-Time Fourier Transform Magnitudes (STFTMs) to synthesize converted speech waveforms with the Griffin-Lim algorithm. To fully exploit the different harmonic structure across different frequencies in the STFTMs, we partition the whole-band STFTMs into multiple overlapping frequency bands. Deep Bidirectional LSTM based RNNs (DBLSTM) have been shown to be able to model well the nonlinear mapping from PPGs to acoustic features in VC systems. However, training and conversion are very slow using such RNN models. To tackle this, the proposed system adopted Convolution Neural Networks (CNNs) with Gated Linear Units (GatedCNNs) to replace DBLSTMs. The VCC 2018 perceptual results show that the proposed system can achieve higher naturalness and similarity performance than the average performance in the non-parallel VC task.

Index Terms: voice conversion challenge 2018, phonetic posteriorgrams, subband, STFTMs, GatedCNNs

1. Introduction

Speech is the most convenient medium of communication among humans, and also enables natural human-computer interaction. Speaker identity corresponding to the speaker's voice conveys important non-linguistic information. However, varieties of voice characteristics, such as voice timbre and fundamental frequency (F0) patterns, are always restricted by speaker's own physical constraints [1]. If a system can generate speech with different speaker identity, a new way of human-computer interaction will be paved, it will lead to a vast array of potential applications.

Voice Conversion (VC) is a promising way to modify and synthesize speech with the desired speaker identity. The goal of VC is to modify a speech signal uttered by a source speaker to sound as if it was uttered by a target speaker. Various methods have been proposed for VC. The two most popular streams of VC techniques nowadays are Gaussian Mixture Model (GMM)-based methods and Artificial Neural Network (ANN)-based methods. GMM-based methods for VC are statistical approaches, which develop a conversion function mapping the source-target feature vectors [2]. To improve the performance of GMM-based method, global variance was used in [3] to alleviate the over-smoothing effect. In [4], a non-negative matrix factorization-based method is proposed, which uses speech exemplars to synthesize converted speech directly. On the other hand, various ANN architectures have been proposed for VC as well: feedforward neural networks [5, 6, 7, 8, 9], restricted Boltzmann Machines (RBMs) and their variations [10, 11, 12] and Recurrent Neural Networks [13, 14, 15].

Conventional vocoders such as STRAIGHT [16] and WORLD [17] have been frequently used in Text-to-Speech Synthesis (TTS) and VC systems. Such vocoders use Mel-Cepstral coefficients (MCEPs), the fundamental frequency (F0) and the aperiodic components (AP) to synthesize speech waveforms. There is a big gap in similarity and naturalness between the original natural voice and the converted voice using such vocoders. Short-Time Fourier Transfrom Magnitudes (STFTMs) have high inter-correlation parameters, which retain rich spectrum information. With the prevalence of neural network techniques and their ability to handle high-dimensional data, STFTMs are becoming more popular. These features were used successfully in the end-to-end TTS system recently [18], where the Griffin-Lim algorithm [19] was used to convert STFTMs to time-domain audio waveforms. The Griffin-Lim algorithm iteratively estimates the unkown phases by repeatedly converting between frequency and time domain representations of the signal using the short-time Fourier transform and its inverse. The iteration process is able to converge within tens of iterations; thus, the Griffin-Lim algorithm is very efficient. To the best of our knowledge, this paper is the first one using STFTMs for VC tasks.

Deep Bidirectional LSTM based RNNs (DBLSTM) architecture including memory blocks and peephole connections makes it possible to learn long-range context-dependencies, which has been shown useful for VC task in the previous work [14, 15]. Training such RNN models, however, is rather slow because the next output depends on the previous hidden states which does not enable parallelization along the inputs. Compared to RNNs, CNNs can be trained much faster. The receptive context of CNNs is of fixed size, however, the effective context can be made larger by stacking several layers on top of each other. While training a deeper network is harder, adding gating mechanisms (GatedCNNs) [20] can allow the network to control what information should be propagated through the hierarchy of layers. GatedCNNs have been successfully applied to do language modeling [20] and sequence to sequence modeling [21].

Our system submitted to the Voice Conversion Challenge 2018 (the VCC 2018) [22] is based on the system proposed in [15] with many new features. We replace DBLSTM with GatedCNNs, which greatly accelerates the training process and the

10.21437/Odyssey.2018-35

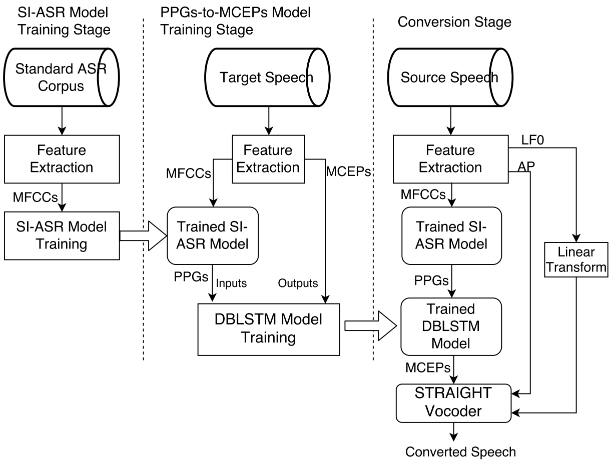

Figure 1: *Schematic diagram of VC model based on DBLSTM using Phonetic PosteriorGrams (PPGs).*

conversion process. We adopt the Griffin-Lim algorithm to synthesize converted speech waveform from predicted STFTMs. To fully exploit the difference in harmonic structure across different frequencies in the STFTMs, we partition the whole-band STFTMs into multiple overlapping frequency subbands and train a GatedCNN for each subband.

The rest of this paper is orgnized as follows: Section 2 introduces the baseline model proposed in [15]. Section 3 describes the proposed system. Section 4 shows the network architecture details and the experimental evaluations. Section 5 concludes this paper.

2. Baseline Model

2.1. Overview

The baseline model uses Phonetic PosteriorGrams (PPGs), which is a time-versus-class matrix representing the posterior probabilities of each phonetic class for each specific time frame of one utterance [23, 24]. The DBLSTM architecture is used to map PPGs to MCEPs. As illustrated in Figure 1, the baseline model consists of three parts: Speaker-independent Automatic Speech Recognition (SI-ASR) model training stage, PPGs-to-MCEPs model training stage and conversion stage. The SI-ASR model is trained for extracting PPGs from speech data. PPGs-to-MCEPs model training stage is to learn the non-linear mapping between PPGs and MCEPs. At the conversion stage, the trained PPGs-to-MCEPs mapping is used to convert the PPGs extracted from the source speech to the target MCEPs features, which are combined together with the linear-transformed Log F0 and AP to synthesize the target speech using the STRAIGHT vocoder.

2.2. Training Stage and Conversion Stage

At the SI-ASR model training stage, a DNN-based speaker-independent ASR system is trained using a standard multi-speaker ASR corpus. After training, T time-steps of MFCC feature vectors, which can be denoted as $X = (X_1, \cdots, X_T)$, are mapped to the posterior probabilities $P = (P(C_1|X), \cdots, P(C_T|X))$, where $P(C_i|X), i = 1, \cdots, T$ is a d-dimension probability vector and d is the number of senone classes when training the SI-ASR model.

At the PPGs-to-MCEPs model training stage, the input of the DBLSTM model are PPGs, computed by the trained SI-ASR model. The output of this model is the MCEP features of the target speaker. Note that the input PPGs and the output MCEPs are both from the target speaker. As PPGs are assumed to only contain speaker-independent articulation information, they can be used to bridge across speakers and thus achieve non-parallel voice conversion [15]. We denote $(Y_1, \cdots, Y_t, \cdots, Y_T)$ as the MCEPs sequence extracted from the target speech data, and denote $(\hat{Y}_1, \cdots, \hat{Y}_t, \cdots, \hat{Y}_T)$ as the output of the PPGs-to-MCEPs model. The cost function of the baseline model is:

$$\sum_{t=1}^{T}(\hat{Y}_t - Y_t)^2 \qquad (1)$$

where T is the number of frames of one utterance.

At the conversion stage, given one source utterance, its MFCCs, F0 and AP are extracted. The MFCCs are put into the trained SI-ASR model as inputs to get the PPGs, which are mapped to MCEPs by the trained DBLSTM model. F0 in logarithmic scale (Log F0) is converted by equalizing the mean and the standard deviation of the source and target utterances. AP is directly copied through. Finally, the STRAIGHT vocoder is used to synthesize the converted speech waveform.

3. Description of the Proposed System

3.1. Overview

The overall schematic diagram of the proposed system is shown in figure 2, where STFTMs are used as the output features and the Griffin-Lim algorithm is applied to synthesize waveforms from the predicted STFTMs. To exploit the different harmonic structure across different frequency bins in the STFTMs, the proposed system partitions the whole-band STFTMs into multiple frequency bands, on each of which we train a GatedCNN model.

3.2. Vocoder Free

The proposed submitted system adopts the Griffin-Lim algorithm to synthesize time-domain audio waveforms from the STFTMs. The Griffin-Lim algorithm iteratively estimates a signal from its modified STFT magnitude by minimizing the mean squared error (MSE) between the estimated STFTMs and the modified STFTMs. It has been shown in the original paper that it decreases the MSE loss monotonically in each iteration, and converges after several tens of iterations. Hence, the Griffin-Lim algorithm is an efficient algorithm and our experiments show that it can decrease the conversion time delay comparing to using conventional vocoder such as STRAIGHT.

High dimensional STFT spectrograms are necessary to synthesize high-quality speech waveforms. At the same time, STFTMs have different harmonic structures across frequency bands: clearer harmonic structure in the low-frequency band, and less so in the high-frequency band. During training, one can give different weights to different frequency bins when computing the loss. As the amount of training data for each speaker is limited in the VCC 2018, high dimensionality of the whole-band STFTMs means that we need a big network to do the PPGs-to-STFTMs mapping, which will cause to perform the over-fitting problems. Thus, inspired by [25], we partition the whole-band STFTMs into multiple frequency subbands, and for each frequency subband we train a small network to perform the

PPGs-to-STFTMs mapping. The details of each step are as follows:

Partition: The whole-band STFTMs are devided into N overlapping frequency subbands, each of which ranges from the f_i^s-th to f_i^e-th frequency, $i = 1, \cdots, N$. The overlap between the i-th and $i+1$-th bands is set at v_i, i.e., $v_i = f_i^e - f_{i+1}^s$. The overlap is set to smoothly concatenate the individual bands during conversion stage.

Mapping: For the input PPGs, we do not conduct any partitioning. The input features for each of the N sub-models are the same. Our experiment shows that injecting fundamental frequency information into the inputs (PPGs) can impove the prosody of the generated output. The reason is that when we extract PPGs from the speech data, we drop the fundamental frequency (F0) information. It is expected that the converted speech will have more natural F0 trajectory when we inject F0 information into the input side.

Concatenation: To mitigate the possible discontinuity when connecting the predicted STFTM subbands, we apply a window function (e.g., Hamming window) to both ends of each subband before concatenation, where the window width is $2v_i$ and half of the window function is applied to each end.

3.3. Convolution Framework

Although RNNs with LSTM [26] can theoretically model arbitrarily long context dependencies, which is very important for VC systems using PPGs that contain rich implicit linguistic information, the fact that the next output depends on the previous hidden state does not enable parallelization over the elements of a sequence. Thus, training and conversion processes can be slow using such RNN frameworks. Convolutional neural networks, however, can be trained in a very fast parallel way since the computation of all input frames can be performed simultaneously. Compared to RNNs, convolutions create representations for fixed size contexts, however, the effective context size of the network can easily be made larger by stacking several layers on top of each other. While deeper convolution networks are harder to train, a simple gating mechanism has been shown very effective for the training process [20].

The proposed system uses GatedCNNs proposed by [20] to perform the PPGs-to-STFTMs mapping. The input to the model is PPGs concatenated with Log F0 frame by frame. We compute the hidden layers $h_0, \cdots, h_L$ as

$$h_l(X) = (X * W_{f,l} + b_{f,l}) \otimes \sigma(X * W_{g,l} + b_{g,l}) \qquad (2)$$

where $*$ denotes one-dimensional convolution operation, $\otimes$ denotes the element-wise product between matrices, σ is the sigmoid function, l is the layer index, f and g denote filter and gate, respectively, W and b are learnable parameters.

3.4. Training and Conversion

As shown in Figure 2, the proposed system is divided into SI-ASR model training stage, PPGs-to-STFTMs model training stage and conversion stage. The SI-ASR model is in the same set as in the baseline model, which is trained using a standard ASR corpus.

At the PPGs-to-STFTMs model training stage, we first extract MFCCs, F0 and STFTMs from a batch of target speech data. Then the trained SI-ASR model computes the corresponding PPGs using the extracted MFCCs. The STFTMs of the entire frequency band are partitioned into N overlapping frequency bands, for each of which we train a GatedCNN network.

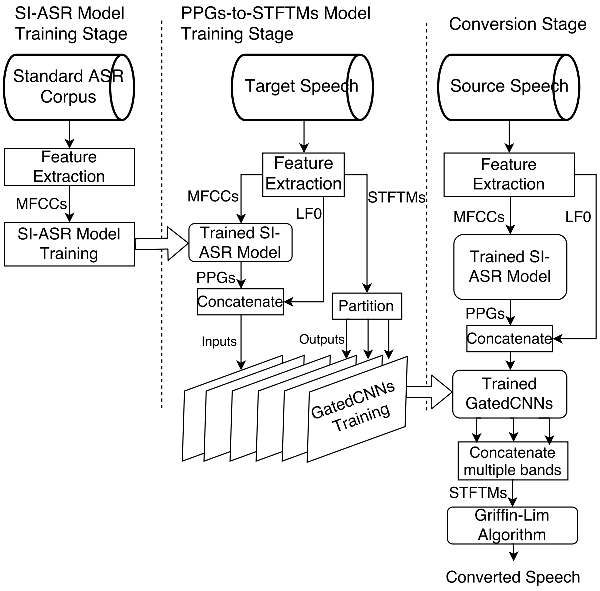

Figure 2: *Schematic diagram of the proposed model. GatedCNNs stand for Convolutional Neural Networks (CNNs) with Gated Linear Units (GLU), while STFTMs stand for Short-Time Fourier Transform Magnitudes.*

In total, we train N GatedCNN networks which have the same inputs, i.e., PPGs concatenated with Log F0 frame by frame. The output of each of these N GatedCNN networks is one frequency subband of the target STFTMs. The simple $l1$ loss is used as the objective function to train the N GatedCNN networks, which can be denoted as

$$\sum_{t=1}^{T} |\hat{\mathbf{Y}}_{\mathbf{t}}^{\mathbf{i}} - \mathbf{Y}_{\mathbf{t}}^{\mathbf{i}}| \qquad (3)$$

where $(\mathbf{Y}_{\mathbf{1}}^{\mathbf{i}}, \cdots, \mathbf{Y}_{\mathbf{t}}^{\mathbf{i}}, \cdots, \mathbf{Y}_{\mathbf{T}}^{\mathbf{i}})$ is the i-th frequency band of the target STFTMs sequence extracted from the target speech data, while $(\hat{\mathbf{Y}}_{\mathbf{1}}^{\mathbf{i}}, \cdots, \hat{\mathbf{Y}}_{\mathbf{t}}^{\mathbf{i}}, \cdots, \hat{\mathbf{Y}}_{\mathbf{T}}^{\mathbf{i}})$ is the corresponding model outputs. T is the number of time frames of one utterance.

At the conversion stage, given one source utterance, its MFCCs and F0 are firstly extracted. Then MFCCs are used to get PPGs by the trained SI-ASR model. PPGs and Log F0 are then concatenated as the input of the N PPGs-to-STFTMs networks. Then, the N output frequency subbands are concatenated after Hamming windowing to obtain the whole-band STFTMs. Finally, we adopt the Griffin-Lim algorithm to synthesize the converted speech waveforms from the STFTMs.

4. Experiments

In this section, we show results of the VCC 2018 to demonstrate the performance of the proposed system. To conveniently compare performance among models that we have tried, we name the models as follows:

- LSTM-MCEP: The baseline model mentioned in section 2.
- GCNN-MCEP: Bidirectional LSTM replaced with GatedCNN in the baseline model.

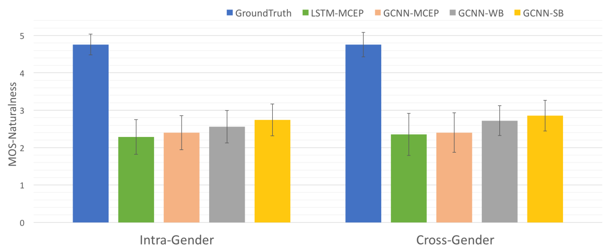

Figure 3: *Mean Opinion Score (MOS) test results for comparing the naturalness perfornance among LSTM-MCEP, GCNN-MCEP, GCNN-WB and GCNN-SB. GroundTruth means target recording.*

- GCNN-WB: Input PPGs concatenated with Log F0 with whole-band STFTMs as output.
- GCNN-SB: Proposed system submitted to VCC 2018, as described in Section 3.

4.1. Dataset Description

All the models compared in this paper use the VCC 2018 training dataset. The training set contains 12 speakers (6 females + 6 males) in total, each of which has 81 recordings. Eight speakers have the same set of 81 text prompts, which are meant to be used for the parallel VC task, while the remaining four speakers have distinct sentences for non-parallel VC task. The proposed system took part in both the parallel and non-parallel VC tasks, hence, we utilize all the released training data. The recordings have a sampling rate of 22.05kHZ with mono channel.

We down-sampled the speech data to 16kHZ and trimmed 80 percent of the leading and tailing silent frames. Acoustic features, including the spectral envelope, F0 (1 dimension) and AP (513 dimensions) are extracted by STRAIGHT analysis using 25-ms window size and 5-ms window shift. The 39th order MCEPs plus Log energy are extracted to represent the spectral envelope. We use Log magnitude spectrogram with Hanning windowing, 25-ms window size, 5-ms window shift and 1024-point Fourier transform to get the STFTMs. We also pre-emphasized (0.97) waveforms before Fourier transform. We normalized PPGs, MCEPs, STFTMs and Log F0 to have zero mean and unit variance. We raised the predicted STFTMs by a power of 1.35 before feeding to Griffin-Lim algorithm to reduce artifacts.

4.2. Model Setup

All models mentioned in this paper use a same set of SI-ASR system, which is implemented using the Kaldi speech recognition toolkit [27] with the TIMIT corpus [28]. The system has a DNN architecture with 4 hidden layers each of which contains 1024 units. Senones are treated as the phonetic class of PPGs. The number of senone classes is 131, which are obtained by clustering at the SI-ASR training stage.

As for the training processes for the PPGs-to-MCEPs mapping models as well as the PPGs-to-STFTMs mapping models, there is no difference between the parallel and the non-parallel VC tasks. The model descriptions which will be given in the sequel are applicable for both the parallel and non-parallel VC tasks.

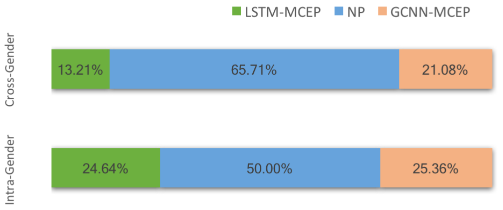

Figure 4: *ABX similarity test results for model LSTM-MGC and model GCNN-MGC. NP means no preference.*

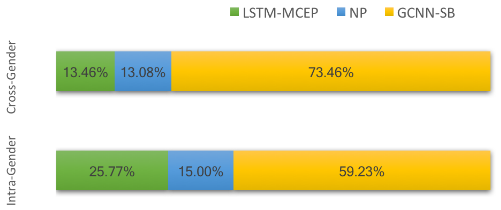

Figure 5: *ABX similarity test results for model LSTM-MGC and model GCNN-SB. NP means no preference.*

The baseline model has 6 hidden layers: 1 dense layer with dropout [29], 4 bidirectional LSTM layers, and another dense layer with dropout. The number of units in each layer is [131, 128, 64, 64, 64, 64, 128, 39], where 64 represents the number of LSTM hidden units of one direction. We use the Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with learning rate 0.0001. No learning rate decay is applied. The dropout rates are both 0.5 during training. In the baseline model and all other mentioned models, we use a batch size of 16, where all sequences are padded to a max length of that batch and masked loss is applied during training. We trained the baseline model and all other models using 71 sentences, while the remaining 10 sentences were used for evaluation.

All the mentioned models use the same configured Gated-CNN framework: 1 dense layer with dropout (dropout rate is 0.5), 3 CNN layers with GLUs, and another dense layer with dropout (dropout rate is 0.5), where the number of units of each hidden layer is [256, 256, 256, 256, 128]. The CNN layers use kernels of size 5. The optimizers and the learning rates are the same as in the baseline model, which are Adam optimizer ($\beta_1 = 0.9$, $\beta_2 = 0.999$) and 0.0001 respectively. No learning rate decay is applied.

For the subband model (GCNN-SB), as the STFTMs have finer harmonic structure in lower frequencies, the frequency bands expand narrower frequency range at the lower frequencies. We divided the STFTMs into 6 overlapping frequency subbands, which are [(0, 66), (34, 116), (84, 166), (134, 216), (184, 316), (284, 513)] with overlap $v_i = 32$, for $i = 1, \cdots, 6$. We connected the converted frequency subbands with Hamming windows which have window width of 64 and half of the window function is applied to each band end.

4.3. Model Selection

We investigated four different models: LSTM-MCEP, GCNN-MCEP, GCNN-WB and GCNN-SB. The model setup details have been given in section 4.2. We conducted a MOS test to compare the naturalness of the output generated by these models, where each model produces 20 intra-gender converted utterances and 20 cross-gender converted utterances. The target recordings are also included in the evaluation set. 16 subjects partipated in the MOS test and the results are given in Figure 3. We only conducted ABX similarity tests for LSTM-MCEP versus GCNN-MCEP and LSTM-MCEP versus GCNN-SB. 20 intra-gender converted utterances and 20 cross-gender converted utterances were used for each model in the ABX similarity tests. 14 subjects did the ABX similarity tests and the results are shown in Figure 4 and Figure 5.

The MOS test results in Figure 3 show that using Gated-CNN instead of bidirectional LSTM can increase the naturalness performance by comparing LSTM-MCEP with GCNN-MCEP. From figure 4, adopting GatedCNN can also increase the similarity score, though the improvement is not so significant. We can also see from Figure 3 that using STFTMs (the model GCNN-WB and GCNN-SB) can raise the naturalness performance compared with using MCEPs as converted features. By comparing the MOS scores of GCNN-WB and GCNN-SB, we observe that partioning the whole-band STFTMs into multiple overlapped frequency subbands and training them independently can improve the naturalness performance of the converted speech. Figure 5 shows the ABX similarity test results comparing the baseline model (LSTM-MCEP) and the proposed model (GCNN-SB), which demonstrates that GCNN-SB model can significantly improve the similarity performance compared to LSTM-MCEP.

Moreover, the GatedCNN architecture adopting the Griffin-Lim algorithm to synthesize converted speech from STFTMs can speed up the training process and the conversion process. Table 1 shows the training and conversion speed among four models, where speech data from one particular target speaker was used to train each model. During conversion, the same 10 utterances from a source speaker are converted. The CPU we used is dual Intel Xeon E5-2640, 8 cores, 2.6 GHZ and the GPU is a NVIDIA Tesla K40. The four models were implemented using TensorFlow. We set the iteration times of the Griffin-Lim algorithm to be 50. The version of STRAIGHT we use is V40-006b. We can see in Table 1 that using GatedCNN can greatly speed up the training process. Using the Griffin-Lim algorithm to synthesize converted speech from STFTMs is also faster than adopting STRAIGHT to synthesize with MCEPs, F0 and AP. We can compare the conversion speed between GCNN-MGC and GCNN-WB: Converting 10 utterances by GCNN-MCEP takes 47.008 seconds which contains 13.321 seconds to compute convered MCEPs and 33.687 seconds to synthesize speech using STRAIGHT, while using GCNN-WB only takes 35.799 seconds which contains 14.725 seconds to obtain predicted STFTMs and 21.074 seconds to synthesize speech with Griffin-Lim algorithm. From the comparison above, we observe that synthesizing with Griffin-Lim algorithm is indeed faster than using STRAIGHT vocoder.

Therefore, taking into account the MOS and ABX tests perfomance as well the training and conversion speed, we chose the GCNN-SB system as the submitted system to the VCC 2018.

4.4. Results at the Voice Conversion Challenge 2018

Our submitted system took part in both the parallel (HUB) and non-parallel (SPOKE) VC tasks. The challenge organizers conducted a large-scale perceptual experiments by crowdsourcing with native English speakers and using a web interface. 106 listeners (49 Female, 57 Male) have taken part in the evaluation. The test consisted of two sections: Mean Opinion Score (MOS) test and Similarity test. The standard MOS test was adopted. In the similarity test, listeners were given pairs of stimuli and asked to determine whether these two samples have been produced by the same speaker.

Table 1: *Training and Conversion speed of the four models*

Models	Training, sec/step	Conversion, sec/(ten utterances)
LSTM-MCEP	8.324	66.828
GCNN-MCEP	0.247	47.008
GCNN-WB	0.331	35.799
GCNN-SB	0.256	44.330

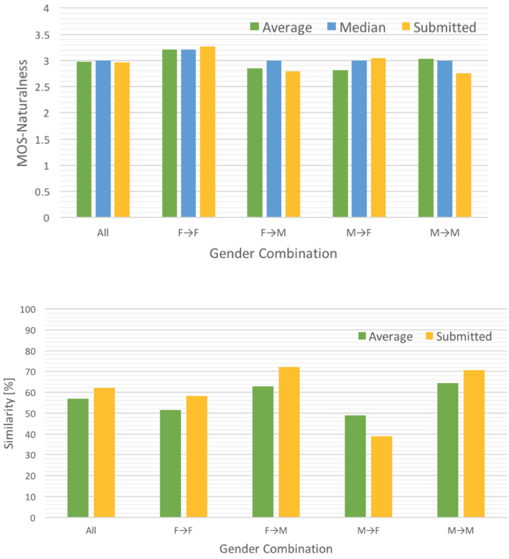

Figure 6: *Naturalness (top) and similarity (bottom) performance of our submitted system in the parallel VC task. F: female; M: male; All means average performance across all speaker combinaitons; F → M means that source speaker is female and target speaker is male and etc.*

The evaluation results for the parallel VC task and the non-parallel VC task are shown in subsection 4.4.1 and subsection 4.4.2 respectively.

4.4.1. Parallel VC task evaluation results

The performance of our submitted system for the parallel VC task is shown in Figure 6, where scores are compared for each gender combination with the average score, the median score and our submitted system score. In terms of naturalness, the submitted system is near the average naturalness performance across all submitted systems. The similarity performance of our submitted system outperforms the average similarity performance in all the gender pairs and is 6% higher than the average similarity score across all submitted systems. However, the speech quality and similarity of the converted voices are still obviously degraded compared to that of natural voices. One reason for such degradation is that we use only 1024-point Fourier transform to get the STFTMs due to the limited amount of train-

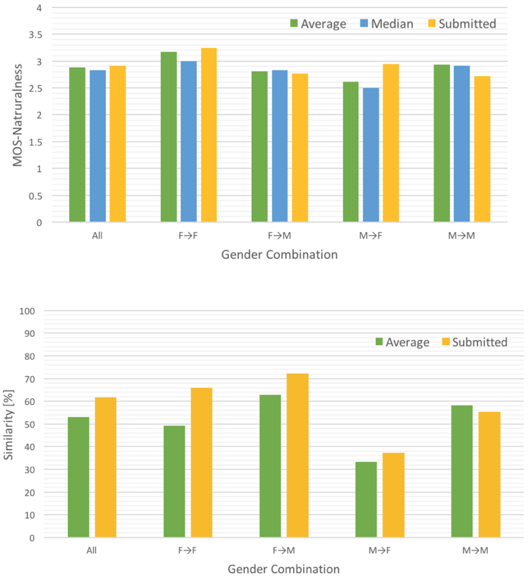

Figure 7: *Naturalness (top) and similarity (bottom) performance of our submitted system in the non-parallel VC task. F: female; M: male; All means average performance across all speaker combinaitons; FM means that source speaker is female and target speaker is male and etc.*

ing data. Our prelimilary experiments show that when we use 2048-point Fourier transform and train the model using 1000 utterances, the quality and similarity performance is much better than what is showed here. Note that with only 1024-point STFTMs, the submitted system can still achieve acceptable similarity performance, which indicates that STFTMs retain good speech identity information such as voice timbre information.

4.4.2. Non-parallel VC task evaluation results

The perceptual evaluation results of our submitted system for the non-parallel VC task is illustrated in Figure 7, where scores are compared for each gender combination with the average score, the median score and our submitted system score. Figure 8 shows the position of our submitted system (N18) in the similarity score versus MOS score scatter plot among all the submitted systems. The MOS score which reflects the naturalness performance of our submitted system is slightly higher than the average naturalness performance among all submitted systems. As for the similarity score, our submitted system is higher than the average similarity performance among all systems submitted to the non-parallel VC task by about 8%. In the non-parallel VC task, our submitted system achieves the highest among all gender combinations in both the naturalness and similarity performance when the conversion gender pair is Female-to-Female.

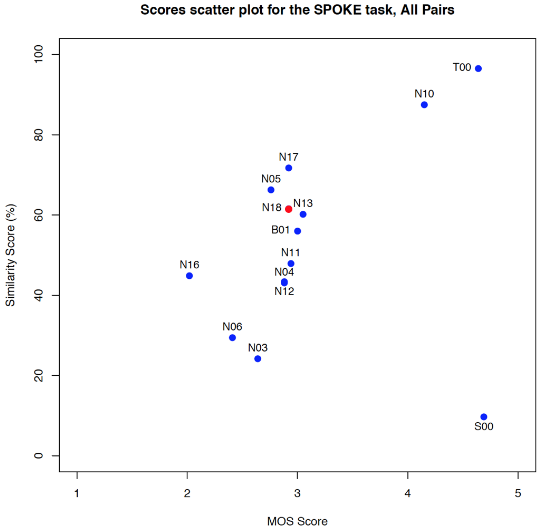

Figure 8: *Similarity score vs. MOS score scatter plot for the non-parallel VC task results for all the submitted systems. Our submitted system is marked in red. S00: Source recording; T00: Target recording.*

5. Conclusions

In this paper, we presented our submitted system to the VCC 2018. we applied the Griffin-Lim algorithm to synthesize the converted speech waveforms from STFTMs. To exploit the harmonic structure of the STFTMs, we partitioned the whole-band STFTMs into multiple frequency subbands. We adopted GatedCNN networks to learn the nonlinear mapping between PPGs and STFTMs, where we also injected Log F0 information into the input side. The percepual evaluation tests conducted by the challenge organizers showed that our submitted system achieved average naturalness among all submitted systems. But for similarity performance, our submitted system achieved much higher performance than the average similarity score.

Future work will include exploring the possibility of using speech data from multiple speakers to train a single model when the training data for a single speaker is limited.

6. Acknowledgements

This project is partially supported by a grant from the NSFC/RGC Joint Research Scheme (Project No. N_CUHK404/15).

7. References

[1] Kazuhiro Kobayashi, Shinnosuke Takamichi, Satoshi Nakamura, and Tomoki Toda, “The nu-naist voice conversion system for the voice conversion challenge 2016.,” in *INTERSPEECH*, 2016, pp. 1667–1671.

[2] Yannis Stylianou, Olivier Cappé, and Eric Moulines, “Continuous probabilistic transform for voice conversion,” *IEEE Transactions on speech and audio processing*, vol. 6, no. 2, pp. 131–142, 1998.

[3] Tomoki Toda, Alan W Black, and Keiichi Tokuda, “Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2222–2235, 2007.

[4] Zhizheng Wu, Tuomas Virtanen, Tomi Kinnunen, Eng Siong Chng, and Haizhou Li, “Exemplar-based voice conversion using non-negative spectrogram decon-

volution," in *Eighth ISCA Workshop on Speech Synthesis*, 2013.

[5] Srinivas Desai, Alan W Black, B Yegnanarayana, and Kishore Prahallad, "Spectral mapping using artificial neural networks for voice conversion," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 954–964, 2010.

[6] Elias Azarov, Maxim Vashkevich, Denis Likhachov, and Alexander A Petrovsky, "Real-time voice conversion using artificial neural networks with rectified linear units.," in *INTERSPEECH*, 2013, pp. 1032–1036.

[7] Li-Juan Liu, Ling-Hui Chen, Zhen-Hua Ling, and Li-Rong Dai, "Using bidirectional associative memories for joint spectral envelope modeling in voice conversion," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 7884–7888.

[8] Seyed Hamidreza Mohammadi and Alexander Kain, "Voice conversion using deep neural networks with speaker-independent pre-training," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 19–23.

[9] Jagannath Nirmal, Mukesh Zaveri, Suprava Patnaik, and Pramod Kachare, "Voice conversion using general regression neural network," *Applied Soft Computing*, vol. 24, pp. 1–12, 2014.

[10] Ling-Hui Chen, Zhen-Hua Ling, Yan Song, and Li-Rong Dai, "Joint spectral distribution modeling using restricted boltzmann machines for voice conversion.," in *Interspeech*, 2013, pp. 3052–3056.

[11] Zhizheng Wu, Eng Siong Chng, and Haizhou Li, "Conditional restricted boltzmann machine for voice conversion," in *Signal and Information Processing (ChinaSIP), 2013 IEEE China Summit & International Conference on*. IEEE, 2013, pp. 104–108.

[12] Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki, "Sparse nonlinear representation for voice conversion," in *Multimedia and Expo (ICME), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1–6.

[13] Toru Nakashika, Tetsuya Takiguchi, and Yasuo Ariki, "Voice conversion using rnn pre-trained by recurrent temporal restricted boltzmann machines," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 23, no. 3, pp. 580–587, 2015.

[14] Lifa Sun, Shiyin Kang, Kun Li, and Helen Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4869–4873.

[15] Lifa Sun, Kun Li, Hao Wang, Shiyin Kang, and Helen Meng, "Phonetic posteriorgrams for many-to-one voice conversion without parallel data training," in *Multimedia and Expo (ICME), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–6.

[16] Hideki Kawahara, Ikuyo Masuda-Katsuse, and Alain De Cheveigne, "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds1," *Speech communication*, vol. 27, no. 3-4, pp. 187–207, 1999.

[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] Yuxuan Wang, RJ Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, et al., "Tacotron: A fully end-to-end text-to-speech synthesis model," *arXiv preprint arXiv:1703.10135*, 2017.

[19] Daniel Griffin and Jae Lim, "Signal estimation from modified short-time fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 2, pp. 236–243, 1984.

[20] Yann N Dauphin, Angela Fan, Michael Auli, and David Grangier, "Language modeling with gated convolutional networks," *arXiv preprint arXiv:1612.08083*, 2016.

[21] Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N Dauphin, "Convolutional sequence to sequence learning," *arXiv preprint arXiv:1705.03122*, 2017.

[22] Jaime Lorenzo-Trueba, Junichi Yamagishi, Tomoki Toda, Daisuke Saito, Fernando Villavicencio, Tomi Kinnunen, and Zhenhua Ling, "The voice conversion challenge 2018: Promoting development of parallel and nonparallel methods.," *Submitted to Odyssey 2018*, 2018.

[23] Timothy J Hazen, Wade Shen, and Christopher White, "Query-by-example spoken term detection using phonetic posteriorgram templates," in *Automatic Speech Recognition & Understanding, 2009. ASRU 2009. IEEE Workshop on*. IEEE, 2009, pp. 421–426.

[24] Keith Kintzley, Aren Jansen, and Hynek Hermansky, "Event selection from phone posteriorgrams using matched filters," in *Twelfth Annual Conference of the International Speech Communication Association*, 2011.

[25] Takuhiro Kaneko, Shinji Takaki, Hirokazu Kameoka, and Junichi Yamagishi, "Generative adversarial network-based postfilter for stft spectrograms," in *Proceedings of Interspeech*, 2017.

[26] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[27] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.

[28] John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," *NASA STI/Recon technical report n*, vol. 93, 1993.

[29] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.