VSUGAN unify voice style based on spectrogram and generated adversarial networks

Tongjie Ouyang1,2*, Zhijun Yang2, Huilong Xie2, Tianlin Hu2 & Qingmei Liu3

In course recording, the audio recorded in different pickups and environments can be clearly distinguished and cause style differences after splicing, which influences the quality of recorded courses. A common way to improve the above situation is to use voice style unification. In the present study, we propose a voice style unification model based on generated adversarial networks (VSUGAN) to transfer voice style from the spectrogram. The VSUGAN synthesizes the audio by combining the style information from the audio style template and the voice information from the processed audio. And it allows the audio style unification in different environments without retraining the network for new speakers. Meanwhile, the current VSUGAN is implemented and evaluated on THCHS-30 and VCTK-Corpus corpora. The source code of VSUGAN is available at https://github.com/oy-tj/VSUGAN. In one word, it is demonstrated that the VSUGAN can effectively improve the quality of the recorded audio and reduce the style differences in kinds of environments.

In recent years, popular online courses like SPOC and the applications of flipped classrooms have brought a large number of demands for regular course recording1. The audio records in different pickups and recording environments may generate additional background noise, which can be clearly distinguished by human ears after splicing2, and lead to different sound qualities to affect recorded courses. Traditionally, this problem can be solved by manually adjusting the sound waveform or frequency spectrum during post-production or removing noise by denoise algorithms3,4. However, the involved post-production in regular course recording applications is usually unprofessional, and manual adjustment is time-consuming. Traditional denoising algorithms, such as spectral subtractive4, subspace6, statistical-model based7 and Wiener algorithms8 only remove part of the background noise and cannot solve the problem completely. Meanwhile, neuronal networks used for speech enhancement, such as SEGAN9,10, mostly focus on obtaining clear speech but fail to unify audio styles according to different environments.

Voice style unification also known as voice style transfer refers to combining a speaker's timbre, paralanguage (mood and intonation), and other characteristics into synthesized audio. Through decades of development, voice style transfer technology has obtained many achievements due to the applications of voice conversion technology. For instance, Valbret et al. proposed a method based on Pitch Synchronous Overlap and Add technology to realize voice transformation11, and Desai et al. used BP neural network to achieve speech conversion12. Thanks to the development of deep learning, especially long short-term memory networks, the performance has been improved significantly13. Moreover, in order to further enhance the quality of voice conversion, Donahue et al. presented wave GAN based on deep convolutional generative adversarial networks (DCGAN)14. However, existing generative adversarial networks (GAN) only solve the fixed one-to-one or many-to-many voice conversion scenarios. Once involving new speakers, the GAN should be retrained with transfer learning, which is reduplicative and unnecessary.

In the current study, a voice style unification model based on generated adversarial networks (VSUGAN) is established to unify voice style in different environments without retraining the network for new speakers. VSUGAN combines the style information from the audio style template and the voice information from the processed audio. In this method, background noise is also considered as a part of the audio style. The input consists of audio style template and noise-mixed audio, while the output is target-style audio. The contributions of this paper are as follows:

1Modern Educational Technology and Practice Training Center, Innovation Laboratory for Undergraduate, Xiamen University, Xiamen, China. 2School of Aerospace Engineering, Xiamen University, Xiamen, China. 3College of Food and Biological Engineering, Jimei University, Xiamen, China. *Email: oytj@xmu.edu.cn
(1) An audio style template is added as input. VSUGAN will adjust the audio style according to the template without retraining the network for new speakers.

(2) By making reasonable assumptions about the training data, unsupervised learning is transformed into supervised learning, so that a large number of existing corpuses can be used in training VSUGAN.

**Spectrogram and voiceprint.** The spectrogram is obtained by short-time Fourier transform (STFT) of the voice signal. The voice signal, i.e., waveform signal, is first divided into a number of overlapping frames according to the time window and then converted to the frequency spectrum by fast Fourier transform (FFT) frame by frame. Next, the frequency spectrum is arranged in the frame order to form a spectrogram. The x-axis of the spectrogram denotes time, and the y-axis stands for frequency. The amplitude of a particular frequency at a specific time is represented by colors, where dark colors correspond to low amplitudes and brighter colors relate to progressively stronger amplitudes. The change of background noise and voiceprint after a piece of audio through VSUGAN can be clearly observed in the spectrogram. In the experiment, the librosa library was utilized to transfer the waveform signals to spectrogram with the FFT window size of 512.

Meanwhile, voice waveforms and spectrograms can reflect different voice recording effects in diverse recording environments. Figure 1a shows the waveform and spectrogram of a piece of audio recorded in four different environments. The corresponding recording settings are shown in Table 1. It can be seen from Fig. 1a1–1–a4-1 that the waveforms of the same sentence are disparate in four different environments. And the same sentence voiceprint are vary in detail from office or classroom environments (Fig. 1a1-2–a4-2, a1-3–a4-3).

**Table 1.** Different recording environments.

| Group | Recording venue | Input devices                      |
|-------|-----------------|------------------------------------|
| a1    | Office          | Sanako SLH-07 (3.5 in. socket)     |
| a2    | Office          | Panasonic WX-4800 & X66            |
| a3    | Classroom       | Sanako SLH-07 (3.5 in. socket)     |
| a4    | Classroom       | Panasonic WX-4800 & X66            |

**Figure 1.** Effects of different recording environments and noises on audio. (a) Waveform (top row), spectrogram (middle row), and partial enlargement (bottom row) of recorded audio in different environments (each column means each environment). (b) Waveform (top row) and spectrogram (bottom row) of audio mixed with different noises (different columns).
Background noise. Background noise in different environments generates noise points with kinds of distributions and shapes on the spectrogram. Figure 1b shows the waveforms and spectrograms of the same audio after overlaying different noises, where b1 is the original clean audio, b2 is mixed with cafeteria noise, b3 is combined with driving car noise, and b4 is integrated with Gauss white noise. It is clear from b2-2, b3-2, and b4-2 that the frequency of cafeteria noise and car noise is concentrated in the low-frequency area, while the Gauss white noise is distributed in the whole frequency area.

Voice style unification based on GAN Algorithm and system structure. GAN consists of two parts: generator G and discriminator D. Generator G learns to map samples z from some prior distribution Pz to the target distribution PG. However, PG is unknown, and discriminator D is designed to judge the similarity between PG and prior distribution Pdata. Discriminator D is trained to distinguish samples x from the prior distribution Pdata and the fake samples generated by G. Similarly, Generator G is trained to make its output data G(z) deceive the discriminator D as much as possible, so that the discriminator D cannot distinguish whether the data comes from the prior distribution Pdata or PG. Alternately training D and G can increase their completion abilities until the data generated by G meets the requirements. This kind of adversarial training in classical GAN can be described as:

$$\min_G \max_D V(G, D) = E_{x \sim P_{data}} \left[ \log D(x) \right] + E_{z \sim P_z} \left[ \log (1 - D(G(z))) \right]$$

Assuming an extra input $x_i$ need to be added into the classical GAN, the output $G(\mathbf{z})$ of the generator can also have some properties related to $x_i$. Thus, the adversarial training with the addition of $x_i$ can be described as:

$$\min_G \max_D V(G, D) = E_{x \sim P_{data} \times x_i \sim P_{data}} \left[ \log D(x, x_i) \right] + E_{z \sim P_z, x_i \sim P_{data}} \left[ \log (1 - D(G(z, x_i), x_i)) \right]$$

In order to train VSUGAN more conveniently and stably, Pearson $\chi^2$ divergence is applied to replace KL divergence in classical GAN:

$$\min_D \max_G V(D) = E_{x \sim P_{data}}, x_i \sim P_{data} \left[ (D(x, x_i) - 1)^2 \right] + E_{z \sim P_z, x_i \sim P_{data}} \left[ (D(G(z, x_i), x_i))^2 \right]$$
$$\min_G V(G) = E_{z \sim P_z, x_i \sim P_{data}} \left[ (D(G(z, x_i), x_i) - 1)^2 \right]$$

The workflow of generator network G is shown in Fig. 2a, which has two encoders and one decoder. The original utterances are down-sampled to 16,384 Hz and sliced into a group of segments with a length of 4 s. The segment length of 4 s is chosen because it is difficult to extract enough style information from shorter speech. The input of the encoder is the 257 × 513 × 1 spectrogram obtained by STFT of voice segments. One encoder (encoder for noise, termed as n-Encoder) is employed to extract the content information (Info Code) from the spectrogram of a-1 audio, and the other encoder (encoder for style template, using s-Encoder for short) is exploited to extract style information (Style Code) from the spectrogram of a-2 audio. Subsequently, content and style information, i.e., Info Code and Style Code, are combined and fed to the decoder. Next, the decoder outputs the spectrogram with a unified style, and the spectrogram generates the target audio through Inverse Short-Time Fourier Transform (ISTFT). Griffin Lim algorithm is used for generate phase signals in ISTFT.

At the same time, Fig. 2b illustrates the discriminator workflow. The input of the discriminator is the combination of the generator output spectrogram and the style template spectrogram. The output is the judgment of whether their styles are consistent.

Generator configuration. The configuration of the generator is illustrated in Fig. 3. The n-Encoder and the s-Encoder which share the same structure in the generator network are designed to extract content and style information. They compress the input spectrogram of 257 × 513 × 1 into the encoding information of 2 × 3 × 2048 by eight encoder units, which are applied to downsample the image by a 3 × 3 convolution kernel with a stride of 2, and no pooling layer is utilized similar to DCGAN. After convolution, the data size changes from $(h, w)$ to $(\frac{1}{2}(h + 1), \frac{1}{2}(w + 1))$, and the convoluted data is activated by Rectified Linear Unit (ReLU). Similar to the structure of the encoder, the decoder decodes the 2 × 3 × 4096 encoded information into 257 × 513 × 1 spectrogram through eight decoder units, which also apply a 3 × 3 convolution kernel with a stride of 2 to upsample the image through fractionally strided convolutions. At the same time, skip connections are concatenating with the output result of the previous decoder and the input of the encoder. Thus, skip connections are exploited to reduce the loss information and solved gradient explosion and gradient disappearance in training.

Discriminator configuration. Figure 4 shows configuration of the discriminator. The discriminator concatenates two 257 × 513 spectrograms into a 257 × 513 × 2 tensor and generates 5 × 9 × 1024 feature maps through six convolution layers with a stride of 2. The convoluted results are normalized by batch normalization and activated by ReLU. Then, the convoluted feature map is fed to the 5-layer fully connected network, and finally a score between [0, 1] is calculated to judge the styles similarity of two input voice segments.

VSUGAN training. Data preparation. The data set used in VSUGAN is constructed by THCHS-30 corpus, which is an open-source Chinese speech corpus and contains 13,388 Mandarin sentences of 60 speakers. All the voice was recorded in a quiet office environment. In addition, the corpus includes three kinds of 0 dB...
noise: cafeteria noise (cafe), car noise (car), and Gaussian white noise (white)\textsuperscript{23}. Firstly, the whole corpus is read and resampled to 16,384 Hz. And then, the speech and noise are segmented into segments per 4 s. According to different speakers, all speech segments are divided into the training set and testing set. The training set contains all the speech fragments of 51 randomly selected speakers, and the testing set includes the speech pieces of the rest nine speakers.

Let the set of all speakers as $A$, and the speech fragments set of one speaker $a$ as $Ca$. During the training, the spectrogram of the sample $ca$ from $Ca$ is processed via image morphology algorithm to obtain $\sim ca$ by changing the voiceprint. Here, $\sim ca$ is used to destroy the original voice details and simulate the voiceprint changes caused by different pickups or environments. Meanwhile, the image morphology algorithm applied to process the spectrogram of the sample $ca$ is randomly selected from the eight algorithms in Table 2.

Besides, a sample noise $n$ from noise set $N$ are mixed proportionally with $\sim ca$ to generate $za$ with the proportion $r = \text{random}(0, 0.3)$:

$$za = \sim ca + (r \times n)$$  \hspace{1cm} (4)

where the set $Za$ including all $za$ is the mixed noise audio set of the speaker $a$.

In addition to $ca$, another sample $xs$ from $Ca$ is considered as a style template. Since the recorded environment and pickup of $ca$ and $xs$ are the same with speaker $a$, it can conclude the assumption that the styles of $ca$ and $xs$ are the same. Therefore, $ca$ is used as a label to evaluate whether the generator output audio style is similar to $xs$ in VSUGAN. It can transform the “judgment of style consistency” from an unsupervised learning problem to a supervised learning problem make. The present design is very important for the discriminators' training.

**Loss function.** The loss function consists of two parts: one is the L1 loss judge the loss degree of information by calculating the output $G(z_a, x_s)$ of the generator and $ca$, the other is the discriminator loss to judge the loss degree of style.

Figure 2. System workflows. (a) Generator workflow, where $a_1$ is noise-mixed audio, $a_2$ is style template audio, and $a_3$ is target style audio. (b) Discriminator workflow, where $b_1$ is the spectrogram of style template audio, and $b_2$ is the spectrogram generated from the generator workflow.
L1 loss. Given the training data of the mixed noise audio $z_a$, the clean audio $c_a$, and the style template audio $x_s$, the L1 loss is defined as:

$$\text{L1 loss} = \sum |z_a - c_a|$$

Figure 3. Generator configuration. (a) Generator structure. (b) Encoder unit. (c) Decoder unit.

Figure 4. Discriminator configuration.
where \( n \) is the number of elements in the matrix of the input spectrogram \( c_a \) or the output spectrogram \( G(z_a, x_s) \) (the sizes of the two matrices are equal).

Discriminator loss. Given the training data of the mixed noise audio \( z_a \), the clean audio \( c_a \), and the style template audio \( x_s \), the discriminator loss is denoted as:

\[
LD = (D(c_a, x_s) - 1)^2 + (D(G(z_a, x_s), x_s))^2
\]  

When training the generator, the output of discriminator \( D \) is considered as a part of the generator loss to measure the loss of style:

\[
L_{GD} = (D(G(z_a, x_s), x_s) - 1)^2
\]  

Total loss. Combining the L1 loss and the discriminator loss mentioned above, the total loss of the generator is:

\[
L_G = L_{GD} + (K \times L_{GL1})
\]  

where \( K \) is termed as the hyper-parameter to control the weights of the two losses. Initially, \( K \) was set to 100. But in VSUGAN, it can be observed that the \( L_{GD} \) was one order of magnitude lower than the \( L_{GL1} \). While \( K \) is set to 10, the two parts of the total loss achieve the best balance.

Training. Adam optimizer is used in both generator’s training and discriminator’s training in VSUGAN. The learning rate of generator is set to 0.001, and the value of discriminator is set to 0.0001. As a default, beta1 and beta2 are set to 0.9 and 0.999, respectively. VSUGAN is trained for 44 epochs using a batch size of 5. If the training is continued more than 44 epochs, the model will result in overfitting.

Experimental results

Evaluation metrics. Signal noise ratio (SNR) is an indicator of the amount of noise in the measured audio. The higher the SNR represent the lower amount of noise. SNR calculates the ratio of the output signal power to the output noise power. The expression of SNR is formulated as:

\[
SNR(dB) = 10 \log_{10} \left( \frac{|signal|_2}{|noise|_2} \right)
\]  

For a group of training or testing data, the mixed noise audio \( z_a \), clean audio \( c_a \), style template audio \( x_s \), and the generator output \( G(z_a, x_s) \) are used as signal. The difference between the audio to be measured and \( c_a \) is used as noise, the formulation changes to:

\[
SNR_{out} = 10 \log_{10} \left( \frac{|c_a|_2}{|out - c_a|_2} \right)
\]

\[
SNR_{za} = 10 \log_{10} \left( \frac{|c_a|_2}{|z_a - c_a|_2} \right)
\]

\[
\Delta SNR = SNR_{out} - SNR_{za}
\]

\( SNR_{out} \) is the SNR of the generator output, and \( SNR_{za} \) is the SNR of \( z_a \). \( \Delta SNR \) is the different between \( SNR_{out} \) and \( SNR_{za} \). Style unification is more effective when \( \Delta SNR \) is higher.

Mel Cepstral Distortion (MCD) is usually utilized in voice conversion tasks to calculate the similarity between target voice and converted voice. Using Mel-Cepstral coefficients (MECPE)\(^{[3]}\), MCD calculates the Euclidean distance\(^{[3]}\) between the Mel-cepstrum of two voice signals. While the MCD value is lower, the similarity between
the two voices is higher. In the testing of VSUGAN, the original clean audio $c_a$ was used as a reference to calculate the MCD of mixed noise audio $z_a$ and generator output $G(z_a, x_s)$. In the implementation, pyworld library, pysptk library, and librosa library were used to analyze MECP and MCD\(26\). In the present work, MCD of the generator output is named $MCD_{\text{out}}$ and MCD of the $z_a$ is named $MCD_{z_a}$. $\Delta MCD$ equals $MCD_{z_a}$ minus $MCD_{\text{out}}$. The $\Delta MCD$ is positively correlated with model performance.

In order to further evaluate the performance of the VSUGAN, two additional indicators (PESQ\(27\) and STOI\(28\)) were chosen. And Pypesq and Pystoi libraries are used for analysis of PESQ and STOI.

**Training Results.** Figure 5 shows the effect of a piece of audio processed after the network. Figure 5a is the audio input of the network with the waveform (a1), spectrogram (a2), and partial enlargement (a3). The audio is a sample from the testing set with destroyed voiceprint details on the spectrogram, mixed with 30% Gaussian white noise. Figure 5b illustrates the audio output from the generator, where another clean audio of the same speaker is used as the style template to perform style consistency calculation on the input audio of Fig. 5a. And the original audio of Fig. 5a is shown in Fig. 5c. Through comparing (a2) and (b2) in Fig. 5, it can be seen that the background noise is basically filtered out by the VSUGAN. Besides, it is clear that the damaged voiceprint details in (a3) are recovered to a certain extent in (b3), by comparing (a3), (b3), and (c3) in Fig. 5.

The statistics of $\Delta SNR$ and $\Delta MCD$ of testing set audio with different background noise is calculated in VSUGAN (Fig. 6). In the testing set, the input of (a) and (b) in Fig. 6 is nine sample audio from nine speakers whose original voiceprint details are destroyed by the image morphology algorithm on the spectrogram and then mixed with noise, i.e., cafe noise, car noise, and white Gaussian noise. The mixed noise intensity is 0–99% of 0 dB with increasing 1% every intensity. Therefore, $\Delta SNR$ and $\Delta MCD$ of nine input and output audio are obtained. With the larger $\Delta SNR$ and $\Delta MCD$, the output style is more consistent. From Fig. 6a, b, we can find that the values of $\Delta SNR$ and $\Delta MCD$ have significant correlation with noise intensity. When the noise intensity exceeds 30%, $\Delta SNR$ value is about 4–10 dB, where Gaussian white noise is significantly greater than other two kinds of noise. On the contrary, the $\Delta MCD$ with Gaussian white noise is lower while the noise intensity exceeds 30%. Due to the repaired voiceprint in VSUGAN, SNR and MCD of output audio are better than that of input audio, even if the mixed noise intensity is 0%.

In Figs. 6c, d, all original audio in the testing set which is mixed with three kinds of noise in the fixed intensity after destroying the voiceprint details are used as the network’s input audio. As shown in Figs. 6c, $\Delta SNR$ of the audio mixed with Gaussian white noise is significantly higher than mixed with other two kinds of noise. However, $\Delta MCD$ value of the audio mixed with Gaussian white noise is lowest in three kinds if noise (Fig. 6d). Thus, it can be seen that VSUGAN has different degrees of improvement for types of noise.

Along with the testing set of THCHS-30, VCTK-Corpus\(29\) and NoiseX-92 dataset were applied to validate the performance of VSUGAN. NoiseX-92 which have 15 kinds of noise (including white noise, pink noise, and vehicle interior noise) is part of the Signal Processing Information Base\(30\). In the performance test, ten audio samples of nine speakers were randomly sampled from two datasets (90 audio samples per dataset). These audio samples were also mixed with the noise of random intensity after the voiceprint details were destroyed. As shown in Table 3, the noise in group1 and group2 has the same source as the noise mixed in during training. The noise in group3 was selected randomly from the NoiseX-92 dataset and has not been trained. Group4 has the same corpus as group1. In order to simulate the real acoustic environment, rir_generator\(31\) is used in group4 to generate reverberation without the image morphology algorithm. This mixed-noise audio was fed into VSUGAN,
and the average values of PESQ, STOI, SNR and MCD from output audio are calculated. The better audio quality is reflected in higher value of PESQ/STOI/SNR and lower value of MCD. It is worth mentioning that only THCHS-30 is utilized during VSUGAN training. SEGAN is used as a baseline for comparison, and the training of SEGAN use the same data set and method from Pascual et al.

The experimental results is shown in Table 3. Compared with input noisy, the indicators of SNR and MCD have significantly improved in all four groups. And PESQ value has advanced in group1 and group2 but not in

| Group1 | THCHS30 | SNR: 4.24 MCD: 6.88 PESQ: 1.95 STOI: 0.76 |
| Group2 | VCTK | SNR: 0.6 MCD: 4.77 PESQ: 1.4 STOI: 0.54 |
| Group3 | VCTK_Noise92 | SNR: 0.03 MCD: 4.19 PESQ: 2.13 STOI: 0.73 |
| Group4 | THCHS30_Reverb | SNR: 0.03 MCD: 4.19 PESQ: 2.13 STOI: 0.73 |

Table 3. Audio processing effect statistics in different corpora.

Figure 6. The ΔSNR and ΔMCD statistics of the testing set. (a) ΔSNR with 0–100% background noise intensity. (b) ΔMCD with 0–100% background noise intensity. (c) ΔSNR with 30% background noise intensity. (d) ΔMCD with 30% background noise intensity.
group 3 and group 4. The STOI value has few difference between group 1–group 3, while is increased slightly in group 4. Thus, the current VSUGAN has stable performance in different data sets.

For the reproducible research, our source code was uploaded to Github (https://github.com/oy-tj/VSUGAN). The data set and trained model are shared on an online disk (https://pan.baidu.com/s/1RwpwZjSZEl7hrLp9vNizA password: 4y8a).

Conclusion

In the present study, we proposed and implemented a model based on GAN by combining the style information extracted from the style template audio and the voice information extracted from the audio. VSUGAN without training for extra speakers generates the audio as same style as the template. VSUGAN is trained with THCHS-30 corpus and tested on two open-source corpora. The experimental results demonstrated that VSUGAN can effectively improve the quality of the recorded audio and reduce the style differences in kinds of environments.

Acknowledgements

This work is supported by XMU Training Program of Innovation and Enterpreneurship for Undergraduates (2019103843433). We appreciate the linguistic assistance provided by TopEdit (www.topeditsci.com) during the preparation of this manuscript.
Author contributions
T.O. designed the project, coordinated the work and write the initial draft. Z.Y. and H.X. assisted this experiment. T.H. and Q.L. reviewed or revised draft articles. All authors have read and agreed to the published version of the manuscript.

Funding
XMU Training Program of Innovation and Entrepreneurship for Undergraduates (Grant No. S201910384343).

Competing interests
The authors declare no competing interests.

Additional information
Correspondence and requests for materials should be addressed to T.O.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher’s note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

© The Author(s) 2021