Deep Learning for Audio Style Transfer

Zhixian Huang
Department of ECE
zhuang31@ur.rochester.edu

Shaotian Chen
Department of ECE
schen121@ur.rochester.edu

Bingjing Zhu
Department of CS
bingjing.zhu@ur.rochester.edu

ABSTRACT

Style transfer, the technique of recomposing one input using the style of other inputs, has increasing popularity recently. Using the power of convolutional neural network, Gatys [1] has achieved great success in generating images of specific artistic style. The success in image style transfer inspires people to use similar methods to do style transfer in audio domain. In this project, we implement an architecture in [2]. We use this architecture on several music pieces and find the performance of speech style transfer is surprisingly and relatively the best, so we do a further experiment on VCTK dataset [3] to prove its effectiveness to extract speaker characteristics with several seconds, which can be used to change speech's style among different speakers.

1. INTRODUCTION AND RELATED WORK

The main problem we want to explore in this project is how to utilize texture network to describe artistic "style" and "content" of audio signal. Style and content are subjective perception of human beings. If we want to achieve artistic style transfer from one image or audio example to another, the key is to capture style and content features accordingly and properly.

In image domain, a successful style transfer algorithm will generate a new image with matching content information (what is in the image), as well as matching stylistic information (artistic manner of expression). In other words, it answers the question "What would a rendering of scene A by artist B look like?" [4] Figure 1 is an example of image style transfer. It is a transfer from a normal photo of a dog to Van Gogh style painting of the dog.

This can be achieved by using a deep CNN (Convolutional Neural Network) model. In [1], the authors use VGG [5] to extract the features of images. This VGG is pre-trained so it can provide proper encoding. The architecture is shown in Figure 2. They take feature maps in some layers in VGG as "content" of an image. To capture the style of the image, they use the correlation between the feature maps in different channels, but same layer can take over the spatial extent of the input image. These feature correlations are also called Gram Matrix. They optimize a white noise into target image by back propagating sum of "content loss", which is the "content" distance between content image and target image, and "style loss", which is the "style" distance between style image and target image. In their experiments, they tune hyper-parameters, for example, the layers where output content loss and style loss.

The performance of output in image style transfer is good in perception. So, can we use similar method in audio style transfer? In fact, an audio signal can also be regarded as an image signal. Audio signal in time domain is always hard to process, so we often do a STFT (Short Time Fourier Transform) to it. Magnitude spectrogram can be regarded as an image.

In [2], they use same method of Gatys [1], taking magnitude spectrogram as input instead of image. In this project, we use magnitude spectrogram of audio as input too.

Figure 1. Image style transfer example. The left one is a picture of a dog, and it is taken as a "content" image in this example. The middle one is a painting of Van Gogh, and it is taken as a "style" image in this example. The right one is a synthesis image by the first two images.[6]

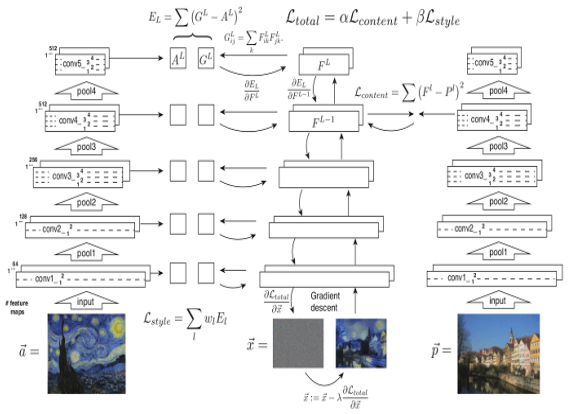

Figure 2. Architecture of image style transfer [1]

For audio signal, we can define "content" as notes for music, texts for speech. But "style" may have ambiguous meanings, especially for music. For one music piece, you can play it in "jazz style" and "pop style", in this case, "style" is defined by music genre.; you can play it with piano only or violin only, in this case, "style" is defined by instruments. Sometimes, one song is interpreted by different performers, and in this case, "style" is defined by dynamic and velocity. While for speeches, style can be defined by speakers generally. But sometimes emotion of a speech can also be regarded as style of the speech. In this project, we use several pairs of audio samples to test our architecture. For quantitative evaluation, we design a experiment and use VCTK [3] dataset to prove the effectiveness of this architecture.

2. METHOD DESCRIPTION

The figure of an overview of the architecture shown in Figure 3. In this section, we explain why we choose random CNN to do feature representation and describe details in this method.

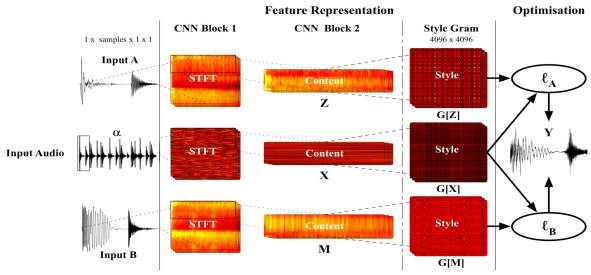

Figure 3. Architecture of this method. Input A is content audio, input B is style audio and we initialize the output with input A. To optimize the output, we use both content loss and style loss. (Adapted from [2])

2.1. Network Selection

In image style transfer, VGG is used to extract feature. VGG is a well-known deep neural network and has good performance in image feature extraction. SoundNet is a CNN designed on a large number of unlabeled videos, and it has state-of-art performance in sound feature extraction. However, the two neural networks are shown to have bad performance in this task [7]. Recent works showed that Wide-Shallow-Random Network can be used to extract style statistics [8,9,10]. So, in our project, we use single-layer CNN with 4096 random filters to extract style feature.

2.2. Pre – processing(STFT)

Compared to raw audio sample in time domain, audio signal in frequency domain have more spatial features. We use Short-Time Fourier Transform (STFT) to convert the raw audio into spectrogram. For each input audio, we use a Hanning window of n samples (2048) with a n/2 hop size to segment it into T frames. As shown in Figure 3, STFT is represented by CNN Block1. Regarded as an image, a spectrogram has T channels and n samples for every channel.[2]

2.3. Feature Representation (Defining Loss Functions)

We use random CNN to extract content feature, and use Gram matrix of output of random CNN to describe style. The result will be gradually optimized by reducing content loss and style loss.

The content loss is the mean squared error between the embedding of white noise image and the content image. For a layer l and the input image $\vec{x}$, let the number of filters be N. The embedding will have N feature maps, each of size M, where M is the height times width. So, the encoded image of layer can be stored in a matrix $F_l \in R^{N|xM|}$, as in the equation (1)

$$L_{content}(\vec{p},\vec{x},l) = \frac{1}{2}\sum_{i,j}(F_{ij}^{l} - P_{ij}^{l}) \quad (1)\ [11]$$

In our architecture, we directly optimize content signal instead of white noise. By optimizing content signal instead of white noise, we want to get a more stable and clear output and a faster optimization. For capturing the style of an artist, a style representation is used. It computes the correlations between the different filter responses, where the expectation is taken over the spatial extent of the input image. These feature correlations are given by Gram Matrix$G^l \in R^{N|xN|}$, where G_{ij}^{l} is

$$G_{ij}^{l} = \sum_{k} F_{ik}^{l} F_{jk}^{l} \quad (2)\ [11]$$

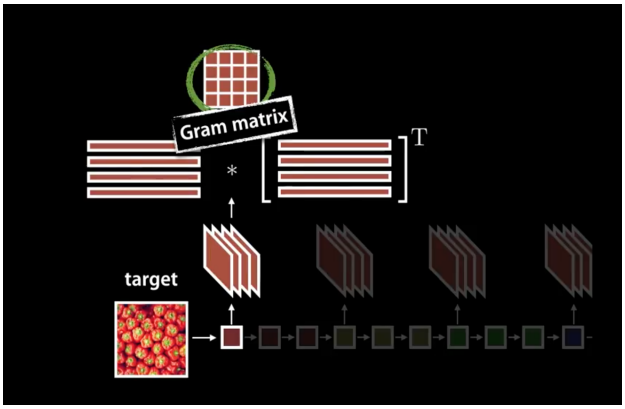

Figure 4. Gram matrix in the texture network. In the highlight part, first we feed the target image into the CNN, then it calculates inner product of every two embeddings in this layer, finally add these multiplications together to get Gram Matrix of this layer. [12]

The style loss is the mean squared error between the Gram Matrix of style image and the white noise image. Let $\vec{a}$ be the style image and $\vec{x}$ be the white noise image. Let A and X be the style representations of style image and white noise image in layer l. So, total style loss of a layer l is E.

$$E_l = \frac{\sum_{i,j}(X_{ij}^l - A_{ij}^l)^2}{4N_l^2 M_l^2} \quad (3)\ [11]$$

The total style loss is

$$L_{style}(\vec{a},\vec{x}) = \sum_{l=0}^{L} w_l E_l \quad (4)\ [11]$$

Where w is the weighting factor of each layer.

In representing style, we use similar style representation. For style extraction, gram matrices are used as they are in image style transfer. Gram Matrix $G \in R^{NxN}$, where G is the inner product between the feature maps i and j represented by vectors, and N is the number of feature maps. The difference is that the feature maps here are 1- dimensional whereas in images they are 2D. As we are using a model with only one layer, there is no notation of layer. Let F be the spectrogram, i.e., the encoding of the audio of ith filter at jth position.

$$G_{ij} = \sum_{k} F_{ik} F_{jk} \quad (5)$$

Style loss is the mean squared error between the gram matrices of style audio and the generated audio, i.e., content audio. Let $\vec{a}$ be the style audio and $\vec{x}$ be the generated audio. Let A and X be the style representations of style audio and generated audio with N number of channels (or number of filters) and M number of samples. Total style loss is $L_{(\vec{a},\vec{x})_{style}}$

$$L_{(\vec{a},\vec{x})_{style}} = \frac{\sum_{i,j}(X_{ij} - A_{ij})^2}{4N^2M^2} \quad (6)$$

The total loss is sum of content loss and style loss.

2.4. PhaseReconstruction

Since we only use magnitude spectrogram as input, we cannot get audio output directly. The phase information is unknown. Here, we use Griffin-Lim algorithm [13] to reconstruct phase.

Algorithm 1 Griffin-Lim algorithm (GLA)

Fix the initial phase $\angle c_0$
Initialize $c_0 = s \cdot e^{i\angle c_0}$
Iterate for $n = 1, 2, \ldots$
$c_n = P_{\mathcal{C}_1}(P_{\mathcal{C}_2}(c_{n-1}))$
Until convergence
$x^* = \mathbf{G}^{\dagger} c_n$

This algorithm estimates phase information from magnitude information by an iterative algorithm. First initial a random phase and then use iterative projection to estimate the true spectrogram. In [14], an approximate way to perform the projection is proposed.

3. EXPERIMENT AND ANALYSIS

Since neural network in our model don't need to train, we can feed music pieces into it directly. Our experiment consists of 2 parts. The first is on audios of three genres and the second is on dataset VCTK [3].

3.1. Dataset

We first test the model in 6 pairs of audio samples. All of the audio files in the dataset are 10 seconds in length. The dataset consists of single-instrument music, symphonic music and speech. Symphonic music has abundant genres, single-instrument music has less and speech has the minimal. However, due to the small size of the dataset and the fact that we can only use spectrogram and human perception to evaluate the synthesis result, it is hard to give a quantitative measure, so we design another experiment to prove the effectiveness of the model. The dataset of the second experiment is CSTR VCTK Corpus, English Multi-speaker Corpus for CSTR Voice Cloning Toolkit, which includes speech data uttered by 109 native speakers of English with various accents. Each speaker reads out about 400 sentences, most of which were selected from a newspaper plus the Rainbow Passage and an elicitation paragraph intended to identify the speaker's accent [3].

3.2. Comparison of spectrograms of 3 typical pairs

In fact, neural style transfer has no specific aim genre to transfer, for example, voice or instruments. It is only an analogy of image style transfer and we use magnitude spectrogram as an image in this analogy. It is very difficult to use a standard evaluation method so we observe the magnitude spectrogram of content audio, style audio and synthesized audio and use human perception to evaluate the result.

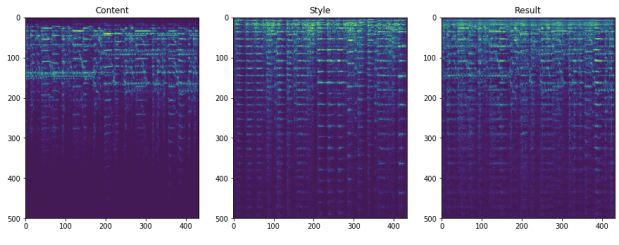

Figure 5. Content audio is a rap music, style audio is a speech. We can see the result audio's spectrogram looks more like content audio's spectrogram and embedded some patterns of style audio's spectrogram. When listening to the result, it sounds like the rapper's voice changes into the speaker's in style audio.

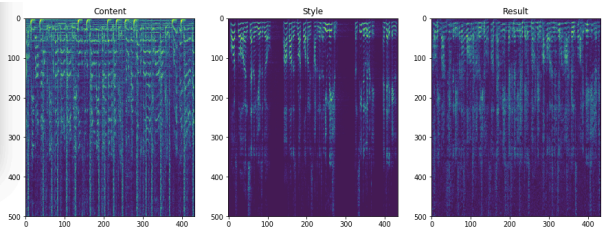

Figure 6. Content audio and style audio are both symphonic music. It is hard to say if the result's spectrogram looks like content's more, and by listening to the result audio, it sounds like the mixture of the 2 audios.

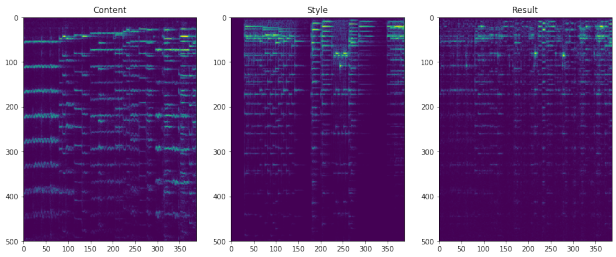

Figure 7. Content audio is violin and style audio are piano. The result spectrogram turns the harmonic of content spectrogram (violin) into the harmonic of style spectrogram (piano). By listening, we find the result audio basically keep the notes of content audio and changes the sound of violin into piano.

If we look at the three spectrograms, we can find that the result contains some statistic in content and style spectrogram. By listening to them, we have some interesting discovery. Though the result of two symphonic pieces sounds like a mixture of the two inputs, the result of rap music (content) and speech (style) sounds like transfer voice in rap music onto voice in speech music; And for violin and piano, the style transfer also works. Based on this discovery, we then test the model in speech audios. (We also want to or should test the model in some single instruments audios, but time is limited.)

3.3. Experiments on VCTK

In order to prove that the Gram Matrix of embeddings of random CNN can really work on style feature extraction, we do an experiment on VCTK. For every speaker, we split the sentences into two parts, one for classifier training, another for testing. The rate of training data and testing data is 1:1. The classifier aims to classify the incoming speech audio into correct speaker identity. It compares the distance between the input speech and all the training speech, and takes the closest one's identity as the classified identity for input speech. The distance is defined by the Gram Matrix of embeddings of random CNN, same with our model in style representation.

Speaker Number	Train/Test	Accuracy
4	80/80	0.925
10	80/80	0.485

Table 1. Test results of experiments on VCTK

Limited by our computer memory, we only test 40 utterances of 4 first speakers and 16 utterances of 10 speakers. The table shows the accuracy.

3.4. Analysis and discussion

The model is inspired by texture network, which is designed for image style transfer. We regard spectrogram as image in order to adapt the network in image style transfer. Instead of using pre-trained CNN (like VGG, SoundNet), we use Wide-Shallow-Random Network to extract features, and take the output of a convolutional layer as "content" while take the Gram Matrix of this layer as "style". Surprisingly, the random CNN has a good performance in speech recognition, which means it can capture features of speaker identity.

4. FUTURE WORK

Our model can achieve simple style transfer with relative low quality, but has limited ability to work on more complex music style transfer. On one hand, it is a very fast model which doesn't require pre-training, and it can also capture simple content and style information with especially good performance in speech style transfer. On the other hand, the architecture can hardly capture complex content and style feature, and to synthesize them correctly. Recently, generative model has become very popular. Google presents a special architecture to process time domain signal in latent space and generate music notes [15], which is called WaveNet. Maybe we can first use style loss of our project as an estimation of speaker identity, which serves as an additional input of WaveNet, and then use WaveNet to generate a high quality of speech in specific speaker style.

5. REFERENCES

1. Leon A Gatys, Alexander SEcker, and Matthias Bethge, "Image style transfer using convolutional neural networks," in Proceedings of the IEEE

Conference on Computer Vision and Pattern Recognition, 2016, pp. 2414–2423.

2. Tomczak, Maciek, Carl Southall, and Jason Hockman. "AUDIO STYLE TRANSFER WITH RHYTHMIC CONSTRAINTS."

3. https://datashare.is.ed.ac.uk/handle/10283/2651

4. Barry, Shaun, and Youngmoo Kim. ""Style" Transfer for Musical Audio Using Multiple Time-Frequency Representations." (2018).

5. Karen Simonyan and Andrew Zisserman, "Very deep convolutional networks for large-scale image recognition," arXiv preprint arXiv:1409.1556, 2014.

6. https://dmitryulyanov.github.io/feed-forward-neural-doodle/

7. Grinstein E, Duong N Q K, Ozerov A, et al. Audio style transfer[C]//2018 IEEE International Conference on

8. Ivan Ustyuzhaninov, Wieland Brendel, Leon A Gatys, and Matthias Bethge, "Texture synthesis using shallow convolutional networks with random filters," arXiv preprint arXiv:1606.00021, 2016.

9. Gilles Puy, Srdan Kitic, and Patrick Pe´rez, "Unifying local and non-local signal processing with graph CNNs," arXiv preprint arXiv:1702.07759, 2017.

10. Kun He, Yan Wang, and John Hopcroft, "A powerful generative model using random weights for the deep image representation," in Advances in Neural Information Processing Systems, 2016, pp. 631–639.

11. Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "A neural algorithm of artistic style." arXiv preprint arXiv:1508.06576 (2015).

12. https://www.youtube.com/watch?v=f7EejzFOKqw

13. D. Griffin and J. Lim, "Signal estimation from modified short-time Fourier transform," Acoustics, Speech and Signal Processing, IEEE Transactions on, vol. 32, no. 2, pp. 236 243, 1984.

14. J. Le Roux, H. Kameoka, N. Ono, and S. Sagayama, "Fast signal reconstruction from magnitude stft spectrogram based on spectrogram consistency," in Proc. 13th International Conference on Digital Audio Effects (DAFx-10), 2010, pp.397–403.

15.Oord A, Dieleman S, Zen H, et al. Wavenet: A generative model for raw audio[J]. arXiv preprint arXiv:1609.03499, 2016.