Timbre Style Transfer

Group E - Scarlett Hwang, Bingying Liu, Joaquin Menendez, Nathan Scheperle,

Muxin Diao

A. Abstract

Image style transfer using convolutional neural networks is a popular example to demonstrate the capabilities of machine learning algorithms, with the goal of taking the style of one image and applying it to the content of another. Our focus in this project is to extend this approach to audio generation. In the domain of music, several types of style transfer have been proposed, including composition style, performance style, and timbre style transfer. Here, we will focus on timbre style transfer by extracting the spectrogram from two inputs and performing image style transfer methods using convolutional neural networks.

B. Introduction

Timbre is the perceived quality of a musical note, distinct from its pitch and intensity, which is often distinguishable from its source instrument. Our main focus of this project is to transfer these perceived qualities from one audio input played on an instrument to the melody played on another instrument while preserving the other characteristics of its content. For example, a producer or composer might want to achieve the aural qualities of a violin, but only has sampled inputs of a melody played on a flute.

One function of our work is to recognize the timbre of an audio input. This is not only a useful tool for analyzing different roles of band instrumentations but also a validation method for the output of the timbre style transfer on how close the sample output sounds to the instrument we appointed it to be. In order to validate our style transfer results, we trained a random forest classifier to measure the performance of timber-transferred audio.

Our work has immediate applications in both music production and sound engineering. While it would obviously be a useful tool for composer and producers, it would also potentially enable higher fidelity sampling when designing music production software and extensions. Additionally, disentanglement between content and style is important for more many existing challenges, including transcription, voice recognition, and other types of style transfer mentioned in the abstract.

C. Background

Several previous and ongoing research projects have made progress on this topic, many focusing on voice synthesis and transfer, interpolation between timbres, and generation techniques.

WaveNet, “a deep neural network for generating raw audio waveforms”, was introduced by Van Den Oord et. al in 2016[1]. When appropriately trained, It is capable of realistically mimicking both speech and musical sounds using a

probabilistic model based on raw audio input.

Engel et. al (2017)[2] introduced NSynth and its associated dataset, a high-quality dataset of musical notes that has a larger scale than other comparable public datasets. They also proposed a WaveNet-style autoencoder that learns temporal hidden codes to effectively capture longer term structure without external conditioning with improved qualitative and quantitative performance. This allows for morphing between instruments, meaningfully interpolating in timbre to create new types of sounds.

Verma and O.Smith(2018)[3] tried timbral transfer from singing voice to music instruments, developing new techniques parallel to the image style transfer done by Gatys et al. (2015)[4]. They proposed a new way to "cross-synthesize imposing the style of one instrument on the content of another".

Perez, Proctor, and Jain(2017)[5] presented an extension of style transfer to speech and gained success with low-level textural features, but not with high-level prosody. Their prosodic features were not transferred very well and listeners described that the result as a continuous buzz. Many strategies were applied to improve the model, such as optimizing log-values of the frequency spectrogram and tuning weight parameters between losses to balance content and style. However it was unable to achieve stable gradient descent using all the layers needed, due to the optimization difficulties.

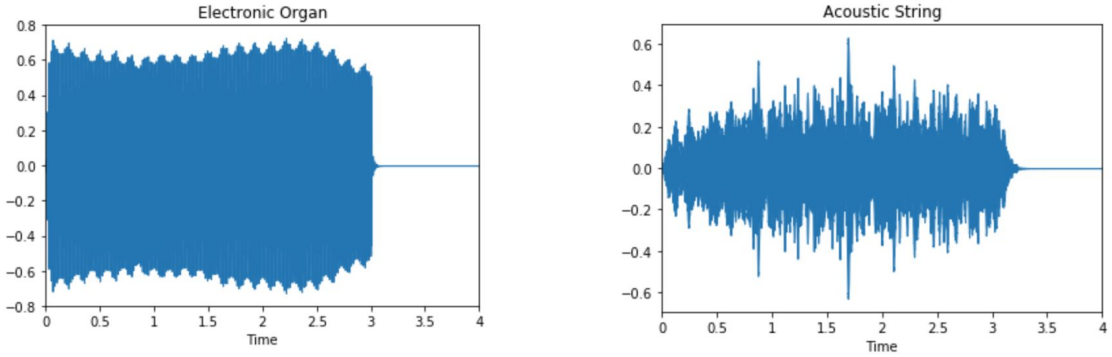
D. Data and Feature Extraction

Our training data comes from NSynth[6]. The NSynth dataset is analogous in audio analysis to the MNIST dataset in image recognition. It is part of an ongoing experiment by Magenta, which contains over 300,000 musical notes with unique pitch, timbre and envelope. Each audio is 4 seconds in length.

For training our VGG-based neural network, we use in total 15,000 labeled audio inputs from three distinct instrument families: electronic organ, acoustic string and acoustic flute. For training classifier, we use all instrument families from all sources: acoustic, electronic and synthetic. For our baseline CNN model, we use random weights to initialize the model and directly input content and style of audios.

In the training/classifier building process, we mainly utilized two ways to extract features from the original soundwaves: the spectrogram and feature vectors obtained from Mel-frequency Cepstral Coefficient (MFCC). A spectrogram is a graphical representation of the intensity of a spectrum frequency changes throughout time. The MFCC is a small set of features that concisely describe the overall shape of a spectral envelope and it scales the frequency to match more closely to what the human ear can hear.

Wave plots:



Spectrograms:

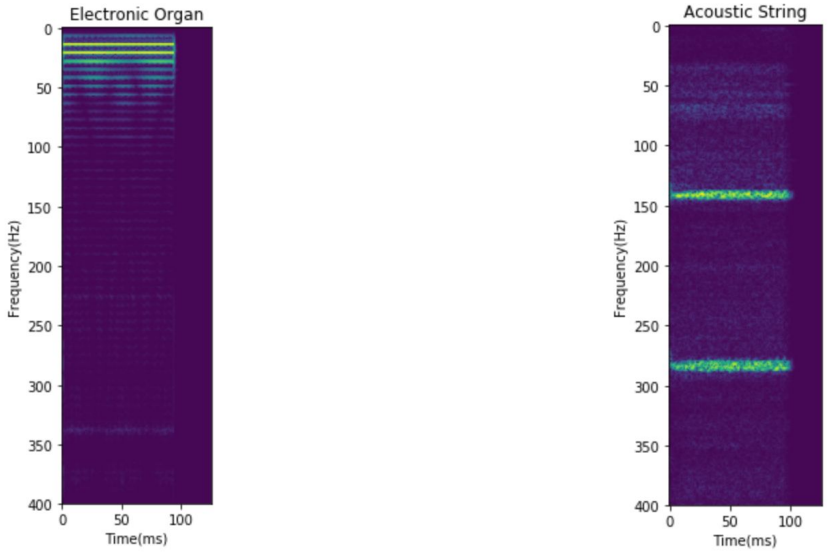


Figure 1. Wave plots and spectrograms

E. Methodology

i) Preprocessing

All NSynth input audio was read using its native sampling rate of 16,000 Hz, and the Short-Time Fourier Transform was done with an FFT-size of 512 and a hop length of 256. After performing STFT to extract the spectrogram from a given audio input, we perform log scaling of its power spectrogram, which is defined as its squared magnitude, to convert from a power spectral density to decibel units. This was done to obtain images with a more linear range of intensities, ideally allowing our image style transfer methods to make better use of the variation in frequency intensity. The resulting matrix is of size 1025 × 126 for a single audio input, where 1,025 represents frequency bin, and 126 represents the time frame. The value of the matrix entries represent the intensity of one frequency bin at a certain time frame.

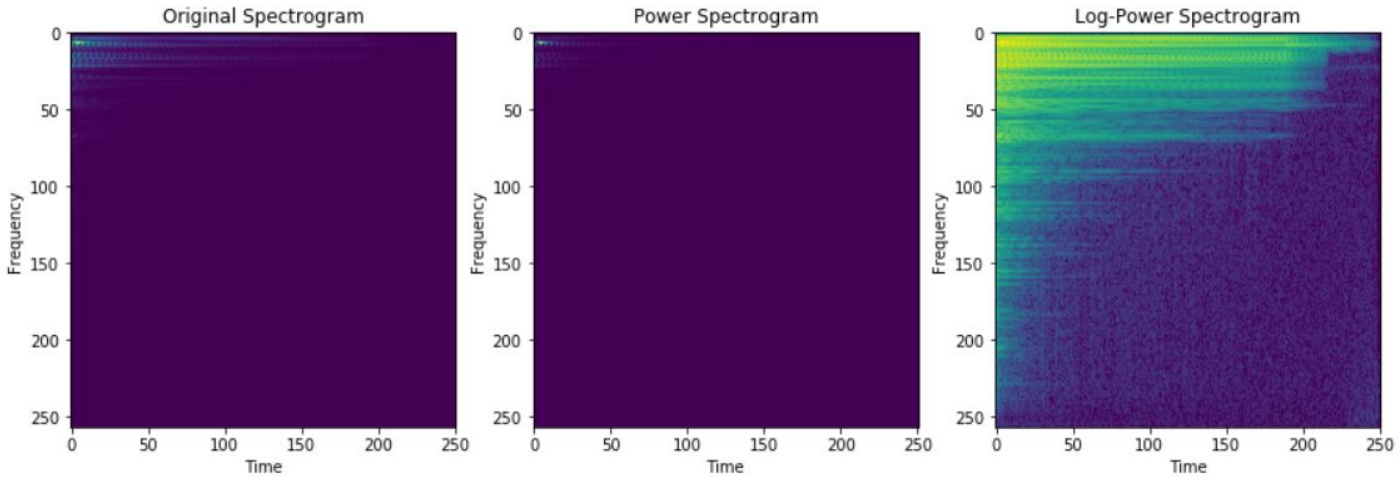


Figure 2. The original spectrogram for an acoustic keyboard, its respective power spectrogram (squared magnitude), and its respective log-power spectrogram (log(squared magnitude))

For MFCC feature extraction of each audio input, we calculate the first 13 mel-frequency cepstral coefficients of the audio file, their first- and second-order derivatives, and concatenate them into a single 39-element feature vector. The feature vector is also standardized so that each feature has zero mean and unit variance.

ii) Style Transfer Models

As a baseline model for style transfer, we constructed a neural network consisting of a single convolutional layer with 4,096 output filter channels and ReLU activation. The kernel of this layer was randomly generated weights. We define the style features as the evaluation of the style input spectrogram passed through this network, and the content features similarly. After computing the features, our goal is to find the spectral representation that minimizes the total sum of the style loss and content loss using gradient descent, defined as the difference between the representation's Gram matrix and the pre-computed style image features Gram matrix, and the difference between the representation's evaluation and the content spectrogram evaluation.

$$X_{transfer} = argmin_X L_{total} = argmin_X \alpha L_{style}(X, X_{style}) + \beta L_{content}(X, X_{content})$$

where

$$L_{style}(X', X_s) := \ell_2(\hat{y}_s^T \hat{y}_s, \hat{y}'^T \hat{y}')$$

and

$$L_{content}(X', X_c) := \ell_2(\hat{y}_s, \hat{y}')$$

The weights were assigned as $\alpha = 2$ and $\beta = .01$, and the optimization method used was L-BFGS.

As a second model, we used existing image style transfer frameworks that make use of the pre-trained image recognition network VGG19. In this method, specific layers have been found to correspond well to the content and style of an image [5], specifically 'conv4_2' for content, and for style ('conv1_1', 'conv2_1', 'conv3_1',

‘conv4_1’ , ‘conv5_1’). In the case of style, correlations between features for each layer in the subset of network layers are computed using the Gramian matrix.

Computing the transfer image using this model was performed as above, minimizing the total loss of style and content.

Finally, we attempted to build a custom model for style transfer, training a CNN to recognize the class of instrument of input audio using the spectrogram. To simplify the problem, we limited the classes of instruments to three: acoustic flute, electronic organ, and acoustic string. The model architecture was similar to VGG, consisting of 5 layers made up of 3x3 convolutional layers with 2x2 pooling. After training the model, we performed style transfer on the spectrogram as above. This model was trained on 15,000 training inputs (5,000 from each class) with a validation set of size 2,570 (1,599 organ, 156 flute, 815 string).

iii) Audio reconstruction

In each case, after optimizing the transfer representation, we perform the Griffin-Lim algorithm to reconstruct audio from the transfer spectrogram. This algorithm is an iterative process in which we first initialize the angle of the waveform phase randomly, then iteratively:

1. Estimate the output audio as the inverse STFT of the input spectrogram times *exp*(imaginary part of the phase angle)
2. Recompute the phase angle as the STFT of the estimate

Additionally, prior to performing this algorithm we convert our log-scaled spectrogram back to its original proportional scale. This process leads to some loss of signal and amplitude, but when run for a sufficient number of iterations, around 1,000 in our case, it produces a suitable reconstruction.

iv) Style classifier

In order to quantify an objective measure of the ‘accuracy’ of the style transfer, we developed two classifiers extracting of each audio the MFCC as features in a 39 element vector. The feature vector was standardized so that each feature has zero mean and unit variance. The main idea is that if an independent classifier classifies the output of our neural network as highly probable to be the instrument that we applied the style then the style transfer was successful.

The first classifier was developed in order to classify between the three possible outputs [‘organ electronic’, ‘flute acoustic’, ‘string acoustic’] of the second Neural Network. Every category was composed of 500 samples, where 375 used to train using an XG-Boosting method [10], meanwhile, the remaining 125 were employed as test samples. This model showed an accuracy classification training = 1.00 and an accuracy classification testing = 0.910.

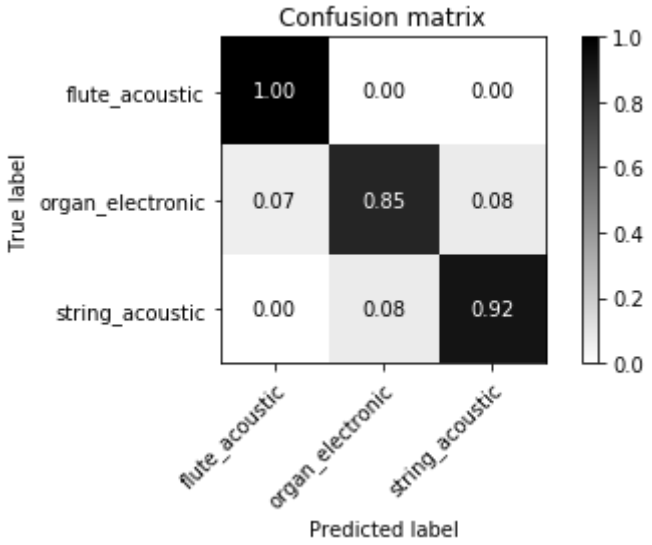


Figure 3. Confusion matrix for XG - Boosting (3 classes)

We decided to train a second classifier employing a bigger number of classes in order to have a more accurate measure of the style quality . To do so, we selected from the 30 possible classes, the ones that had at least 500 occurrences within Nsynth data set. After filtering we ended with 21 different classes [see figure 4 for more detail]. We employed a similar procedure as in the former classifier. We used 10500 audios files, divided into 375 training samples and 175 test samples for each category. This model showed an accuracy classification training = 0.721 and an accuracy classification testing = 0.434

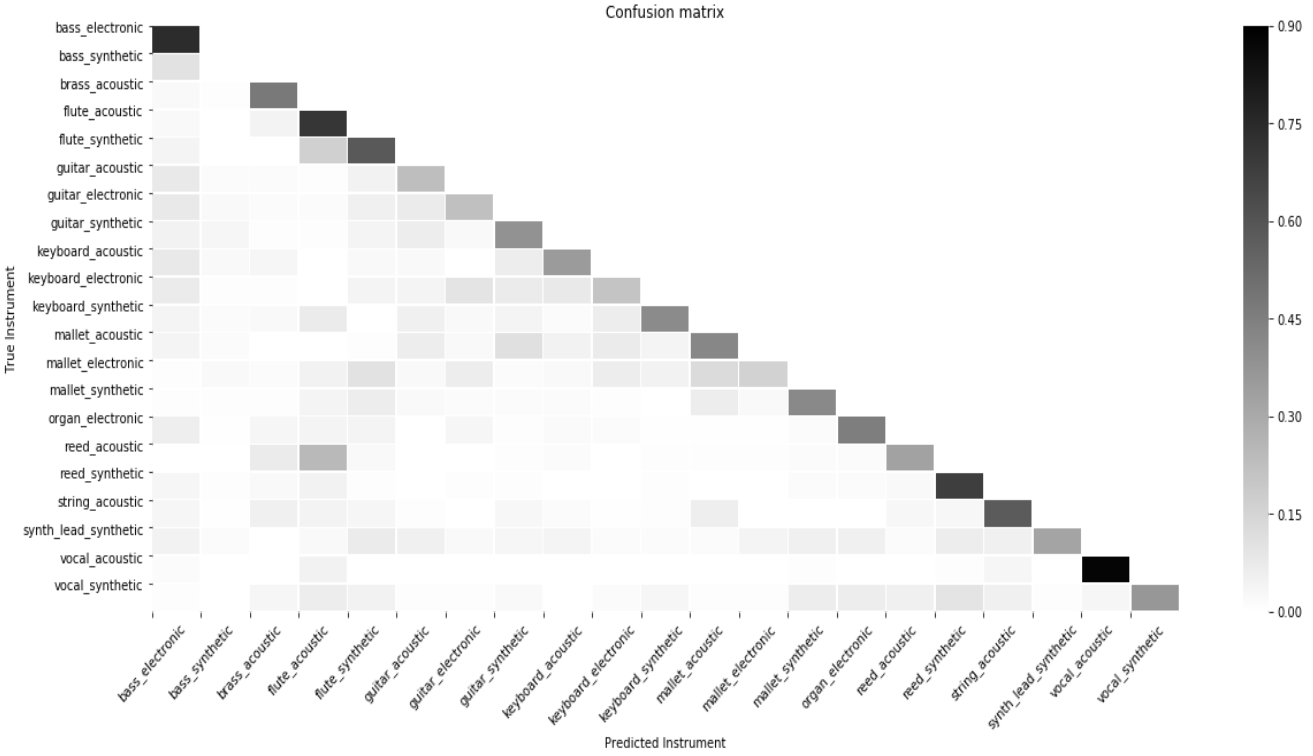


Figure 4. Confusion matrix for XG - Boosting (21 classes)

F. Results

In this section, we analyze one output from our model. In this case, we applied an Organ Electronic style[1] to a Flute Acoustic content[2] in order to get a timbre style transfer preserving the content [3][see figure 5].

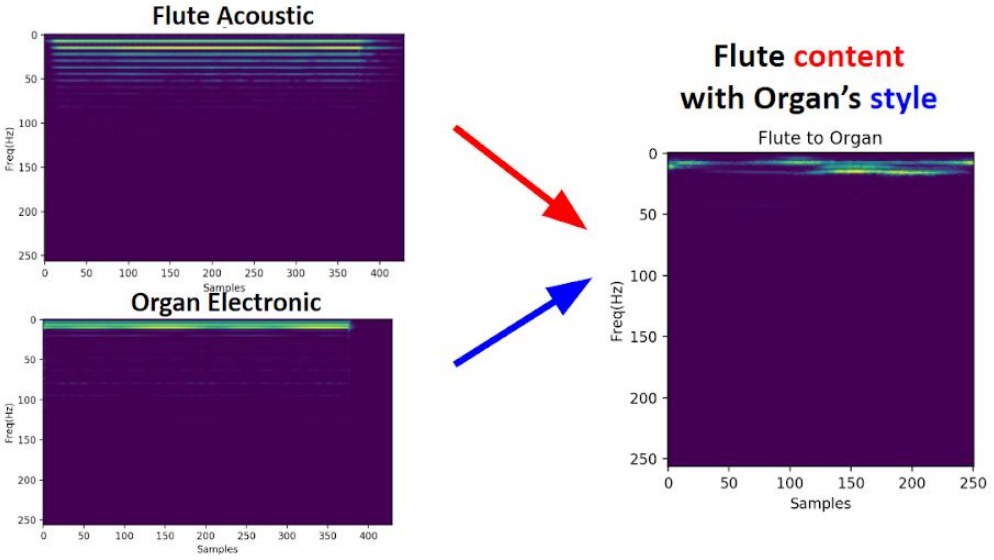


Figure 5. Example of one case of style transfer using VGG. Spectrograms of the original sounds are shown in the left. The output of the neural network is on the right.

We reconstructed the the transferred sound from the and we classified using the classifier mentioned above. We observed a likelihood of being an 'organ electronic' of 0.99 when we employed the three classes classifier. We observed a likelihood of 0.289 of being an 'organ electronic' when we employed the 21 classes classifier. We should highlight that in the case of the second classifier 'organ electronic' was the class with the biggest likelihood [see table 1].

We found that qualitatively the baseline model produced the most satisfying results when supplied with more complex inputs (e.g. multi-layered sound clips as opposed to single notes), while the VGG model produced better results when given these single note inputs, as shown above. Our custom model was less useful as trained and tuned.

[1] https://drive.google.com/file/d/1uQdzxy7GEfhRPLsiOVp8KpbJVxyRGOsm/view
[2] https://drive.google.com/file/d/1mGLLxbNoYR_FqCXAnyc8RVIDLkobsv8U/view
[3] https://drive.google.com/file/d/1Q6db3DaKbyKrG73QhQp4hoDt1b7FRb2i/view

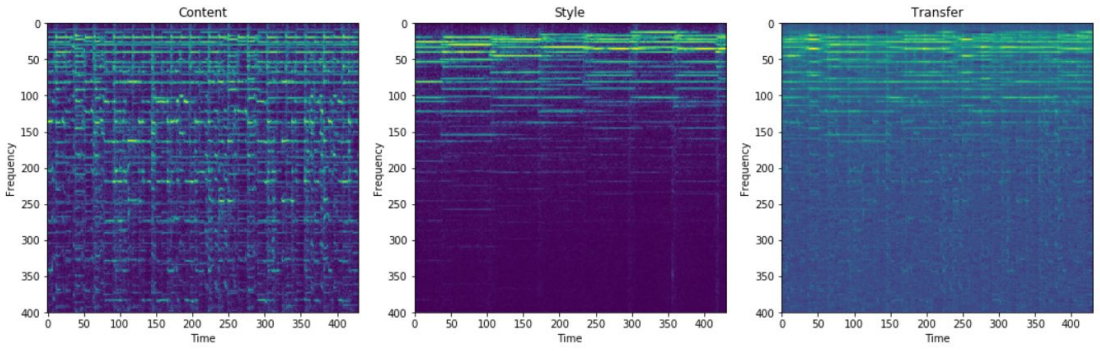


Figure 6. Example of one case of style transfer using the baseline model.

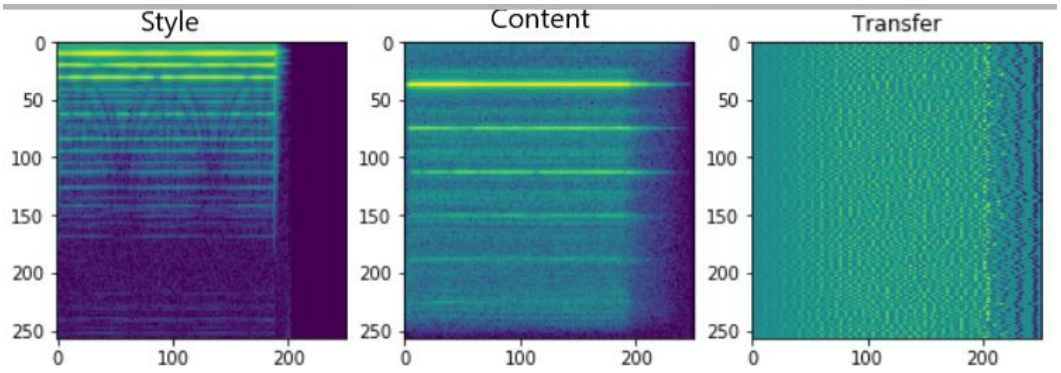


Figure 7. Example of one case of style transfer using our custom model.

Table 1. Probabilities assigned by the 21 classes for the flute content with the organ's style.

Instrument	Likelihood
bass_electronic	0.04329029
bass_synthetic	0.00188515
brass_acoustic	0.00457035
flute_acoustic	0.05407558
flute_synthetic	0.08487992
guitar_acoustic	0.00138424
guitar_electronic	0.00767361
guitar_synthetic	0.00199876
keyboard_acoustic	0.00847936
keyboard_electronic	0.01563566
keyboard_synthetic	0.00488552
mallet_acoustic	0.00121379

mallet_electronic	0.01692154
mallet_synthetic	0.15493688
organ_electronic	**0.28923148**
reed_acoustic	0.00492667
reed_synthetic	0.00490197
string_acoustic	0.0136157
synth_lead_synthetic	0.03008245
vocal_acoustic	0.23907918
vocal_synthetic	0.01633188

G. Conclusions

Though we were not yet able to produce satisfactory timbre style transfer, our results indicate this is a hopeful application of the style transfer method. Given more time for training and computational resources, we believe our methods can produce novel and interesting results based on various inputs. Unlike with image inputs, the axes of our data are not arbitrary, i.e. whereas an image rotated 90 degrees still contains the same information, a spectrogram rotated 90 degrees loses its meaning entirely. Thus the methods for image style transfer may not be wholly appropriate when using spectrograms as inputs. Rather than using convolutional neural networks, which are highly effective in image processing, it would be beneficial to explore time series signal analysis models such as LSTM (long short-time memory), which tends to preserve long-term dependency. As other projects have proposed, a better model would likely consist of multiple networks: one to distinguish the components of sound, and one to generate new audio [2, 8]. Additionally, using the Short-Time Fourier Transform may be less than optimal for feature extraction, with other methods such as Mel-Frequency Cepstrum Coefficient possibly yielding improved results [1]. However, one advantage we found to using STFT was the existence of a relatively straightforward process to reconstruct raw audio from the spectrogram. Our results were promising, if limited, and we are excited about the progress the future holds for this particular problem.

H. Roles

Scarlett Hwang: Data preprocessing, literature review, intro/background/references section,video editing/plotting/recording using Adobe premiere pro.

Bingying Liu: Dataset and feature selection/reference, literature review, data retrieval from database, XGBoosting classifier building, training and testing, video recording/plotting.

Joaquin Menendez: XGBoosting and Random Forest classifiers training and testing, data retrieval from the database, develop of the pipeline to analyze the style

transfer quality, video recording

Nathan Scheperle: Data collection from AudioSet and NSynth databases, literature review of current audio processing and generation methods, baseline and VGG model construction and training.

Muxin Diao: Data preprocessing, literature review, and intro/background/references section. Video editing/plotting/recording using Adobe pro.

I. References

[1] Aaron van den Oord, Sander Dieleman. “WaveNet: A Generative Model for Raw Audio”. arXiv:1609.03499v2 [cs.SD] (19 Sep 2016)

[2]Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Douglas Eck, Karen Simonyan, and Mohammad Norouzi. "Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders." (2017).

[3] Prateek Verma, Julius O.Smith. “Neural Style Transfer for Audio Spectrograms”. Center for Computer Research in Music and Acoustics (CCRMA), Stanford University(2018)

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

[5] Anthony Perez, “Style Transfer for Prosodic Speech”. Stanford University (2017).

[6] NSynth dataset, https://magenta.tensorflow.org/datasets/nsynth

[7] Leon A. Gatys, Image Style Transfer Using Convolutional Neural Networks”, Centre for Integrative Neuroscience, University of Tubingen, Germany

[8] Dabi Ahn, Kyubyong Park. “Voice Conversion with Non-Parallel Data: Speaking Like Kate Winslet” https://github.com/andabi/deep-voice-conversion

[9] Dmitry Ulyanov, Vadim Lebedev. “Audio texture synthesis and style transfer

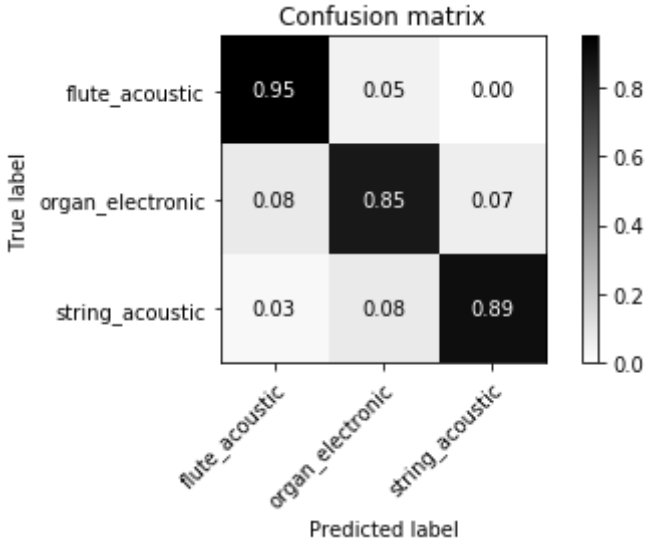
[10]James, G., Witten,D., Hastie, T., & Tibshirani, R. (2013d). 8 Tree-Based Methods In *An introduction to statistical learning* (pp. 303-321). New York: Springer.

J. Appendix

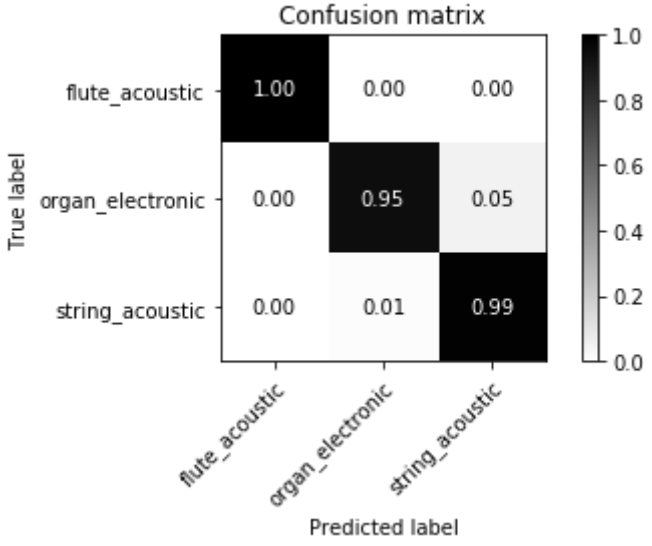
During the developing of the classifiers used to evaluate timbre style transfer, we tried several models employing the SFFT spectrum and the MFFC as features.
Despite the fact that we observed that using the output of the SFFT was more accurate to predict the instrument (both for 3 classes as for 21) we were not able to reconstruct a sound with the same quality that the ones used to train the classifiers. This caused inconsistencies between the number of features used to train with the number of features of the reconstructed audio. Below are shown the models that were excluded from the final document.

Random - Forest 3 classes

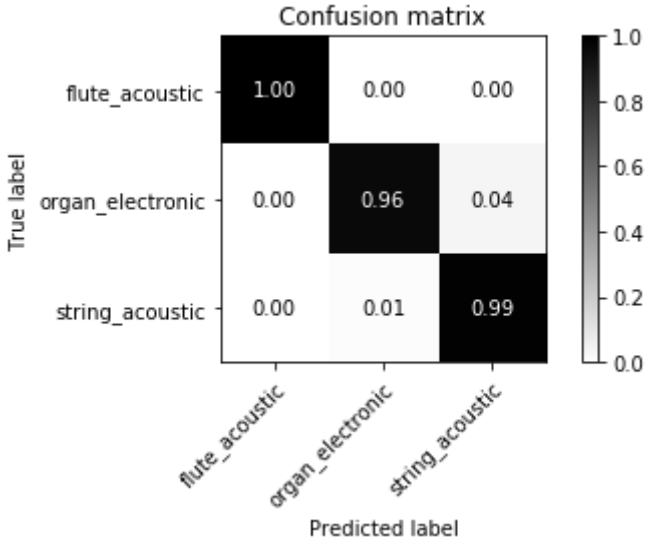
Using MFCC



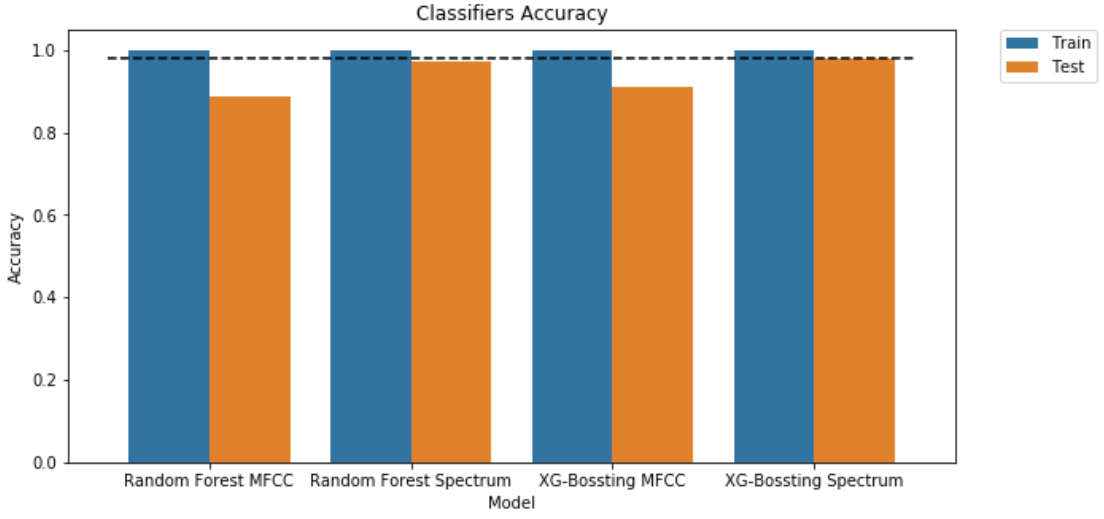
Using Spectrum



XG - Boosting for 3 classes
Using spectrum



Accuracy comparison for the different models tried to classify 3 classes.



Random Forest for 21 Classes using Spectrum

Criteria:
Only classes with 500 or more samples
Taining (300) per class
Testing (50) per class
Accuracy classification Training = 1.0
Accuracy classification Test = **0.739**
F1 score = 0.728