Style Transfer for Audio using Convolutional Neural Networks

Bhaumik Choksi
K.J Somaiya College of Engineering, Vidyavihar, Mumbai

Alisha Sawant
K.J Somaiya College of Engineering, Vidyavihar, Mumbai

Swati Mali
K.J Somaiya College of Engineering, Vidyavihar, Mumbai

ABSTRACT
Convolutional neural networks have recently become extremely popular in various deep learning applications. One such application is style transfer for images. Following this trend, this paper explores how this technique can be applied to audio data. The technique discussed involves combining the content features of one audio sample with the style features of another audio sample. The results produced show how a Convolutional Neural Network can be used to extract features from audio signals. The paper also discusses the various modifications made in the algorithm used for image style transfer in order to apply it to audio signals.

General Terms
Machine Learning, Neural Networks.

Keywords
Convolutional Neural Network, Deep learning, Style Transfer, Gram Matrix.

1. INTRODUCTION
Style transfer with respect to images is the process of synthesizing the texture of one image and applying that texture to a target image in order to achieve a stylized version of the target image. Visually, this creates an image that has the structural elements of the target image but the textural elements of the style image, resulting in an artistic representation of the original image. Gatys et al. use this to obtain impressive results [1].

A style transfer for audio signals is analogous to that of images. The "texture", or style, of one audio sample is extracted using CNNs which is then transferred to the target audio sample. This creates an audio with the structural and tonal elements of the target data but the beat of the styling audio. The evolution of Deep CNNs to extract high-level semantic information from audio signals has enabled the synthesis of textures for the audio signal concerned.

Convolutional Neural Networks (CNNs) are a class of deep neural networks that are highly effective at analyzing images and other connected patterns. CNNs are powerful and effective tools for processing spatial data, as compared to regular fully connected neural networks. Most CNNs use Rectified Linear Unit (ReLU) activations to provide non-linearity to their output. ReLU activation is more efficient that other activation function and usually trains the network several times faster. CNNs have the ability to extract high-level features or textures from the given data, so as to provide an abstract representation of the image. Adding more convolutional layers provides higher-level and more complex features at every subsequent level.

This paper discusses an application of Convolutional Neural Networks to perform style transfer between two audio files. The style of an audio file may consist of tonal and rhythmic elements of the audio. With the help of CNNs, it is possible to extract these elements and transfer them to another audio sample.

2. RELATED WORK
Style transfer is an interesting problem and has lately received a lot of attention. This work draws inspiration from implementations of style transfer for images done by Gatys et al. [1]. These techniques make use of Convolutional Neural Networks to synthesize textures from the given data which are then used to style the target data.

Recent methods involve using Gram matrices [1], [9] of neural activations from a given layer of a CNN to represent the artistic features of an image. Previous studies have also demonstrated the Short Term Fourier Transform (STFT) can be used instead of regular Fourier Transform in order to minimize the time taken for computations.

3. METHODOLOGY

3.1 Audio Preprocessing
The audio files used for this paper were in WAV format, which are sampled at 44100 Hz. It is necessary to use formats like WAV as they store data in an uncompressed from which facilitates easy data extraction. For the sake of simplicity, this implementation operates on only single channel audio files and not stereophonic audio. The inputs consist of two audio files, the content audio and the style audio.

First, the audio signal is transformed from the time domain to the frequency domain using Fourier Transform. Since the computational cost of Fourier Transform over the entire audio file is very high, Short Term Fourier Transform (STFT) is performed instead. STFT performs involves dividing the audio signal into equal size segments and then performing Fourier Transform on these segments individually. For N segments each of length L, the output with will be a matrix given as

$$M \in \mathbb{R}^{N \times L}$$

Since hearing works on a logarithmic scale, the log of each element m_{ij} in the matrix is calculated as

$$M_{ij} = log(1 + m_{ij})$$

3.2 Modelling the Convolutional Network
The convolutional layer is required to extract high-level features from the audio signals. This implementation uses a single convolution layer consisting of 4096 filters. The filters

are initialized using the Glorot Normal Initializer which draws samples from a truncated normal distribution centered on 0 with

$$stddev = \sqrt{\frac{2}{fan_{in} + fan_{out}}}$$

where fan_in is the number of input units and fan_out is the number of output units. ReLU activation is applied to the output.

3.3 Style Extraction

The matrices obtained after preprocessing the content and style audio signals are reshaped appropriately to match the input shape requirement of the convolutional network. Both the content and style matrices are individually passed through the Convolutional Neural Network to obtain the content feature map and the style feature map respectively.

In order to extract the high-level style features, we use a Gram matrix. A Gram matrix is the inner-product of the style feature map with itself. This representation captures the covariance of the style features. The Gram matrix is given by

$$G = \begin{bmatrix} X_1 \cdot X_1 & X_1 \cdot X_2 & \cdots & X_1 \cdot X_n \\ X_2 \cdot X_1 & X_2 \cdot X_2 & \cdots & X_2 \cdot X_n \\ \vdots & \vdots & \ddots & \vdots \\ X_n \cdot X_1 & X_n \cdot X_2 & \cdots & X_n \cdot X_n \end{bmatrix}$$

3.4 Defining the Loss

Content loss and style loss are defined independently. In order to determine the loss, an input sample with random values is fed into the network and the output obtained from it is processed to extract features from it. A Gram matrix representation for these output features is also obtained.

The loss determines the ability of the network to generate the same response as the content and style features from the random input sample.

The content loss is calculated as the squared error between the content features obtained during the previous step and the output features for the random input. The content loss is given by formula

$$\mathcal{L}_{content} = \sum_{i=0}^{n} (y_{net} - y_{content})^2$$

The style loss is defined as the squared error between the Gram matrix for the style audio sample and the Gram matrix obtained from the features of the random input. It is given by the formula

$$\mathcal{L}_{style} = \sum_{i=0}^{n} (Gram_{net} - Gram_{style})^2$$

The total loss is defined as the linear combination of the content loss and the style loss and is given by the formula

$$\mathcal{L}_{total} = \alpha \mathcal{L}_{content} + \beta \mathcal{L}_{style}$$

3.5 Training the Network

The network is trained so as to minimize the total loss. The L-BFGS-B optimization algorithm is used for training, since it is known to provide good results for the style transfer application. The learning rate can be adjusted suitably to obtain better results. This implementation uses a learning rate of 0.001. The maximum epochs for training can be limited so as to lower the training time, although too few epochs will result in lower output quality.

3.6 Reconstructing the Audio

After the training is complete, the final output from the network is obtained. This output needs to be processed in order to reconstruct the audio from it. This is done by first performing exponentiation of each element in the output. The Inverse Short Term Fourier Transform (ISTFT) is calculated for each element in order to convert the signal back into the time domain from the frequency domain. The output is cast to integer format and written to a WAV file.

4. EXPERIMENTS

The audio dataset consisted of instrumental and vocal music. The instrumentals included pianos, trumpets and drums. They were converted to single channel WAV files sampled at 44100 Hz. The length of the audio files used for training was 20 seconds each and the length of the output audio file was 10 seconds.

The model was trained for 500 epochs to generate each output. A learning rate of 0.001 was chosen. The value of alpha (trade-off between content and style in the output) was set to 0.1. The block size chosen for STFT was 2048 values per sample.

The model was implemented in Python using the Tensorflow library. The model was run on a GeForce 940M GPU.

The loss was recorded for every epoch, and it was observed that the loss reduced in general, as the number of epochs increased. In order to account for the large variations of loss during training, a log of the loss was plotted, in order to better visualize the change in loss.

.

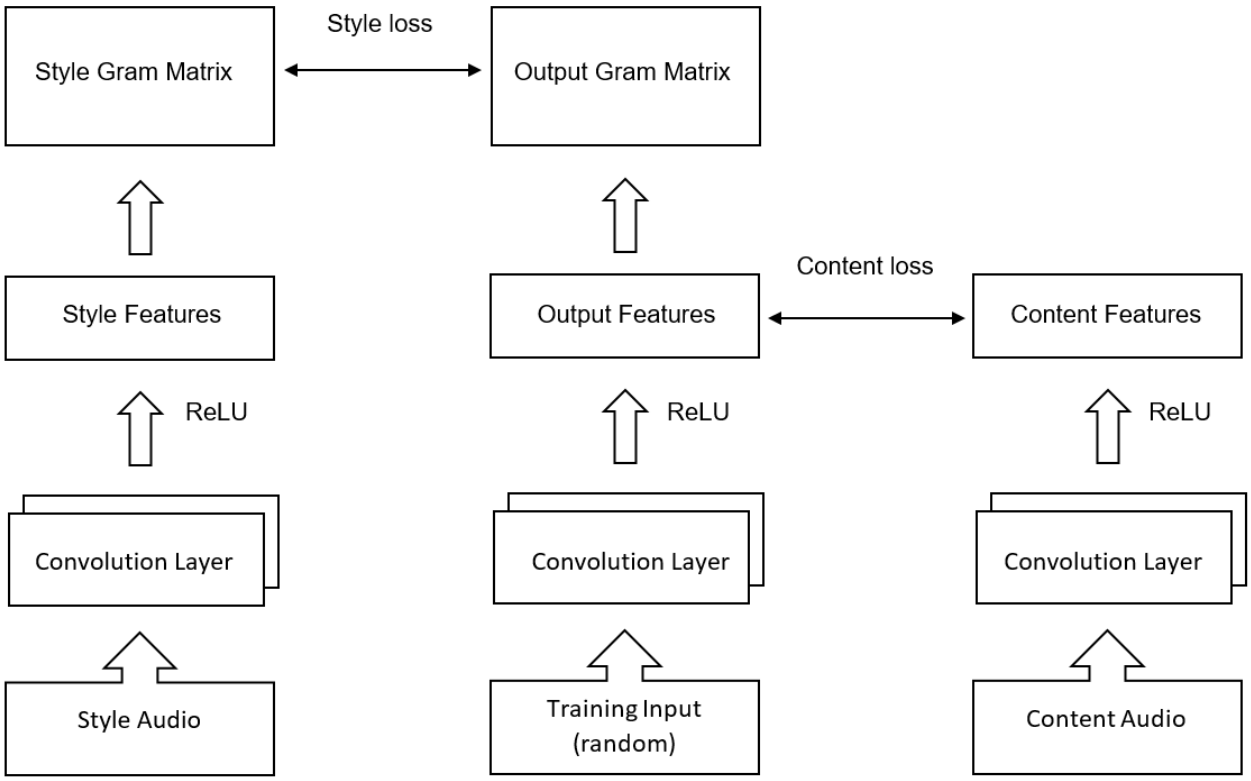

Fig 1: Convolutional Neural Network Architecture for Audio Style Transfer

5. RESULTS

The graph obtained by plotting the loss against the number of epochs shows that the loss decreases in general as the number of epochs increases. The decrease in loss is rapid initially, but eventually remains constant, showing very little change thereafter.

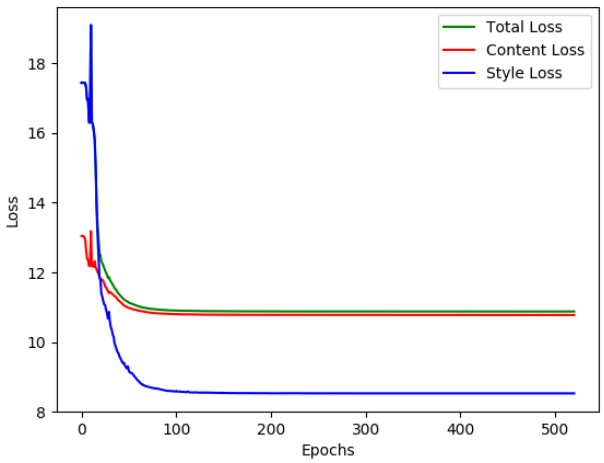

Fig 2: Plot of the loss versus the number of training epochs

The plot of the correlation shows that the similarity between the content audio and the output audio increases rapidly in the beginning and then remains constant. This is in agreement with the steep initial decrease of the total loss. This shows that the similarity between the output features and the content features increases as the number of training epochs increase.

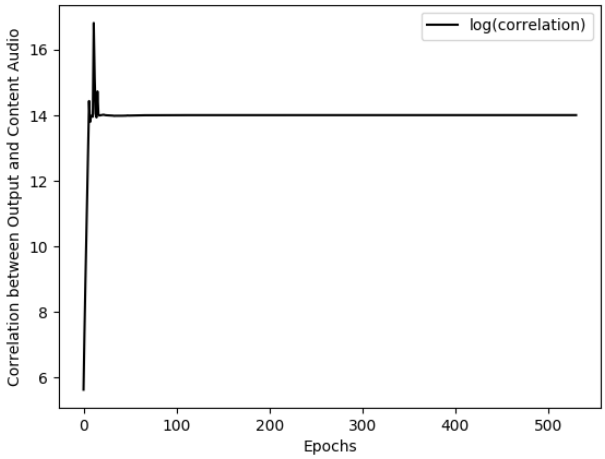

Fig 3: Plot of the correlation between the output and the content audio on a logarithmic scale

A blind review of 8 people was conducted and each person was asked to rate 5 audio outputs, generated from various combinations of content and style audios, on a scale of 1 to 5, 1 being the lowest. The average of the 8 ratings for each audio was calculated and the results are tabulated below.

This is done since the quality of the output is a subjective term, and does not have a purely mathematical meaning, nor can it be depicted using equations or graphs. The blind tests give a better understanding of how pleasant the generated outputs sound to humans.

Table 2. Average rating given by 8 blind testers for different combinations of content and style audio

Type of Music	Average Rating
Speech + Drums	2.38
Instrumental + Trumpet	2.25
Instrumental + Drums	3.75
Rhythm + Drums	3.44
Rhythm + Violin	2.5

The blind test results show that certain combinations of instrumental content audio and style audio produce better results, as compared to others. Speech samples performed poorly, since the various transformations caused the speech to lose its integrity and structure, leading to a muffled output. On the other hand, rhythmic and instrumental audio samples produced decent to good results.

6. CONCLUSION

The results obtained from this work can be easily distinguished from conventionally produced audio, due to the mathematical origins of the audio, as opposed to audio composed entirely by humans using various instruments. The output shows that it is possible to transfer the style features of one audio to another to create an entirely new audio that is a unique blend of the two. The degree to which the style is transferred can be varied as desired. This technique could find applications in accent modification of recorded human speech, song remixing, rhythm manipulation and so on.

7. FUTURE SCOPE

An interesting prospect for the future would be increasing the number of convolutional layers to extra more high-level features to facilitate better style transfer. Also, to improve the quality of the output, two optimizations can be made. These are, to first reduce the noise level in the output as much as possible and second, to process multi-channel audio files. Further, the number of epochs could be increased to improve the output audio quality, but this is highly hardware dependent.

A variation in this implementation could be the transfer of multiple styles, that is, the use of two or more style audios for styling the content.

8. REFERENCES

[1] Gatys, Leon A., Alexander S. Ecker, and Matthias Bethge. "Image style transfer using convolutional neural networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

[2] Neumann, László, and Attila Neumann. "Color style transfer techniques using hue, lightness and saturation histogram matching." Computational Aesthetics. 2005.

[3] Ruder, Manuel, Alexey Dosovitskiy, and Thomas Brox. "Artistic style transfer for videos." German Conference on Pattern Recognition. Springer International Publishing, 2016.

[4] Bruckner, Stefan, and M. Eduard Gröller. "Style transfer functions for illustrative volume rendering." Computer Graphics Forum. Vol. 26. No. 3. Blackwell Publishing Ltd, 2007.

[5] Johnson, Justin, Alexandre Alahi, and Li Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution." European Conference on Computer Vision. Springer International Publishing, 2016.

[6] Selim, Ahmed, Mohamed Elgharib, and Linda Doyle. "Painting style transfer for head portraits using convolutional neural networks." ACM Transactions on Graphics (ToG) 35.4 (2016): 129.

[7] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Advances in neural information processing systems. 2012.

[8] Drineas, Petros, and Michael W. Mahoney. "On the Nyström method for approximating a Gram matrix for improved kernel-based learning." journal of machine learning research 6.Dec (2005): 2153-2175.

[9] Li, Yanghao, et al. "Demystifying neural style transfer." arXiv preprint arXiv:1701.01036 (2017).

[10] Griffin, Daniel, and Jae Lim. "Signal estimation from modified short-time Fourier transform." IEEE Transactions on Acoustics, Speech, and Signal Processing 32.2 (1984): 236-243.