Dreaming in Music: Audio Classification and Modification through Deep Learning

Itrat Akhter
iaakhter@cs.ubc.ca

Alexandra Kim
kimalser@cs.ubc.ca

Sijia Tian
candice@cs.ubc.ca

***Abstract*— The goal of this work is to test whether or not spectrograms are a reasonable way to visually represent music as image input to feed to deep learning networks. We implemented music classification on songs' spectrograms using a CNN and then utilized the trained network to modify spectrograms, similar to the Google's deep dream method. We have also attempted to change the spectrograms through neural style transfer and CycleGAN. In all modification methods, the spectrograms were converted back to audio and assessed qualitatively. While classification achieves a good accuracy, music modification may require tuning and post-processing but does produce promising sound tracks.**

I. Introduction

In this work, we use deep learning methods to classify and modify music. There are numerous ways that music can be represented when using deep learning methods. Some of the possible representations are - MIDI, text, notes, metadata like pitch class [1]. In our case, we visually represent music using spectrograms.

We can reason about the effectiveness of using spectrograms in deep learning by classifying them based on genres of music. If our classification results were not up to par, it would be safe to say that representing music using only spectrograms is not ideal. Our classification method using just spectrograms, however achieves a good accuracy which led us to further exploring their use in music modification.

Deep learning has been successfully applied to a number of areas, such as music classification and generation. However, there is a dearth of research in the area of modifying music using deep learning approaches. We used Google's Deep Dream approaches, neural style transfer and CycleGAN to modify music with spectrograms as input. The modified spectrograms were then converted back to audio. Our intention was for the modified track to be of reasonable quality and yet noticeably different from the original version.

II. Related Work

A. Learning in Music Generation

To the best of our knowledge, there has not been any work done in using deep learning methods to modify music. However, there has been a lot of focus on generating music. Google's Magenta team used RNNs for polyphonic music generation [2], [3]. They modeled polyphony as a single stream of note events with special START, STEP_END, and END symbols. Deep Jazz generated Jazz music using RNN [4]. They used MIDI file to represent the music. One of the common aspects in such existing methods is the use of RNN and non-visual representation of music. In this project, we attempted to modify music by representing it visually using spectrograms.

B. Music Classification

The work in audio classification using convolutional neural networks (CNNs) has been studied in [5]. Music genre classification in particular has been explored in [6], [7], [8], and spectrogram based classification has been successfully implemented in [9], [10], [11] using various models, varying from k-nearest neighbors to deep CNNs. We are especially interested in classification of music spectrograms using CNNs for it will allow us to learn most prominent features of each music genre to later 'amplify' those in a given audio track, similar to the deep dream method.

C. Deep Dream

One of interesting methods of image modification is Deep Dream [12], [13]. Deep dream iteratively enhances an input image to elicit a certain behaviour by feeding the previous iteration's output as an input, and in every pass, 'tweaking' the image to look more like a certain class of objects (from the perspective of the neural network). Eventually, the network will modify the image so much that it can 'see' the objects of that certain class, with high confidence.

D. Neural Style Transfer

Another method to modify visual input is through neural style transfer [14]. The core idea behind it is to define two distances $\mathcal{L}_{content}$ and $\mathcal{L}_{style}$ that would measure how different an input image is to the given content and style images based on the features from a CNN. Then, the input image is run several times through the net in order to minimize both of the distances.

E. GANs

Previously, image-to-image translation has been achieved by learning the mapping between input and output images using a set of aligned image pairs. But in many scenarios, paired training data are not available. In [15], the authors managed to learn a mapping between two domains in the absence of paired data by using generative adversarial network and introducing a cycle consistency loss. Inspired by this work, we applied the same model to spectrogram images from different genres and achieved music genre transformation.

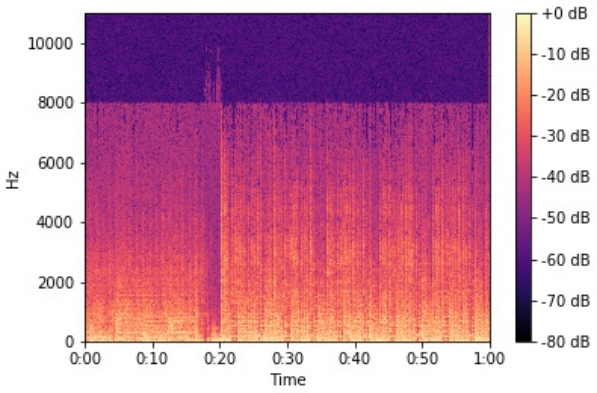

Fig. 1. An example of a spectrogram.

III. Methods

We converted music to spectrograms, and then used a CNN to classify them based on the genres of the music - hip hop, rock, classical and electronic. The trained CNN was then used to implement Google's Deep Dream which allowed us to modify spectrograms. We also evaluated the effectiveness of using neural style transfer and GANs on spectrograms to modify music.

A. Preprocessing

Spectrograms depict the spectrum of frequencies of sound as they vary with time. The audio undergoes a few transformations before resulting in a spectrogram of desired scale that is later used in our experiments. We extracted the raw waveform data x from an mp3 file using python's librosa library. The spectrogram was extracted from x by the following sets of transformations.

$$D = stft(x),$$

where $stft$ is short-time Fourier transform and D is a 2 dimensional complex matrix where $D[f, t]$ holds information about the amplitude and phase of frequency bin f at time t.

$$M = 10 * log(|D|),$$

where each entry of M is the amplitude in decibal units. M is the spectrogram that is used in our methods. Figure 1 shows an example of such a spectrogram.

One disadvantage of using spectrograms to represent music is that the conversion from audio to spectrograms is not invertible, since spectrograms only hold the amplitude information. Hence, in our music modification processes, it is important to supply phase information that can be combined with the modified spectrogram to get the audio back. In our cases, we mostly used the original phase of the unmodified song to get the audio relevant to the modified spectrograms. In our deep dream approach, we also experimented with modifying the phase of the original song and using the modified phase for conversion to audio.

To make the size of the spectrogram more reasonable and also to expand the dataset, we divide each 30-second period song into three 10-second chunks. Each chunk is converted to spectrogram using the method described above, resulting in a matrix of size 1025×860. The amplitude of the spectrogram is then converted to gray scale image by scaling to the range (0, 1). To preserve sound quality, each gray scale image is resized to 512×512, instead of the regular input size of image classification networks.

B. Classification

For classification we deployed ResNet-18 model, first introduced in [16]. Given a spectrogram of a song, we trained the ResNet-18 to classify it as being one of the following 4 music genres: classical, hiphop, rock and electronic. Cross-entropy loss is used during training.

As discussed in preprocessing, we are not sure how to supply phase information to modified songs. A naive approach would be using the original phase of the song. Besides this, we think it would be interesting to see if modification on amplitude and phase separately or jointly would give better results in deep dream. So besides classifying on amplitude alone as previous works, we also experimented with classifying on phase information alone, and on amplitude and phase at the same time.

As mentioned above, the input to the network for classifying on either amplitude or phase alone are gray scale images of size $512 \times 512 \times 1$. For classifying on both amplitude and phase, we stack the gray scale images of phase and amplitude information of the same chunk, resulting in an image of size $512 \times 512 \times 2$.

C. Deep Dream

Utilizing the trained ResNet-18 III-B, we applied Google's Deep Dream method of image modification to transform an input spectrogram. The goal of deep dream is to emphasize a particular layer of the trained CNN on the input spectrogram. This is done by maximizing the L2 norm of activations of the layer in the neural network. In this version of the deep dream, the modified spectrogram depends on the most prominent features learned by the CNN. However, it is also possible to attempt to modify a song so that it sounds similar to another song (known as the guide song) from a different genre. This is done by maximizing the dot product between activations of the original spectrogram and best matching correspondences of the guide spectrogram. Maximization in both cases is done by applying gradient ascent at multiple octaves (resolutions) of the spectrogram images.

D. Neural Style Transfer

Applying image style transfer [14] to the spectrograms, we have tried to transfer the style of one song to another song. Following the original work, we have used a pretrained VGG-19 neural network to extract features. To transfer the style, we create a new image that matches the content of the original track's spectrogram and the style of a spectrogram of a song of a different genre. To do so, we minimize both

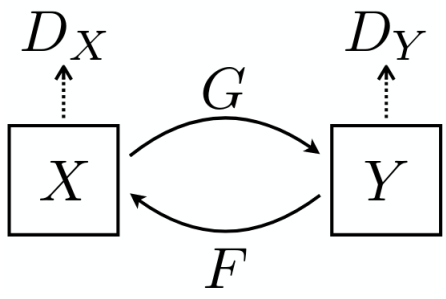

Fig. 2. Overview of cycleGAN model

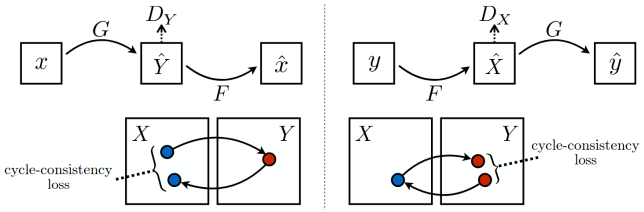

Fig. 3. An illustration of cycle consistency loss

the content distance $\mathcal{L}_{content}$ and the style distance $\mathcal{L}_{style}$. The loss function is defined as follows:

$$\mathcal{L}_{total}(c, s, g) = \alpha\mathcal{L}_{content}(c, g) + \beta\mathcal{L}_{style}(s, g),$$

where x is an image being generated, c and s are the content and style images, and α and β are parameters weighting the importance of content and style reconstruction respectively. Refer to the original paper [14] for the definitions of the two distances $\mathcal{L}_{content}$ and $\mathcal{L}_{style}$.

E. CycleGAN

As we have talked about in Section II-E, [15] successfully achieved image-to-image translation using GANs. In our project, we examined if this architecture can be applied to spectrograms of different genres of songs and achieve song style transformation. As shown in Figure 2, the model is learning two mapping functions, $G : X \rightarrow Y$ and $F : Y \rightarrow X$ between two domains (i.e. spectrogram images from two music genres). Discriminator D_Y encourages G to translate X into outputs indistinguishable from domain Y, and vice versa for D_X and F. However, this mapping cannot guarantee that the learned function will map an individual input to the desired output. To solve this issue, the authors in [15] proposed another cycle consistency loss to make sure that the translated image can be translate back to the original image, as shown in Figure 3.

IV. Dataset

We ran our experiments on the Free Music Archive (FMA) dataset that contains full-length music tracks along with their pre-computed features, and metadata including genre, tags, artist information, etc. The dataset has over 100,000 music files corresponding to 161 genres arranged in a hierarchical order of genres [17], [18].

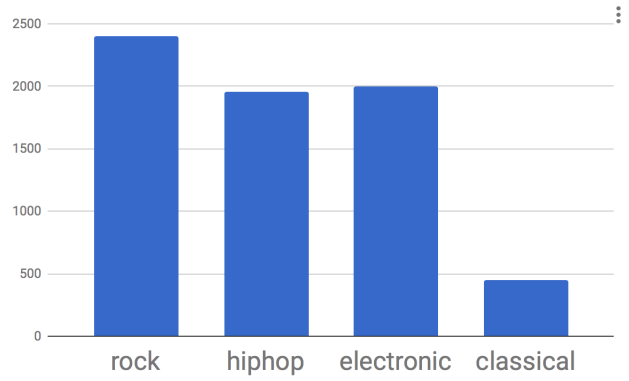

Fig. 4. Distribution of classes

TABLE I
Classification Results

Using only amplitude	83.48%
Using only phase	63.31%
Using both amplitude and phase	81.50%

In the scope of our project, we used four genres, which are rock, pop, electronic and classical. A total of 6,811 examples were obtained from a subset of the FMA dataset referred to as *fma_medium*, which consists of 25,000 30-second tracks. Dividing each example into 3 chunks gave a total number of 20,433 examples. 80% of the data was used for training, and 20% of the data was used for testing. The distribution of the number of examples in each class is shown in Figure 4.

V. Experiments

A. Classification

During training, we used the architecture described above, and used a batch size of 16 with an initial learning rate of 10^{-3}. The training process was ran for 500 epochs [19].

Results for classification is shown in Table I. As can be seen from the table, classifying on amplitude alone gives the best accuracy, which is 83.48%. Using both amplitude and phase gives a slightly lower accuracy, while classifying on phase alone has the lowest accuracy, which is 63.31%. This might imply that amplitude contains more information regarding a song's genre than phase.

The confusion matrix for classifying only on amplitude is shown in Figure 5. From the confusion matrix we can see that even though classical has much less examples than the other classes, it has the highest class accuracy. Electronic and hip hop are sometimes misclassified as each other. Electronic, hip hop and rock are rarely misclassified as classical.

B. Deep Dream

There are two kinds of deep dream experiments that we performed - one where we do not try to control the modification process (Dream) and the other where we try to control the modification process with a guide song/spectrogram (Dream Control). We applied the deep dream approaches at the 13th convolutional layer of the trained ResNet-18. The

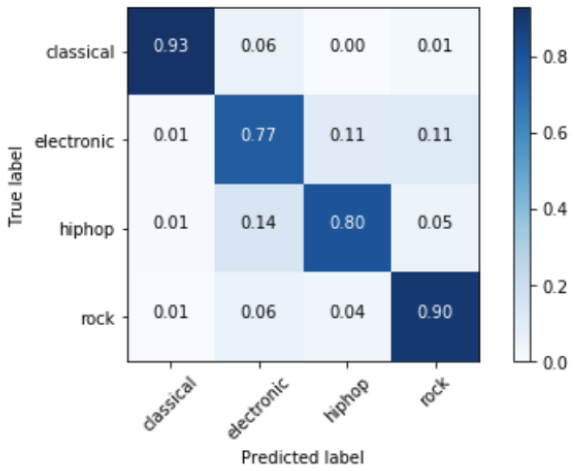

Fig. 5. Confusion matrix for classifying on only amplitude

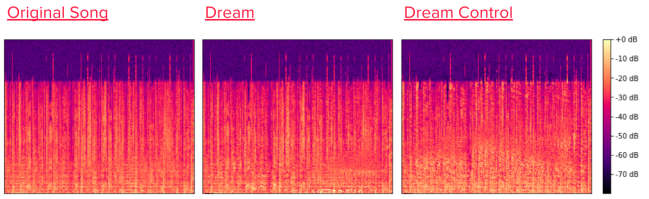

Fig. 6. Deep Dream Results. From left to right: original hip hop song, modified hip hop song by deep dream and modified hip hop song by dream control with a classical song as a guide song

number of octaves used was 6 and the octave scale used was 1.4 - meaning that the deep dream was applied on 6 different resolutions of images where each resolution was 1.4 times lower than the previous image starting from the original image. At each octave, gradient ascent was applied for 20 iterations [20].

Figure 6 shows the output of a deep dream example. Dream like features are evident in the lower portion of the spectrograms. In this case, the original song is a hip hop song. In the Dream version, the modified song ends up having an unusual and pleasant tone towards the end of the song. However, in the Dream Control version, we attempted to make the hip hop song sound like a classical song but the results are not satisfactory. The modified version does not have any classical elements in it and sounds like the original version with added noise. The audio results can be found in `https://kimalser.github.io/dreaminginmusic`.

In the examples discussed above, deep dream approaches were applied on the CNN trained on just amplitude of frequencies and the conversion of the modified spectrograms back to audio involved the phase of the original song. We also experimented with applying deep dream on the CNN trained on just phase of the frequencies and the CNN trained together on both the amplitude and phase. In these cases, the conversion of the modified spectrograms back to audio

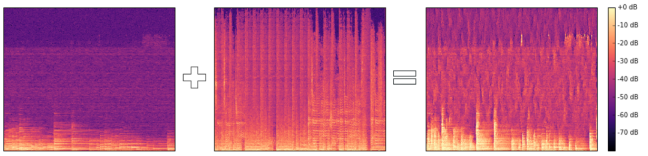

Fig. 7. Neural Style Transfer Results. From left to right: original classical piece, "stylistic" rock song, and the resulting generated song.

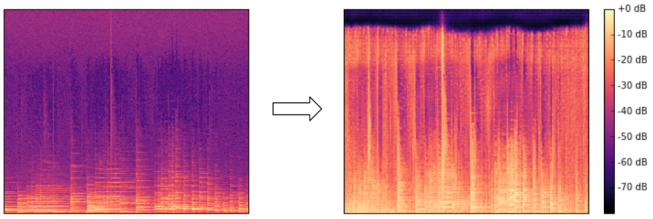

Fig. 8. Cycle GAN Results.

involved the use of modified phase resulting from the deep dream approaches. The quality of the results in either of these cases is worse than that resulting from using the original phase of the song.

C. Neural Style Transfer

For neural style transfer [21] we attempted to convert some classical songs to rock. For this set of experiments, we used the ratio of content to style weights α/β of 1×10^{-3}. In Fig.7, there is an example of how the original classical song's spectrogram has changed after applying a stylistic change of a rock song. The audio results for the shown spectrograms can be found in `https://kimalser.github.io/dreaminginmusic`. To convert the spectrograms back to audio, they were combined with the original songs' phases.

D. CycleGAN

We have also tried converting classical pieces into rock songs using a CycleGAN model [22]. The training set consists of 5397 rock and 1074 classical spectrograms. Due to time and hardware limitations we were only able to train CycleGAN for 13 epochs with a batch size of 1. Additionally, to speed up training, the spectrogram images were resized from 512×512 to 256×256 prior to training. The losses for CycleGAN generators and discriminators still oscillate by the end of the 13th epoch, which doesn't seem to be atypical, given the number of epochs. Training for a larger number of epochs (e.g. 100, 200) might be needed for losses to converge.

Fig. 8 shows a sample classical song conversion. The audio results can be found in `https://kimalser.github.io/dreaminginmusic`.

VI. Conclusion

In this work, we used visual representations of songs as inputs to state-of-the-art deep learning methods to test whether they are suitable for music classification and modification. The results show us that classification on amplitude

of songs yields a good accuracy, and confusion matrix of the classification result is well in accordance with the features of the genres, suggesting that amplitude alone carry sufficient information regarding their genres, and spectrograms are well suited for the task of music classification. In our modification methods, deep dream, neural style transfer and CycleGAN, resulting spectrograms suggest that the methods can be applied to spectrograms successfully in theory. And they did produce some interesting and promising sound tracks. However, many converted songs have problems such as unapparent changes from original songs and noise. Post-processing such as filtering might help with improving the quality of the results. More reasonable modifications of songs might require a better understanding of the relationship between phases and songs, or operating on the waveform directly.

VII. Future work

In the future, the first step would be applying post-processing to current results and see if this could produce music tracks of higher quality. It would also be of interest to see whether changing 2-D convolution in the network to 1-D, and operating on waveform directly could solve the problem of missing proper phase information. It might be a more natural way of processing music since it inherently contains time dimension as its component.

Appendix

A. Timeline

Week	Task	Person
March 4 - 10	Audio/image conversion	Itrat, Sijia
March 11 - 17	Train CNN, Neural Style Transfer	Sijia, Alex
March 18 - 24	Deep Dream and GANs	Alex, Itrat
March 25 - 31	Evaluation	All
April 1 - 15	Evaluation, Writeup	All

TABLE II
Timeline and distribution of work for this project

Table II shows how the work has been distributed among the team members and the approximate dates and deadlines for each milestone.

References

[1] Jean-Pierre Briot, Gaëtan Hadjeres, and François Pachet. Deep learning techniques for music generation - A survey. *CoRR*, abs/1709.01620, 2017.

[2] Magenta: Make music and art using machine learning. https://magenta.tensorflow.org/.

[3] Magenta's polyphony rnn. https://github.com/tensorflow/magenta/tree/master/magenta/models/polyphony_rnn.

[4] Ji-Sung Kim. Deep jazz. https://deepjazz.io/.

[5] Shawn Hershey, Sourish Chaudhuri, Daniel PW Ellis, Jort F Gemmeke, Aren Jansen, R Channing Moore, Manoj Plakal, Devin Platt, Rif A Saurous, Bryan Seybold, et al. Cnn architectures for large-scale audio classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, pages 131–135. IEEE, 2017.

[6] Sander Dieleman and Benjamin Schrauwen. End-to-end learning for music audio. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 6964–6968. IEEE, 2014.

[7] Tom LH Li, Antoni B Chan, and A Chun. Automatic musical pattern feature extraction using convolutional neural network. In *Proc. Int. Conf. Data Mining and Applications*, 2010.

[8] Weibin Zhang, Wenkang Lei, Xiangmin Xu, and Xiaofeng Xing. Improved music genre classification with convolutional neural networks. In *INTERSPEECH*, pages 3304–3308, 2016.

[9] Hrishikesh Deshpande, Rohit Singh, and Unjung Nam. Classification of music signals in the visual domain. In *Proceedings of the COST-G6 Conference on Digital Audio Effects*, pages 1–4, 2001.

[10] Honglak Lee, Peter Pham, Yan Largman, and Andrew Y Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in neural information processing systems*, pages 1096–1104, 2009.

[11] Yandre MG Costa, Luiz S Oliveira, and Carlos N Silla Jr. An evaluation of convolutional neural networks for music classification using spectrograms. *Applied soft computing*, 52:28–38, 2017.

[12] Inceptionism: Going deeper into neural networks. https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html.

[13] deepdream. https://github.com/google/deepdream.

[14] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. A neural algorithm of artistic style. *arXiv preprint arXiv:1508.06576*, 2015.

[15] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *arXiv preprint arXiv: 1703.10593*, 2017.

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[17] Kirell Benzi, Michaël Defferrard, Pierre Vandergheynst, and Xavier Bresson. FMA: A dataset for music analysis. *CoRR*, abs/1612.01840, 2016.

[18] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. Fma: A dataset for music analysis. https://github.com/mdeff/fma.

[19] Two stream action recognition in pytorch. https://github.com/jeffreyhuang1/two-stream-action-recognition.

[20] Implementation of deep dream with pytorch. https://github.com/SherlockLiao/Deep-Dream.

[21] Pytorch neural-transfer tutorial. https://github.com/alexis-jacq/Pytorch-Tutorials/blob/master/Neural_Style.ipynb.

[22] Cyclegan and pix2pix in pytorch. https://github.com/junyanz/pytorch-CycleGAN-and-pix2pix.