MUSICAL STYLE TRANSFER IN SPARSE AUDIO REPRESENTATION DOMAINS

Flor Sanders

Columbia University

ABSTRACT

This project explores efficient low-dimensional techniques for musical style transfer between classical and jazz genres using convolutional autoencoders. The data samples undergo preprocessing and feature extraction through Mel-scaled short time Fourier transform. In total four model architectures are investigated. While twin convolutional autoencoders achieve good self-reconstruction, meaningful style transfer remains a challenge. Introducing twin variational autoencoders enhances style transfer but slightly reduces self-reconstruction quality. The addition of generative adversarial networks (GAN) and adaptive instance normalization improves style transfer, yet the generated audio lacks harmonic quality. Twin variational GAN autoencoders combine variational autoencoders and GANs but exhibit diminished self-reconstruction and produce cartoonish style transfer results.

Index Terms— Autoencoder, Convolutional Neural Network (CNN), Generative Adversarial Networks (GAN), Musical Style Transfer, Variational Autoencoders (VAE)

1. INTRODUCTION

In recent decades, an explosion in terms of data availability as well as processing power has occurred, which has ushered in a new era of data-driven applications and learning-based model development. In this field, sparse and low-dimensional models have emerged as a powerful toolset to efficiently represent and manipulate such data [1]. Music is no exception to this trend, with independent artists being able to publish their art in a free an open way on the world wide web. As such, large collections of musical pieces with open licenses are nowadays available. Furthermore, spoken and musical audio are notorious for being well-structured, which allows for efficient sparse and compressible representation in transform domains by using wavelet or Fourier basis functions [2].

Within the world of music, artists continuously draw inspiration from one another, by covering and adapting existing works. This process often results in a phenomenon called style transfer, in which a music piece is recomposed in a novel style, while the harmonic and structural properties of the original piece are preserved [3]. Due to the general complexity of this task and as many different interpretations of music and musical genres are valid, this is generally a task performed by skilled musicians and composers. Recently, however, breakthroughs have been made in the field of generative deep learning models for other domains, including language and image models [4, 5, 6]. It turns out that these novel techniques translate quite well to the field of audio processing, including audio encoding, audio generation and music generation and style transfer [7, 8, 9].

A major criticism that can be leveled against the aforementioned models is their immense scaled and related computational complexity as well as power consumption. As such, this project explores more efficient low-dimensional techniques that still harness the power of learning models for style transfer between musical pieces. More specifically, instrumental classical and jazz music are chosen as the two genres to recompose.

The remainder of this report is structured as follows. Section 2 outlines the technical fundamentals underlying the project experiments, which are discussed in Section 3. A discussion of the results, as well as a comparison with the state of the art, is provided in Section 4. Finally, Section 5 presents further outlooks and draws conclusions.

2. TECHNICAL APPROACH

Diving into the technical aspects of this paper first requires the construction of a suitable dataset, followed by an investigation of the features to be extracted for further processing. Finally, the fundamentals of convolutional autoencoders, lying at the basis of the presented experiments, is discussed.

2.1. Dataset Construction

In the world of music processing, some well-known datasets are available. One such example is MusicNet, containing a collection of 330 freely licensed musical recordings, including annotations regarding the timing and notes being played [10]. This dataset is adopted for the classical music genre.

For the jazz music genre, no dataset similar to MusicNet exists to the best of the author's knowledge. This immediately presents one of the major challenges regarding the style transfer problem: qualitative labeled datasets are few and far between, making supervised learning impractical. If one decides to adopt an unsupervised learning approach, as is done

further in this paper, significantly more options become available. In this project, we opted to collect freely licensed jazz recordings from the Free Music Archive [11]. The resulting dataset consists of 163 total jazz pieces. The music pieces in its raw form are available on Google Drive for reference[1].

In its raw form, however, audio data is not well structured. The files can be encoded using various formats, its contents can be mono or stereo, the pieces can be of different lengths and normalized to different volumes. As such, preprocessing is required before the collected data is ready to be used for model training. The preprocessing steps are:

1. All pieces are converted to wav format, which encode the raw waveform contents, with only a mono channel and a sample rate of 44,100 Hz.

2. The volume for each waveform is normalized to an average of -20 dB.

3. Each waveform is cut into segments of length 131,072, roughly corresponding to a duration of three seconds.

As a result, we respectively obtain 41,168 segments of classical music and 15,619 segments of jazz music, each with a duration of three seconds. A sample waveform is visualized in Figure 1.

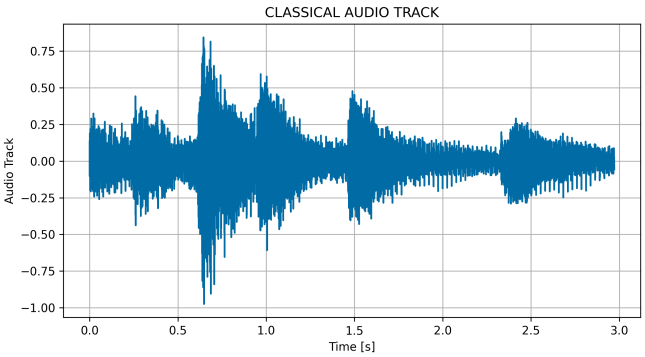

Fig. 1: Music sample waveform.

2.2. Feature Extraction

While direct waveform sequence-to-sequence audio processing is an option, working with extracted spectral or cepstral features is more common in the realm of audio style transfer [12, 13]. The operation typically used for this is the short time fourier transform (STFT), which extracts spectral information for windowed sections of the input audio. In this project, the librosa package for audio and music processing is used to perform such transform extractions [14]. Figure 2 presents the STFT of the sample waveform from Figure 1.

[1] https://drive.google.com/drive/folders/1ExFoqDbWS6UquuIA_TQy2TzNnAwCMUQK?usp=sharing

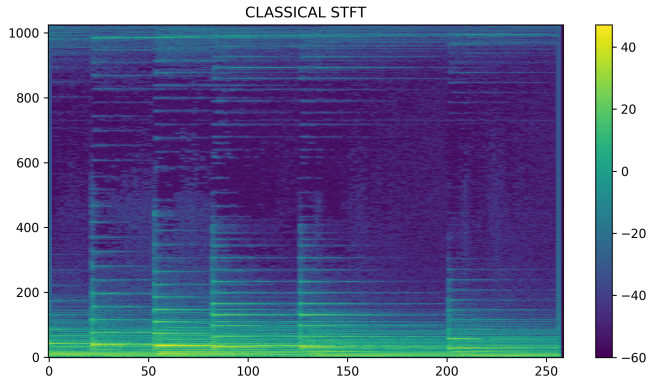

Fig. 2: Music sample STFT.

With 1024 spectral coefficients extracted and 259 temporal windows remaining, however, the dimensionality of this transform is substantially larger than the original waveform at 311,836 unique elements. At the same time, this feature sparse, as a lot of zeros are present. One way to pack more information in such as similar transform is to make use of Mel frequency scaling, where frequencies are spaced on a logarithmic scale that appear equidistant to human hearing [15]. By experimenting with the number of mels, the equivalent to frequencies, one can confirm that 256 is the minimum number that is needed to reconstruct the audio with good, though imperfect, quality. The Mel spectrum for the music sample of Figure 1 is provided in Figure 3.

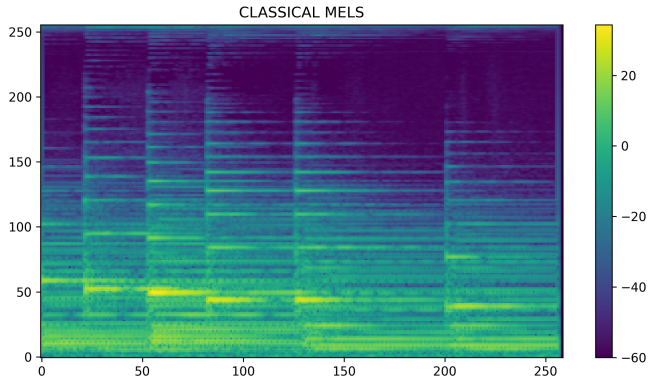

Fig. 3: Music sample Mel spectrum.

As such, qualitative features are extracted from the original dataset. Their dimensionality measures $259 \times 256 = 66,304$, which compresses the original waveform by roughly a factor two. Figure 3 shows that the musical content, both in terms of frequency harmonics and temoral rythm are well-structured in these features. Furthermore, the feature space is still both sparse and low-rank, since the harmonics are generated by a constraint physical process, so further compression of the data should be possible. The final step required before the features can be used for training is normalization, for

which the following procedure is followed. σ_X represents the overall standard deviation of the training data, while $\sigma_{X_2,f}$ represents a spectral scaling based on the previous normalization step for the training data. The effect of this normalization on the mean and standard deviation across the Mel spectrum is shown in Figure 4.

1. $X_1 = X/\sigma_X$

2. $X_2 = \log_1 0(1 + X_1)$

3. $X_n = X_2/\sigma_{X_2,f}$

Finally, in order to reconstruct the audio from these features, the normalization procedure is inverted. After that, the Griffin-Lim algorithm is used for accuracte phase reconstruction, as this information is lost when working with power spectra, implemented in the librosa python library [14, 16]. It should be noted that this operation is quite expensive, taking up the majority of the processing time during inference for our models (cfr. Section 3).

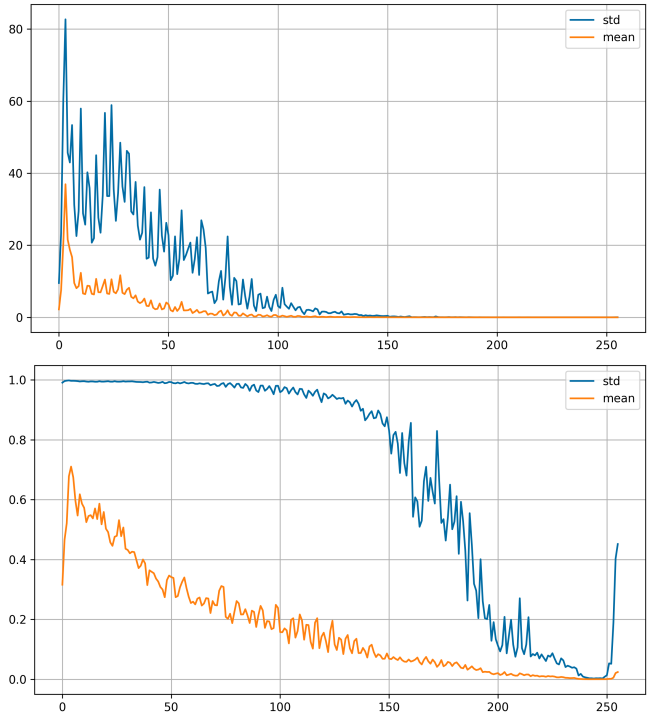

Fig. 4: Distribution of the Mel spectrum before normalization (top) and after normalization (bottom).

2.3. Convolutional Autoencoders

Convolutional neural networks (CNN) lie at the basis of the modern machine learning revival, and have proven to be capable of excellent performance for various tasks in the realm of image processing. Autoencoders are a particularly interesting architecture for CNNs, as they allow for representation learning in an unsupervised manner. The goal of an autoencoder is to learn the parameters for two networks. The encoder transforms the input to a latent space, $h = f(X)$, while the decoder tries to reconstruct the original data from that latent variable $\hat{X} = g(h)$ [17]. The hope for such networks is often for the latent space to capture semantic meaning. In this project, this latent space could represent a low-dimensional manifold which captures the style and content of the musical pieces in a separable manner.

3. EXPERIMENTS

The experiments presented in this project all share a common architecture: the twin autoencoder, visualized in Figure 5. For each of the respective music genres, an encoder-decoder pair is trained, with the goal of obtaining a latent feature space that is structured in such a way to make style transfer between the models possible by swapping the decoders. This is the main challenge of the style transfer problem. A collection of experiment results for each of the architectures discussed below is distributed through Google Drive[2].

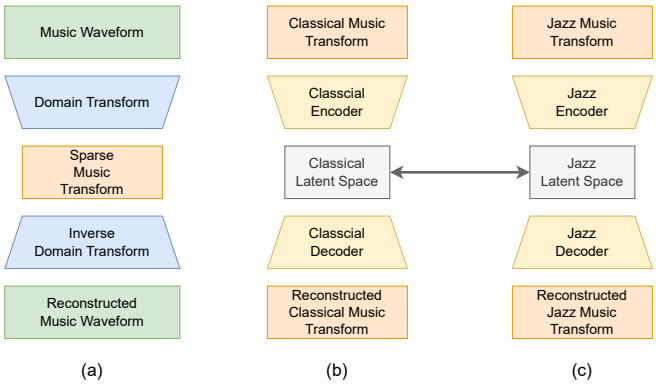

Fig. 5: Style transfer model architecture, showing:
(a) Common domain transform pre-processing step.
(b) Classical music auto-encoder model.
(c) Jazz music auto-encoder model.

[2] https://drive.google.com/drive/folders/1nOpKh44qfV5pXqruawBtPoPVHukpLToh?usp=sharing

3.1. Twin Convolutional Autoencoders

This first experiment adopts the most straightforward approach imaginable given the common architecture. The training procedure consists of four steps, listed below and visualized in Figure 6.

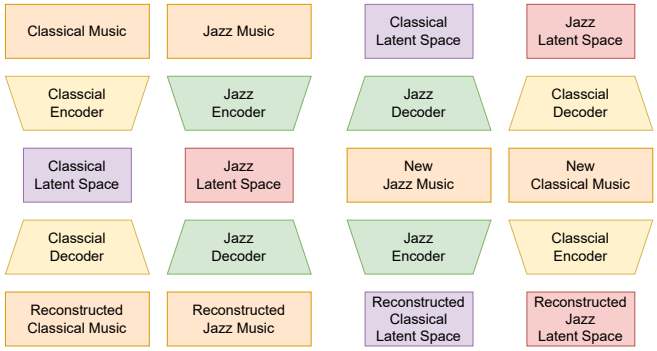

Fig. 6: Twin CNN training strategy.

1. The classical and jazz features X_c, X_j are encoded by their respective encoders, resulting in latent space representations h_c, h_j for both genres.

2. These latent spaces are decoded using the proper decoders, in order to reconstruct the original features $\hat{X}_c$, $\hat{X}_j$.

3. Now, the decoders are switched, in order to generate new sample spectra $X_{j,new}$, $X_{c,new}$.

4. The newly generated spectra are encoded again in order to reconstruct the original latent spaces $\hat{h_c}$, $\hat{h_j}$.

Based on these transformations, two loss contributions can be introduced. First is the self-reconstruction loss, given in Equation 1, which compares the original normalized Mel spectrum with the output of the decoders. Additionally, the latent space reconstruction loss, given in Equation 2, compares the encoded latent spaces with the decoder swapped reconstructions. In this way, both the linkage of the latent spaces and the goal of style transfer is implicitly contained in the training strategy.

$$L_r = \frac{1}{2}\left|X_c - \hat{X}_c\right| + \frac{1}{2}\left|X_j - \hat{X}_j\right| \tag{1}$$

$$L_r = \frac{1}{2}\left|h_c - \hat{h_c}\right| + \frac{1}{2}\left|h_j - \hat{h_j}\right| \tag{2}$$

The models are trained by presenting randomized pairs of classical and jazz samples, feeding these forward through the network as described and computing the loss function. Afterwards, gradient descent is applied and the Adam optimization algorithm, a variant of stochastic gradient descent (SGD) with momentum and parameter scaling, is used to update the model

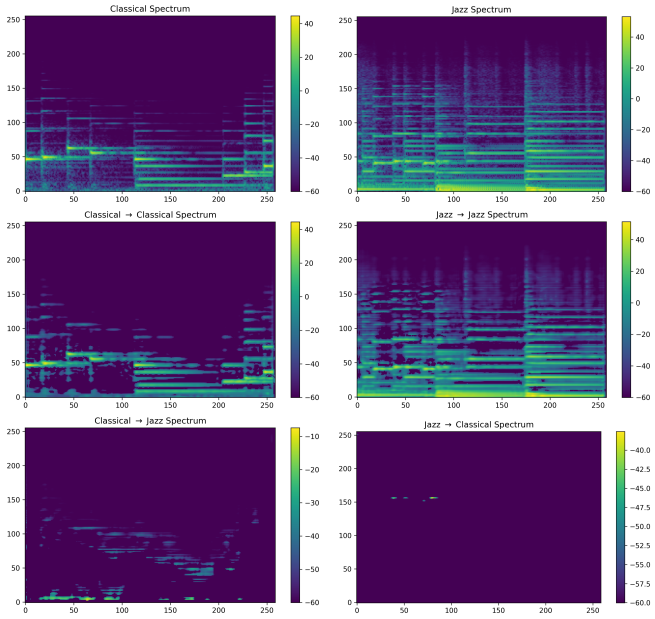

Fig. 7: Twin convolutional autoencoder results. The left and right columns represent classical and jazz inputs. From top to bottom the original spectrum, its self-reconstruction and the style transfer results are given.

parameters. Most of these steps are handled behind the scenes by the Tensorflow framework, used to facilitate implementation in this project [18]. All hyperparameters of the network have meticulously been tuned to obtain optimal performance by manual iterative experimentation. For reference, these are available in the notebooks of the project implementation on Github [3].

The results after training for 50 epochs are visualized in Figure 7. While the self-reconstruction shows good results, with a test-time self-reconstruction loss of 0.05073, the style transfer spectra leave a lot to be desired. Generally they are either empty or contain a few lines that seem entirely unrelated to the original content. The reason for this is that the autoencoder's latent spaces are not as well-behaved as initially hoped.

3.2. Twin Variational Autoencoders

One technique often used to enfore structure on the latent space of autoencoder is the variational autoencoder (VAE) architecture. In this case, the latent space is explicitly interpreted to represent the parameters for a random distribution, commonly mean μ and log-variance $\log(\sigma^2)$ of the normal distribution. Samples z are drawn from this distribution by drawing a sample ζ from a normalized distribution and performing reparametrization as described by Equation 3. An

[3] https://github.com/FlorSanders/AutoStyleTransfer

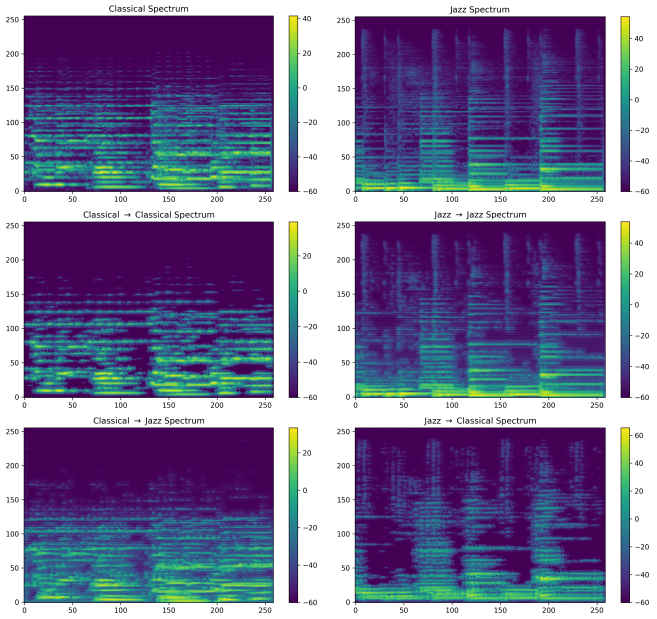

Fig. 8: Twin variational autoencoder results. The left and right columns represent classical and jazz inputs. From top to bottom the original spectrum, its self-reconstruction and the style transfer results are given.

additional loss factor can then be introduced that forces the distribution to its normalized variant. For this purpose, the Kullback-Leibler divergence is introduced, given in Equation 4 [17, 19].

$$z = \sigma\zeta + \mu \tag{3}$$

$$L_{kl} = \frac{1}{2}\left(\sigma^2 + \mu^2 - 1 - \log(\sigma^2)\right) \tag{4}$$

The same training procedure as for Section 3.1 is adopted and the results after 50 epochs of training are presented in Figure 8. The self-reconstruction takes a slight hit compared to the twin convolutional network at a test-loss value of 0.05388, though the results of the style transfer are significantly improved. In the bottom right spectrum of Figure 8, one can notice that a lot of the tones present in the original piece are raised in frequency, which corresponds to the higher energy for the high frequencies in the classical piece. Correspondingly, the bottom left spectrum displays signs of notes smoothing together over time, which could be influenced by the jazz music where the notes are attacked less often. Since the audio quality takes a significant hit for these outputs, however, it is not possible to determine exactly what effect can be attributed to the style transfer and which is because of the degraded quality.

3.3. Twin GAN Autoencoders

While the twin variational autoencoders show signs of style transfer, the results are still a long way of from the final goal. In an effort to improve these, a more direct method of regularizing the style transfer output is investigated. In generative adversarial networks (GANs), an additional neural network is introduced for each autoencoder (generator) that is tasked with discriminating between real inputs and generated samples by predicting the probability that a given sample is original or generated [17, 20]. The twin autoencoders and the discriminator are trained in alternating fashion. The chi-squared GAN loss for the training of the discriminator, given in Equation 5, promotes low probabilities for generated samples and high probabilities for original samples. During optimization of the generator, these priorities are switched, corresponding to Equation 6 [21]. In this way, the twin autoencoders should converge towards a state where the generated samples are comparable in characteristics to the originals they are compared against.

$$\begin{aligned} L_{gan,dis,1} &= \left(\text{prob}\left\{X\right\} - 1 - \frac{1}{N}\sum_{n=1}^{N}\text{prob}\left\{X_{new}\right\}\right)^2 \\ L_{gan,dis,2} &= \left(\text{prob}\left\{X_{new}\right\} + 1 - \frac{1}{N}\sum_{n=1}^{N}\text{prob}\left\{X\right\}\right)^2 \\ L_{gan,dis} &= \frac{1}{2}\left(L_{gan,dis,1} + L_{gan,dis,2}\right) \end{aligned} \tag{5}$$

$$\begin{aligned} L_{gan,gen,1} &= \left(\text{prob}\left\{X\right\} + 1 - \frac{1}{N}\sum_{n=1}^{N}\text{prob}\left\{X_{new}\right\}\right)^2 \\ L_{gan,gen,2} &= \left(\text{prob}\left\{X_{new}\right\} - 1 - \frac{1}{N}\sum_{n=1}^{N}\text{prob}\left\{X\right\}\right)^2 \\ L_{gan,gen} &= \frac{1}{2}\left(L_{gan,gen,1} + L_{gan,gen,2}\right) \end{aligned} \tag{6}$$

One additional method often used for GAN style transfer method is adaptive instance normalization (AdaIN). Here the encoder latent space is explicitly split up into seperate parts for content and style $h = (c, s)$, which are combined as described by Equation 7 before being fed into the decoder. In case of style transfer, the content pieces of the latent space are kept and only the style elements are swapped over. This explicit separation of style and content in the latent space ought to facilitate mixing aspects of both musical pieces in the end result [22].

$$\text{adain}(c, s) = \sigma_s\left(\frac{x - \mu_x}{\sigma_x}\right) + \mu_s \tag{7}$$

The results after 50 epochs of training are provided in Figure 9. In terms of self-reconstruction, this model performs worse than the previous ones presented in this project, with a test-time loss of 0.07116. However, the output of the swapped decoders indicate the presence of style transfer. We can see that for the new jazz spectrum (bottom left), some high frequency content that was previously not present is generated. On the other hand for the new classical spectrum (bottom right), we see that the upper harmonics are suppressed and that multiple attacks on the notes are introduced with respect to the jazz tracks. Unfortunately, however, the model seems to have no concept of harmonic beauty and the resulting audio tracks sound quite nightmarish to the human ear.

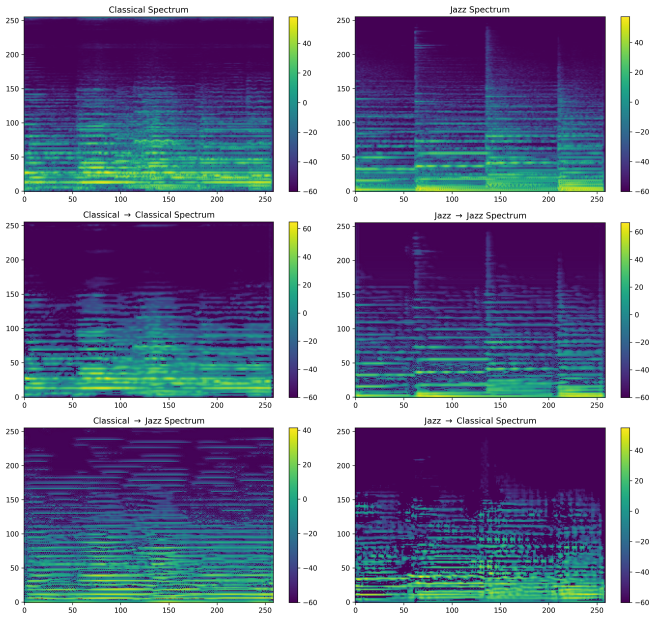

Fig. 9: Twin GAN + AdaIN autoencoder results. The left and right columns represent classical and jazz inputs. From top to bottom the original spectrum, its self-reconstruction and the style transfer results are given.

3.4. Twin Variational GAN Autoencoders

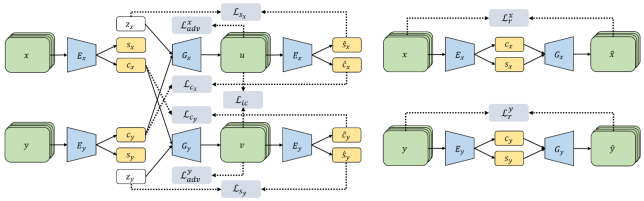

Fig. 10: Twin variational GAN architecture, from [13].

For the final model, the techniques from Sections 3.2 and 3.3 are combined. The separation of the latent space into content and style is maintained. When generating the new samples, however, a style code is drawn from a normal distribution with zero mean and unit variance. With these final changes, we arrive at the architecture proposed in [23] for unsupervised image style transfer and used in [13] in the context of musical style transfer. A visual representation of the architecture is given in Figure 10.

The training process is the same as previously described and the results after 50 epochs are presented in Figure 11. First, the self-reconstruction performance takes a further dive to a test loss of 0.08146. Qualitatively, this is quite apparent as a lot of detail in the spectrum of the classical reconstruction (center left) in Figure 11 are missing. Unfortunately, this reduction in self-reconstruction performance does not correspond to an improvement in style transfer results. While some of the same trends as in Section 3.3 are visible, the spectra look quite cartoon-ish and the resulting audio samples reflect nothing less, being firmly present in the uncanny valley.

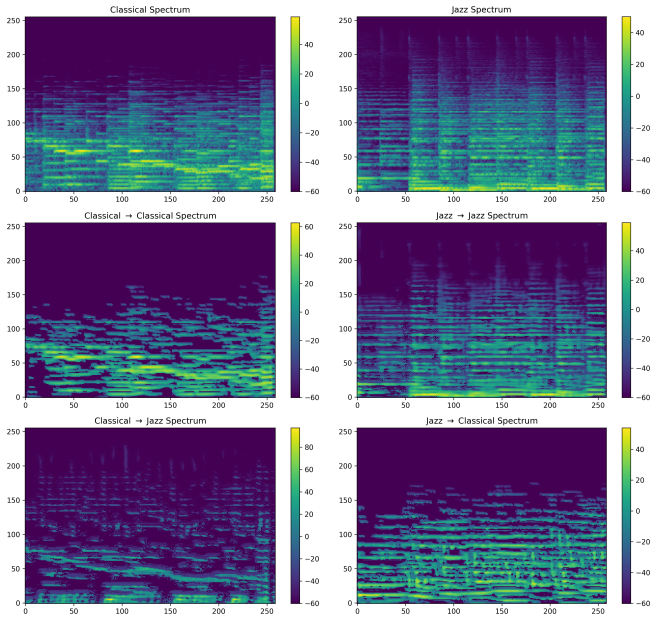

Fig. 11: Twin Variational GAN + AdaIN autoencoder results. The left and right columns represent classical and jazz inputs. From top to bottom the original spectrum, its self-reconstruction and the style transfer results are given.

4. DISCUSSION

Since convincing style transfer between classical and jazz music was not reached in this project, the main conclusion should be that style transfer is a difficult one. However, the presented models display promising behaviour in the direction of the end goal. The twin variational autoencoders architecture from Section 3.2 achieve a frequency shift reflecting the different source pieces. Moreover, the twin GAN autoencoders from Section 3.3 achieve style transfer spectra

that are somewhat convincing to the human eye, though not to the human ear after once the audio signal is reconstructed.

One interesting question is whether there is a fundamental limit on the performance that can be expected when using architectures, which are still relatively simple compared to the state of the art today [9]. For this, we turn to the established literature. As referenced before, the inspiration for the architecture in Section 3.4 was drawn from [13]. In this work, the architecture from Figure 10 is used for style transfer between instrumental guitar and piano pieces, using multiple extraced features beside the Mel spectrum. This task is relatively more simple than the one attempted in this project, since only a single instrument is present in each music track and style transfer is attempted between instruments rather than musical genres. The results, compared to this work, are nonetheless impressive. The outcome samples presented in the paper display style transfer between the instruments that is mostly convincing to the human ear, save some small artifacts.

Compared to the present day state of the art however, represented by [9] developed at Meta, the capabilities presented in [13] are quite unsophisticated. The model in [9] combines audio input with a textual prompt in order to inform the style transfer. The results are extremely convincing and (almost) competitive with human performance. The architecture that enables this is based on the more recently developed transformer model, with the audio being tokenized by the neural network encoder presented in [7]. Naturally, the number of parameters of this network is orders of magnitude larger than the models presented in [13] or developed in this work. This would indicate that to achieve convincing multi-genre musical style transfer a different scale of network and compute is needed beyond the novel architecture.

5. CONCLUSION

In conclusion, this paper delves into the challenging task of musical style transfer between classical and jazz genres using efficient low-dimensional techniques and convolutional autoencoders. Despite achieving notable progress in self-reconstruction, meaningful style transfer remains elusive, indicating the intricacies of capturing the nuanced stylistic elements of diverse musical genres. The exploration of twin variational autoencoders, generative adversarial networks (GANs), and adaptive instance normalization reveals improvements in style transfer, yet challenges persist, notably in maintaining harmonic quality in the generated audio. The study underscores the complexity of achieving convincing multi-genre musical style transfer with current architectures and suggests the potential necessity of more advanced models, such as transformer-based architectures, for enhanced performance in this domain. As data-driven applications and learning-based models continue to evolve, future research may unlock innovative approaches to address these challenges and push the boundaries of musical style transfer.

6. REFERENCES

[1] John Wright and Yi Ma, *High-Dimensional Data Analysis with Low-Dimensional Models: Principles, Computation, and Applications*, Cambridge University Press, 2022.

[2] Andreas Spanias, Ted Painter, and Venkatraman Atti, *Audio Signal Processing and Coding*, Wiley, Dec. 2005.

[3] Shuqi Dai, Zheng Zhang, and Gus G. Xia, "Music style transfer: A position paper," 2018.

[4] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto, "Stanford alpaca: An instruction-following llama model," `https://github.com/tatsu-lab/stanford_alpaca`, 2023.

[5] OpenAI, "Gpt-4 technical report," 2023.

[6] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever, "Zero-shot text-to-image generation," 2021.

[7] Alexandre Défossez, Jade Copet, Gabriel Synnaeve, and Yossi Adi, "High fidelity neural audio compression," 2022.

[8] Felix Kreuk, Gabriel Synnaeve, Adam Polyak, Uriel Singer, Alexandre Défossez, Jade Copet, Devi Parikh, Yaniv Taigman, and Yossi Adi, "Audiogen: Textually guided audio generation," 2023.

[9] Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez, "Simple and controllable music generation," 2023.

[10] John Thickstun, Zaid Harchaoui, and Sham M. Kakade, "Musicnet," 2016.

[11] "Royalty free music - free music archive," `https://freemusicarchive.org`, 2023.

[12] Marco Pasini, "Melgan-vc: Voice conversion and audio style transfer on arbitrarily long samples using spectrograms," 2019.

[13] Chien-Yu Lu, Min-Xin Xue, Chia-Che Chang, Che-Rung Lee, and Li Su, "Play as you like: Timbre-enhanced multi-modal music style transfer," 2018.

[14] librosa.org, "Librosa: A python package for music and audio analysis," `https://github.com/librosa/librosa`, 2023.

[15] S. S. Stevens, J. Volkmann, and E. B. Newman, "A Scale for the Measurement of the Psychological Magnitude Pitch," *The Journal of the Acoustical Society of America*, vol. 8, no. 3, pp. 185–190, 06 2005.

[16] Nathanaël Perraudin, Peter Balazs, and Peter L. Søndergaard, "A fast griffin-lim algorithm," in *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2013, pp. 1–4.

[17] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep Learning*, MIT Press, 2016.

[18] tensorflow.org, "Tensorflow: An open source machine learning framework for everyone," `https://github.com/tensorflow/tensorflow`, 2023.

[19] Diederik P Kingma and Max Welling, "Auto-encoding variational bayes," 2022.

[20] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial networks," 2014.

[21] Xudong Mao, Qing Li, Haoran Xie, Raymond Y. K. Lau, Zhen Wang, and Stephen Paul Smolley, "Least squares generative adversarial networks," 2017.

[22] Xun Huang and Serge Belongie, "Arbitrary style transfer in real-time with adaptive instance normalization," 2017.

[23] Xun Huang, Ming-Yu Liu, Serge Belongie, and Jan Kautz, "Multimodal unsupervised image-to-image translation," 2018.