Computer Evaluation of Musical Timbre Transfer on Drum Tracks

by

Keon Ju Lee

A Thesis Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science

in the Department of Computer Science

Computer Evaluation of Musical Timbre Transfer on Drum Tracks

by

Keon Ju Lee

Supervisory Committee

Dr. George Tzanetakis, Supervisor
(Department of Computer Science)

Dr. W. Andrew Schloss, Outside Member
(Department of Music)

ABSTRACT

Musical timbre transfer is the task of re-rendering the musical content of a given source using the rendering style of a target sound. The source keeps its musical content, e.g., pitch, microtiming, orchestration, and syncopation. I specifically focus on the task of transferring the style of percussive patterns extracted from polyphonic audio using a MelGAN-VC model [57] by training acoustic properties for each genre. Evaluating audio style transfer is challenging and typically requires user studies. An analytical methodology based on supervised and unsupervised learning including visualization for evaluating musical timbre transfer is proposed. The proposed methodology is used to evaluate the MelGAN-VC model for musical timbre transfer of drum tracks. The method uses audio features to analyze results of the timbre transfer based on classification probability from Random Forest classifier. And K-means algorithm can classify unlabeled instances using audio features and style-transformed results are visualized by t-SNE dimensionality reduction technique, which is helpful for interpreting relations between musical genres and comparing results from the Random Forest classifier.

Contents

List of Figures

ACKNOWLEDGEMENTS

I would like to thank:

My Parents, for always supporting me with unconditional love.

Dr. George Tzanetakis, including my supervisory committee for mentoring, support, encouragement, and patience.

University of Victoria, for funding me with a Scholarship and protecting me from COVID-19, lucky to be here in these unprecedented times.

And also special thanks to **Dr. Philippe Pasquier** for giving me constructive feedback.

Hopefully, my future self will live a better life than the present and be winning more battles with myself. Life is a matter of managing and caring myself in every aspect.

by Keon

DEDICATION

to people who care about my research.

ACRONYMS

ADT Automatic Drum Transcription

AFE Audio Feature Extraction

AFS Audio Feature Selection

AR Auto-Regressive

AI Artificial Intelligence

AST Audio Style Transfer

ANN Artificial Neural Networks

ANOVA-F Analysis Of Variance F-value

CNN Convolutional Neural Networks

CQT Constant-Q Transform

DL Deep Learning

DNN Deep Neural Networks

DGC Drum Genre Classification

DCGAN Deep Convolutional Generative Adversarial Network

DT Decision Trees

D Discriminator

GBM Gradient Boosting Algorithm

GUI Graphical User Interface

G Generator

GANs Generative Adversarial Networks

HPSS Harmonic Percussive Source Separation

HPS Harmonic Percussive Separation

IG Information Gain

IRM Interquartile Range Method

LR Logistic Regression

MTT Musical Timbre Transfer

ML Machine Learning

MIR Music Information Retrieval

MIDI Musical Instrument Digital Interface

MGC Musical Genre Classification

MelGAN Mel-spectrogram Generative Adversarial Networks

MLP Multi Layer Perceptrons

MI Mutual Information

MGC Music Genre Classification

MFCCs Mel-Frequency Cepstral Coefficients

NAR Non-autoregressive (NAR)

NST Neural Style Transfer

PGC Percussive Genre Classification

PCA Principal Component Analysis

RF Random Forest

RFE Recursive Feature Elimination

SL Supervised Learning

SST Symbolic Style Transfer

S Siamese network

SNN Siamese Neural Networks

SVMs Support Vector Machines

STFT Short-Time Fourier Transform

TraVeLGAN Transformation Vector Learning GAN

t-SNE t-distributed Stochastic Neighbor Embedding

TTS Text-To-Speech

UL Unsupervised Learning

VC Voice Conversion

VAEs Variational Autoencoders

Chapter 1

Introduction

Music Information Retrieval (MIR) research is an interdisciplinary area combining with computer science and music. MIR includes aspects of machine learning, artificial intelligence, signal processing, computer music, and musicology. Typical MIR tasks can be summarized as follows: music classification or recommendation systems, audio feature extraction including manipulation, audio source separation, automatic music transcription, and music generative systems. Audio Style Transfer (AST) using Generative Adversarial Networks (GANs) in music domain can also be classified as a MIR task, because it accompanies audio manipulation, audio feature extraction, and audio texture synthesis using deep learning. AST algorithms train acoustic properties and timbral textures from arbitrary length audio that has distinctive characteristics depending on the musical genre. The algorithms can transfer

the acoustic properties from an original genre to a target genre. As a result, the output has the style of the target music genre based on the original genre track. And its computational evaluation methodology is an important topic to analyze the output, since the evaluation in most AST papers simply compares spectrograms of each case: source, target, and style-transformed audio, without analytical methods.

1.1 Motivation

The motivation for this research is to explore how GANs can be used to perform music style transfer in the audio domain. More specifically, I focus on two problems: 1) experimenting with the MelGAN-VC algorithm for transforming drum parts of arbitrary polyphonic audio from one music genre to another musical genre, 2) proposing a computer-based methodology based on supervised and unsupervised learning that can be used to evaluate musical timbre transfer.

The motivation for this research starts from classification for monophonic drum samples which are regarded as a relatively simple problem that could be solved by traditional machine learning algorithms, such as decision trees (DT), random forest (RF), linear support vector machines (Linear SVMs) and multi-layer perceptrons (MLP) with more than 95% accuracy. From the result, I conclude it is overly simple

to classify short-length monophonic drum sounds, such as each drum sample for a hi-hat, bass, and snare, respectively. After the trial of the previous problem, automatic drum transcription (ADT) [79] is the next step in my experiments. The state-of-the-art approach for ADT is OaF drums[1], which considers note velocity level for the first time in ADT research area and data sets so it enhances the perceptual quality of drums for listeners, invented by Google Magenta team [12]. Figure

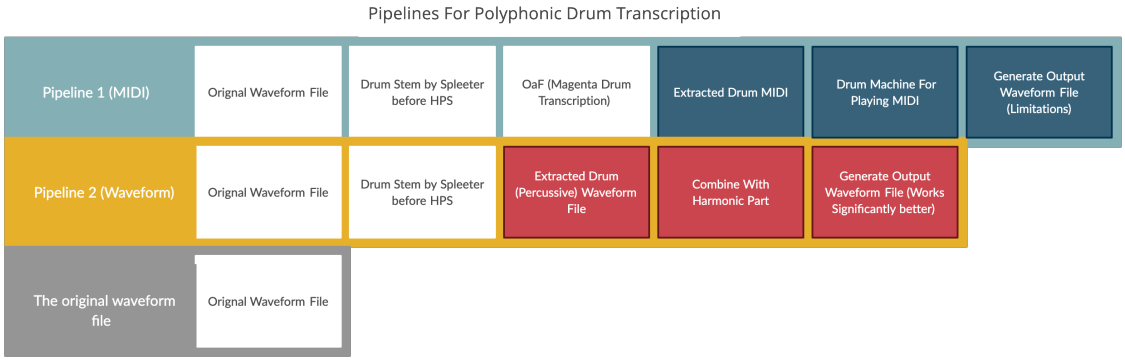


Figure 1.1: Pipelines for polyphonic drum transcription.

1.1 is showing implemented experiments to compare methods. Audio source separation tool called Spleeter[2] [33] and harmonic percussive source separation (HPSS[3]) are used for both methods for pre-processing data sets to extract drum stems from the original audio waveform files. The first method is using extracted Musical Instrument Digital Interface (MIDI) to generate drum audio; on the other hand, the second

[1]https://magenta.tensorflow.org/oaf-drums
[2]https://github.com/deezer/spleeter
[3]https://librosa.org/librosa_gallery/auto_examples/plot_hprss.html

approach does not include any symbolic or MIDI processing, so it is computationally inexpensive and sounds more similar as opposed to the original waveform files because of three main limitations on traditional drum transcription models. The first limitation is that the ADT model is only working well with drum-only data sets and still mostly focuses on transcribing snare, hi-hat and bass (kick) drums, although even these three classes can not be perfectly transcribed by the state-of-the-art method, such as Oaf drums. Another shortcoming is ADT models are not able to detect differences between drum articulations. According to the Superior Drummer 3 software[4], snare drum articulations can be classified as 6 categories: center, off-center, edge, rim-shot, cross-stick (or side-stick) and rim-only. And the last downside is a symbolic approach requires extra steps to obtain an output, because MIDI data should be pre-processed.

I decide to work on drum track style transfer using MelGAN-VC [57] based on a wav-to-wav approach in the audio domain. To the best of our knowledge, there is no well-established quantitative evaluation methodologies for neural audio (music) style transfer, although there is a good example for evaluating style imitation corpora [21] and quantifying musical style in symbolic domain [20]. Therefore, analytical evaluation pipelines for audio style transfer are proposed as shown

[4]https://www.toontrack.com/product/superior-drummer-3/

in Figure 4.1. In most music generative systems, it is ideal to have human judgements who could tell the difference between music genres easily and evaluate the results; however, it is also helpful to have computational evaluation methods, since it is more efficient and easier to integrate with a creative AI system. In addition, the subjectivity of music genre makes human listeners evaluation more challenging.

1.2 Problem

A neural algorithm of artistic style [24] can be applied to 2-dimensional representation of audio. For example, Figure 1.2 illustrates content image and style (or target) reference image for each audio file. Each audio contains different timbre information. And the neural style transfer algorithm [5] is implemented with the content and the style reference image. The algorithm blends them together and the output may look similar to the content image, but it is painted with the style of reference (or target) image, as shown in Figure 1.3. Thus, the stylized image for audio has the mixed version of timbre information between the content audio and the reference audio. For the example of musical timbre transfer, Timbre-Enhanced Multi-Modal Music Style Transfer [45] is implemented and the model can transfer music pieces to many

[5] https://www.tensorflow.org/tutorials/generative/style_transfer

pieces in another style. And as another example, modulated variational auto-encoders (VAEs) are introduced for many-to-many musical timbre transfer [3] and the model provides a single architecture, which can perform many-to-many transfer (improved from one-to-one and one-to-many architectures), including controlling parameters during training.

I study the AST problem which transfer acoustic properties from

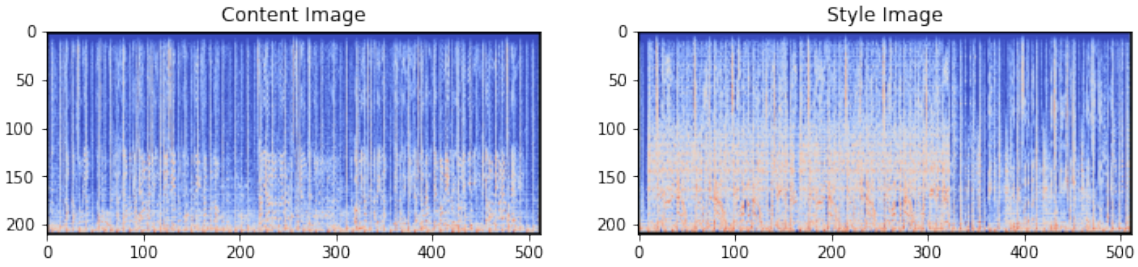


Figure 1.2: Content and style audio for neural style transfer.

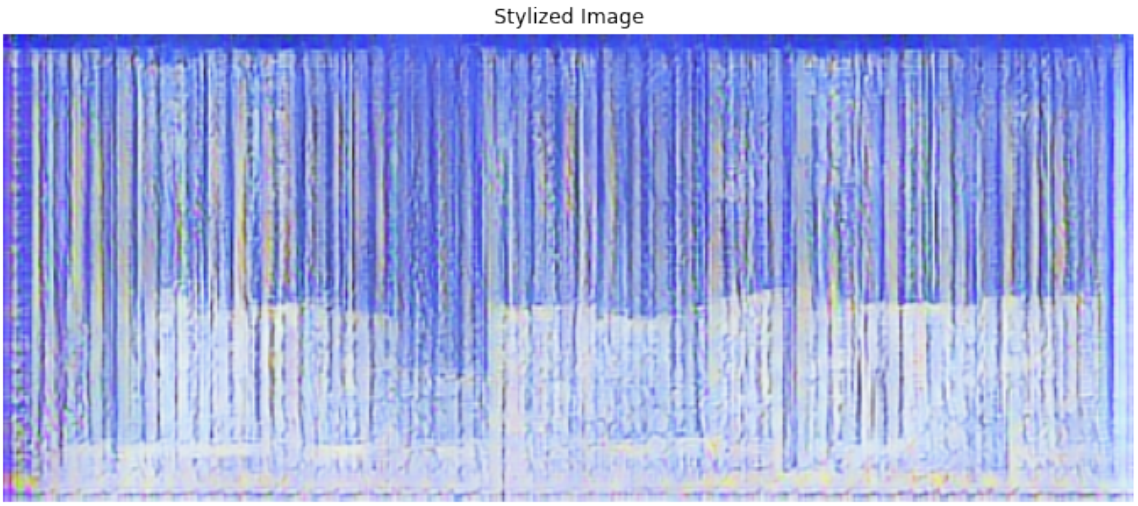


Figure 1.3: Stylized audio by neural style transfer algorithm [24].

hip-hop drum tracks to metal drum tracks, and vice versa. Tempo-

ral properties is not considered in this case and all drum tracks are extracted by source separation software on original polyphonic music tracks. MelGAN-VC model [57] is used to achieve the musical timbre transfer (MTT) on drum tracks. For the computational evaluation, audio features are extracted from drum parts and analyzed based on the musical genre. Thus, each genre has different characteristics and can be classified by genre using machine learning algorithms, such as supervised and unsupervised learning including visualization. This task could be called percussive genre classification. And the output of the MTT is evaluated applying the percussive genre classification technique. Research contributions to the MTT are:

1. Experiments using MelGAN-VC [57] for musical timbre transfer of drum tracks.

2. Pipelines for quantitative evaluation of musical timbre transfer (Figure 4.1).

3. Exploration of audio feature importance for evaluating musical timbre transfer on drum tracks.

4. Evaluation for musical timbre transfer of drum tracks using supervised and unsupervised learning with visualization.

1.3 Organization

Chapter 1 explains motivation for this research and addresses problems for musical timbre transfer.

Chapter 2 describes background knowledge, including previous works for neural style transfer, of musical timbre transfer, and its computational evaluation methods.

Chapter 3 gives explanations about the purpose, procedure, analysis of musical timbre transfer.

Chapter 4 provides details about evaluation methods for musical timbre transfer. The details include audio feature extraction and selection. And evaluation pipelines using supervised and unsupervised learning are proposed with results accompanying visuals and plots.

Chapter 5 contains conclusions, limitations and directions for future research.

Chapter 2

Related Work

Background and previous work are discussed in this chapter. For the background, helpful concepts are covered to understand the previous work for Neural Style Transfer (NST) and tackled problems in musical timbre transfer (MTT) research. After that, previous works for NST are tackled specifically for audio style transfer (AST) and symbolic style transfer (SST) in music domain.

2.1 Background

Background and concepts for understanding the research problem are covered as following.

2.1.1 Generative Adversarial Networks (GANs)

GANs is a deep learning (DL) algorithm and mostly using two neural networks. The concept of GANs was invented by Ian J. Goodfellow, et al [27] at the university of Montreal in 2014; however, the initial version of the GANs architecture was very challenging to train and not stable enough to employ in real applications. Thus, the standardized version of the GANs architecture called DCGAN (Deep Convolutional Generative Adversarial Network) was introduced by Alec Radford, et al [60]. Most GANs architecture are directly or indirectly based on the DCGAN. Machine Learning (ML) algorithms can be classified into two categories: Supervised Learning (SL) and Unsupervised Learning (UL). When training SL algorithms, models have both input data (X) and output labels (Y) so the training data initially contain the ground truth Y, corresponding to X. When it comes to UL algorithms, models predict output labels (Y) based on the input data (X), without having the ground truth (Y). Thus, the UL algorithms tend to more dependent on the training data or the input, compared to the SL algorithms. GANs can be considered as a UL task, since the model generates data based on the training data without labels; however, this framework also translates the UL problem as a SL problem by using an adversarial relation between two neural networks. One of the networks is generating

artificial data based on the training data and the other network is classifying the artificial data whether the generated instance is fake or not. In this context, GANs is using the SL concept as a loss to improve the whole architecture performance. This framework is to predict generative models by using an adversarial relation between a generator (G) and a discriminator (D), so both G and D should be trained simultaneously to take advantage of their adversarial process, which is similar to a minimax two-player game problem [26]. In this case, there are a maximizer and a minimizer. The maximizer can be considered D and the minimizer translated to G. The D is trained for maximizing probabilities classifying synthetic data from the G. And the G is to generate realistic synthetic data to deceive the D with training instances. Thus, the D is for minimizing the difference between the training samples and synthetic instances generated by G. In other words, the nature of adversarial process is the backbone of GANs. The main purpose for using this algorithm is to generate data sets, such as audio, images, videos, speech, MIDI, etc. Deepfake [78] is a noticeable example for GANs. Deepfakes are synthetic or artificial data, including videos and images based on existing media. It is a powerful and controversial technique to generate fake videos and images of public figures, since this tool can potentially lead to spreading false information to the public.

Generator

The role of a generator model is that it takes a random input vector from training data sets and generates synthetic data based on that. In the case of music domain, GANs can learn latent variables, or hidden variables in music data which can not be observed directly by humans. The latent variables can be regarded as compressed data distribution from the training data. Thus, the GANs architecture is able to generate new synthetic data based on the statistically learned music latent variables from the training data. For example, Google Magenta DDSP [19] can be an efficient tool to explore and analyze latent variables. There is a specific example for exploring latent timbre space for synthesis [67].

Discriminator

The role of a discriminator model is that it takes the synthetic data from the generator as input. And the model implements binary classification for the synthetic data whether it is real or fake. If the generated data is real, the classification accuracy is increased and it denotes the generator model is able to produce plausible examples to confuse the discriminator. On the other hand, if the synthetic data is classified fake, the classification accuracy is decreased and loss functions in GANs try to improve the poor performance of the generator. And iterations of

the GANs training process should be repetitive, until the model can generate reliable data from the generator.

Deep Convolutional Generative Adversarial Network (DCGAN)

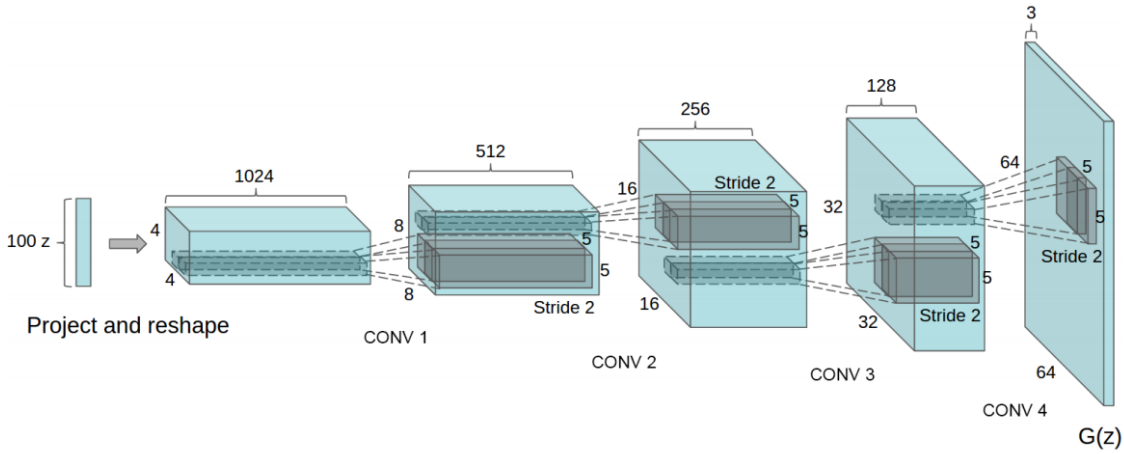


Figure 2.1: The diagram of DCGAN generator from the original paper [60].

DCGAN [60] is a relatively stable version of the original GAN architecture [27], which was introduced in 2014. For this architecture, the generator (shown in Figure 2.1) uses 2-D transposed convolutional layers for upsampling to generate an image based on a random seed as input. First, it takes the seed as input and then typically it reshapes the shape of the input. After that, it upsamples with four layers of convolutional neural networks (CNNs). When it comes to the discriminator, it is a CNN-based image classifier to classify the synthetic image from the generator as fake (negative values) or real (positive values).

The whole DCGAN architecture is trained by the adversarial process between the generator loss and the discriminator loss.

2.1.2 WaveGAN vs MelGAN

WaveGAN

The idea of WaveGAN architecture was devised by the paper titled *Adversarial Audio Synthesis* written by Donahue et al in 2018 [18]. WaveGAN makes possible to train audio signals including temporal information with GANs. Before the advent of WaveGAN, GANs architectures were mostly based on DCGAN [60] and applied to generate images, which are considered as 2-dimensional (2-D) data. The WaveGAN is

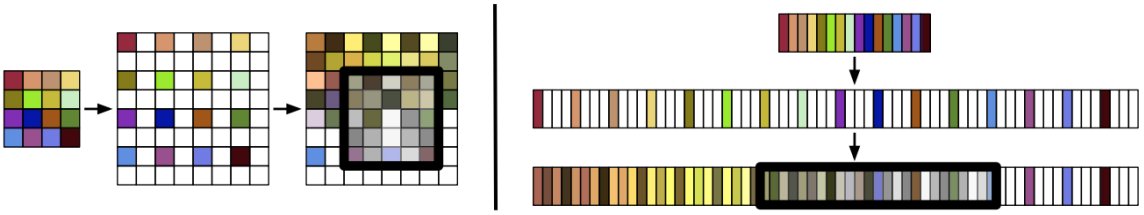


Figure 2.2: Descriptions of DCGAN convolution (left) and WaveGAN convolution operation (right) from the original paper [18].

able to convert 2-D data to 1-D data, including temporal information of audio signals, because time information is a focal point in raw audio signals, unlike images. In other words, the WaveGAN model is based on the DCGAN, but it produces raw audio signals. For example, the

DCGAN model includes calculations of the convolution, which treat 5 * 5 size of the image as 2-D arrays. On the contrary, the WaveGAN model operates the transposed convolution by using the 1-D filter to upscale 5 * 5 shape into length-25, as shown in Figure 2.2. In addition, the WaveGAN uses "Phase Shuffle" technique to avoid pitched noise artifacts that can reduce audio quality and interrupt GAN training, when using standard transposed convolution. The phase shuffle operation alters the phase of each layer of the feature map by the range of -n and +n samples. This technique is useful to improve discriminator's performance by adding some randomness in the procedure and the generator is able to generate higher quality audio signals.

MelGAN

MelGAN [40] was first initiated by Lyrebird AI in 2019 and inspired by the concept of WaveGAN. WaveGAN proved generating coherent raw audio waveform files is challenging to accomplish using GANs. With the MelGAN architecture (shown in Figure 2.3), it is possible to obtain coherent raw audio waveforms with high-quality, although modelling raw audio still has limitations, such as random noise is included with the output, and massive training data sets are required. MelGAN is a non-autoregressive (NAR) model, so the model is computation-

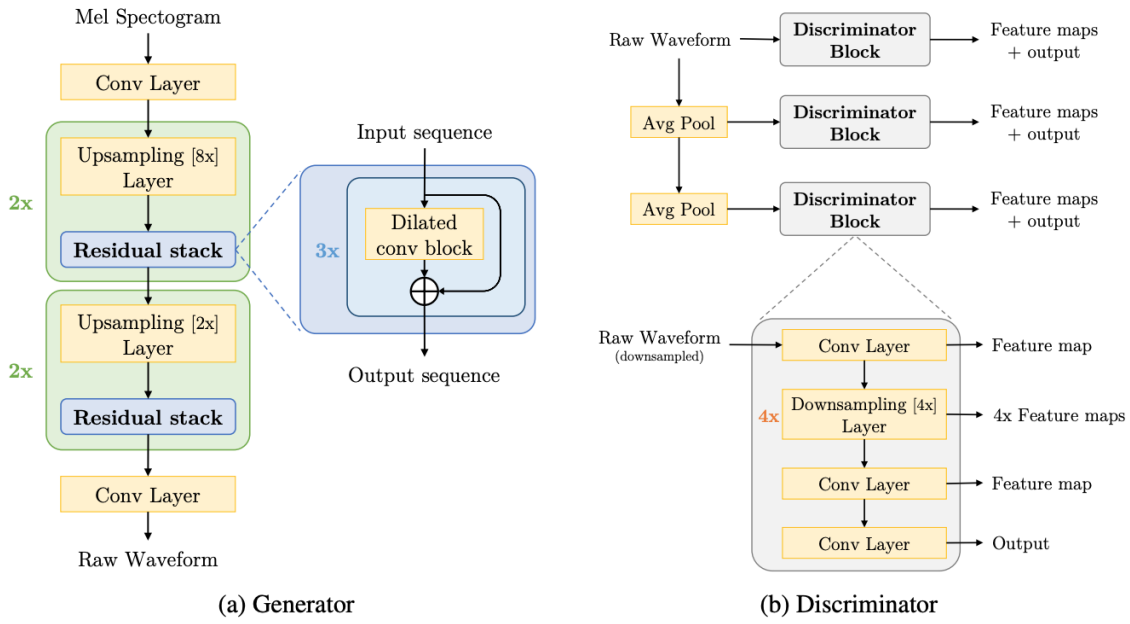


Figure 2.3: MelGAN Architecture, (a) Generator and (b) Discriminator, from the original MelGAN paper [40].

ally efficient, which makes it possible to train in real-time applications. NAR models [61] generate a sequence of data in parallel; however, autoregressive (AR) models are based on the time-series, which consider previous time steps to decide the next step. MelGAN architecture is robust enough to be deployed for designing a Text-To-Speech (TTS) system, because it is the first model that can successfully convert mel-spectrograms (2D-image representation for raw audio signals) to speech from text, without using perceptual loss functions. The TTS system with MelGAN consists of two procedures: 1) the model designs mel-spectrograms conditional on text, 2) the model generates raw audio signals conditional on the mel-spectrograms from the first procedure.

The MelGAN is not limited to generate a single raw waveform, and this model does not require causal dependency from previous audio signals because of the characteristic of the NAR model. For the architecture, it uses three discriminators to train based on the widow size and dilated covolutional networks are used to make the progress parallelizable.

2.1.3 Siamese Neural Networks (SNN)

The concept of Siamese Neural Networks (SNN) was introduced for the first time in 1993 [7]. SNN is also called as a twin neural network because of its architectural characteristic. It is comprised of two identical neural networks. The twin neural networks have same weights and each network accepts each input, respectively. Before the output layer of this network, a contrastive loss function is used to calculate the similarity between two inputs as a part of the optimization process.

Transformation Vector Learning GAN (TraVeLGAN)

Transformation Vector Learning GAN (TraVeLGAN) [2] is the fundamental architecture for MelGAN-VC [57] architecture: generators, discriminators, and siamese networks. The TraVeLGAN was introduced to solve CycleGAN's [82] limitations. CycleGAN only uses generators and discriminators without siamese networks. It is widely used for solving image style transfer problems using a pixel-wise approach

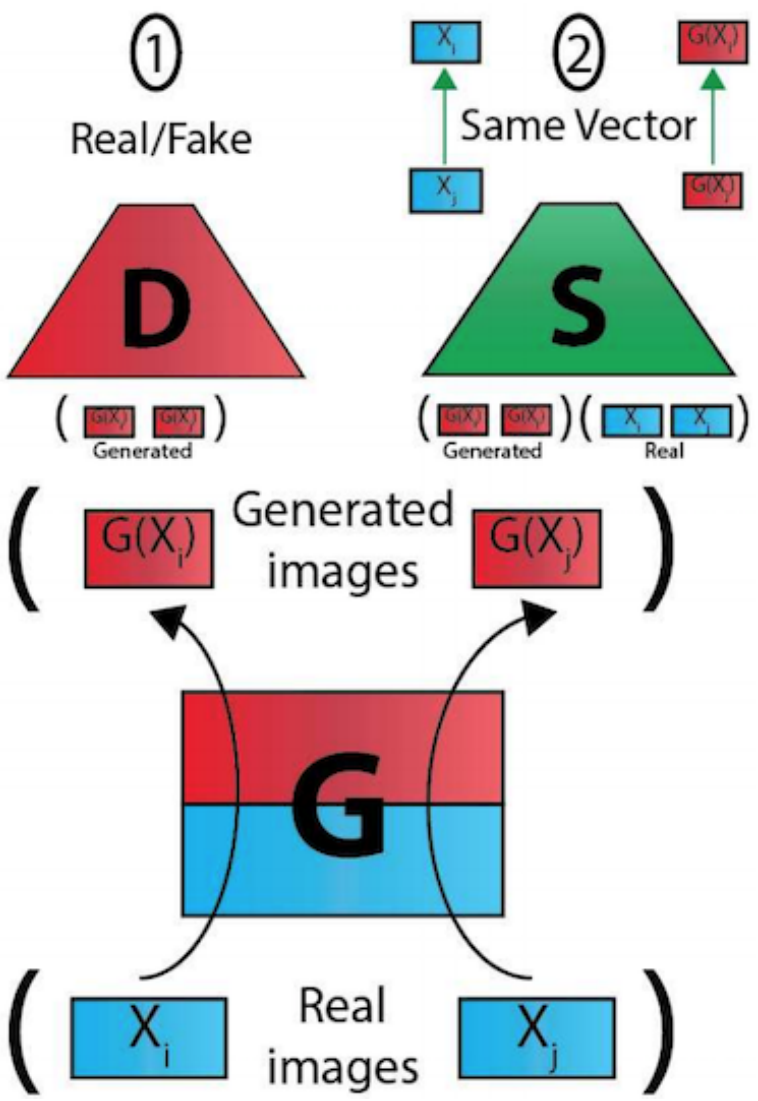


Figure 2.4: TraVeLGAN architecture from the original TraVeLGAN paper [2].

with cycle consistency. The main limitation of the CycleGAN is that it shows lower performance for transferring diverse domains, not similar domains, because it cannot preserve high-level semantic differences [80, 25, 5], such as shapes, types of specific objects, and composition, as opposed to low-level pixel differences, e.g., color, resolution, and lines.

On the other hand, TraVeLGAN works better and can keep content information to solve an image style transfer problem in diverse domains. In other words, the TraVeLGAN architecture can learn and use high-level semantic differences for style-transferring images, when there are significant differences between a source image and a target image. As an audio style transfer example, each musical genre has significant differences when both audio waveforms are represented as spectrograms, and this problem is considered diverse domains, so it is challenging to use CycleGAN to implement the audio style transfer from one genre to another genre; however, TraVeLGAN works better in this case, because it can keep content information from the specific musical genre from the spectrogram. In this scenario, the main role of siamese networks is to preserve vectorized semantic information from the spectrogram and train for generating images from real images (as shown in Figure 2.4) for implementing audio style transfer with a testing instance.

Traditional Deep Learning vs One-shot Learning

Conventional Deep Learning (DL) requires massive pre-processed data sets for training models, and it is the biggest criticism on DL algorithms, because a large amount of refined and labeled data sets are not available in most real world applications. When it comes to one-shot

learning, it reduces the amount of training data sets significantly, and this concept is more similar to human's learning experience. As an example, children learn new concepts without studying large amount of labeled data sets, especially when they are in early childhood in their cognitive development. They could learn new words, and identify objects with a few examples or even one example, and sometimes without any previous instances (they could make an inference based on similar concepts learnt) [81]. When it comes to the application of SNN, the siamese network can be applied to achieve one shot image recognition [36]. One-shot learning models are able to recognize a random instance based on the single observation of the target label. Thus, the models do not need to require training with massive images, but the performance of the models are as high as traditional DL algorithms. There are variations with one-shot learning. K-shot learning or few-shot learning is one of them. K-shot learning model recognizes the "K" number of instances from the target class during training the DL model. On the other hand, zero-shot learning is the different case that the DL model cannot observe any instances from the target label during the training.

2.1.4 Audio Feature Extraction (AFE)

Extracting and processing audio features are critical to analyze audio waveforms in music information retrieval (MIR), because most MIR systems are built based on the low-level audio features. As an example, Tzanetakis and Cook used timbral information, pitch, and rhythmic features for the audio genre classification [71]. Peeters extracted spectral rhythmic features to specifically implement the rhythm classification [58]. In addition, AFE technique was used even for the problem of scene classification by analyzing audio features from sounds in the specific scene content of a video sequence [43], because analyzing audio from the scene was computationally more efficient, compared to analysis of images or videos. There are a number of AFE toolboxes [55] that MIR researchers use, such as Essentia [4], Librosa [50], MIR Toolbox [41], jAudio [51], Marsyas [70], etc.

Essentia

Essentia is the most comprehensive MIR toolboxes writeen in C++ language including Python binding. This library covers all features in MPEG [48] and can extract the highest numbers of features, as shown in Figure 2.5.

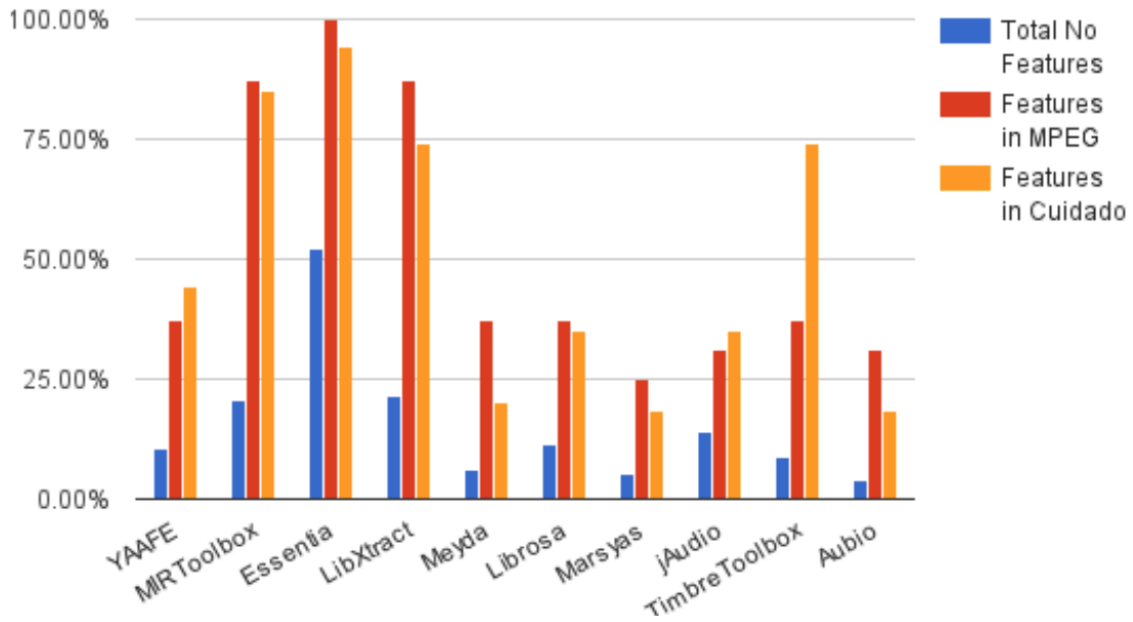


Figure 2.5: Percentage coverage of multiple feature sets depending on the audio feature extraction toolbox, from the paper by Moffat et al [55].

Librosa

Librosa is convenient to use and pre-processs data sets, especially for Python users, when they are experimenting with machine learning algorithms, although it requires longer computation time, compared to the C++ built library: Essentia.

MIR Toolbox

MIR toolbox is used in Matlab and this API can extract a large amount of features. It includes pre-processing, classification, and clustering. Moreover, distance matrices and audio similarity can be extracted. It

can be worked in Matlab and Python environment.

jAudio

jAudio works with Java and it is based on Graphical User Interface (GUI) with a standalone application. Thus, data mining software called Weka [31] can be compatible easily with this library.

Marsyas

Marsyas is the C++ framework including Python binding, which can be easily applied to real-time applications. In addition, Marsyas tools can be used as Vamp plug-ins in Sonic visualiser [13]. And this framework includes stereo panning features [73].

Applications Of Audio Features

Applications of audio features can be summarized as follows [55]:

1. Data Mining [42].
2. Data Classification [52].
3. Similarity Measures [30].
4. Evaluating Synthesis Techniques [32].
5. Feature-based Synthesis [34].

6. Statistical Synthesis [49].

7. Feature Extraction Linked to Audio Effects [65].

Features for content-based audio retrieval are classified and explained thoroughly by Mitrović et al [54].

2.1.5 Feature Importance & Feature Selection

Estimating feature importance and selecting appropriate features are the most important part in feature engineering. And the feature engineering is a critical procedure to train machine learning algorithms, because the performance of ML models are heavily dependent on the features. When training ML models, the models tend to be over-fitting, where a number of less important features, or redundant features are included in the training set. Thus, selecting proper features in each specific scenario is one of the useful techniques that models can avoid over-fitting and improve performance of the models including less computation time and reducing non-informative features. There are plenty of feature selection methods and most of them are based on statistics. The feature selection methods [14] can be classified into two groups, such as supervised methods and unsupervised methods, which are typical approaches to summarize data mining models. The classification and concepts are based on the book titled *Applied Predictive Model-*

ing [39]. The classification between supervised and unsupervised selection methods is the existence of target variable. Supervised selection methods use the target variable and remove irrelevant variables. On the other hand, unsupervised selection methods do not use the target variable, so it means the unsupervised methods remove redundant variables. Therefore, the distinction between them is that the selection techniques are called unsupervised methods, when the target vector is not considered during the elimination of predictors.

Statistics for Feature Selection

There are a number of statistical values for computing feature importance. Mutual information (MI) is widely used in many cases and also called as information gain (IG). IG is one of criteria, when splitting decision trees. MI is a type of IG that calculates the relation between two variables, so it measures the amount of information from one given random variable to the other variable. MI value can be mathematically represented as 0, if two variables are not dependent [37]. And ANOVA-F (Analysis Of Variance F-test) value is another statistical method that can be applied to measure feature importance. F-statistics (or F-test) was coined by George W. Snedecor, in honour of Sir Ronald A. Fisher. It calculates a ratio of two variances. The definition of F-test is the fol-

lowing: F = (variation between sample means) / (variation within the samples). The ANOVA-F value explores multiple features and checks whether the feature comes from same distribution or not [38]. Permutation importance is another common method to measure importance of features with the introduction of random forest algorithm by Breiman [6]. It measures the importance of a feature by calculating the increase of prediction error with the model. For example, a feature has higher importance, if permuting the feature increases the prediction error. On the contrary, a feature has lower importance, if permutation of the feature does not change the prediction error, which means the feature is not affecting the result of the model.

Supervised Feature Selection

Supervised selection methods can be classified into three main types: intrinsic, filter, and wrapper. Intrinsic methods automatically select features during training, such as decision trees including ensemble structure of decision trees and random forest. Filter methods select a subset of features based on the relevance of predictors, which pass certain criteria. Wrapper methods create and evaluate a multitude of models with a different subset of features. After the procedure, the wrapper methods choose proper features and models based on the performance,

so this method tends to be computationally expensive because of its computational complexity.

Unsupervised Feature Selection

When it comes to unsupervised feature selection, dimensionality reduction techniques, such as PCA (Principal Component Analysis) [1] and t-SNE (t-distributed Stochastic Neighbor Embedding) [75] are solutions to tackle the problem called "curse of dimensionality" [76, 2]. The curse of dimensionality occurs, when data mining models train with multiples features and high-dimensional data, because visualization can only represent 2-D and 3-D spaces. For example, provided that 120 audio features are extracted to classify musical genres, the audio features can be compressed to three PCA feature vectors for visualizing correlation between features and results.

2.1.6 Harmonic Percussive Source Separation (HPSS)

Harmonic Percussive Source Separation is an algorithm that can separate a harmonic part and a percussive part of music from an original source of audio. It is introduced by using tensor factorisation [23] and median filtering technique [22] by FitzGerald. Music source separation is a challenging problem to solve. Source separated parts cannot be mechanically divided in most cases, because the percussive part often

contains residuals of the harmonic part, and vice versa, although the quality of them are reasonable enough to experiment as research purposes, but not for music production purposes. After the prevalence of deep learning, music source separation problem was tackled by the pre-trained DL model called Spleeter by Hennequin et al [33]. Spleeter has a better performance, compared to the traditional HPSS algorithm and it also has options to choose the number of stems. If the number of stems is 2, the source will be separated into vocals (singing voice) and accompaniment. If it is 4 stems, it will be separated into four parts: vocals, drum, bass, and other. And 5 stems is the last option, as the source will be separated into five parts: vocals, drum, bass, piano and other.

2.1.7 Percussive Genre Classification (PGC)

Musical Genre Classification (MGC) task is a typical music information retrieval (MIR) problem that classifies musical genre by extracting audio features and analyzing the features. This problem is widely approached with machine learning algorithms, after one of the most cited classical MIR papers titled *Musical Genre Classification Of Audio Signals* by George Tzanetakis and Perry Cook [71]. Percussive genre classification (PGC) is a sub-field of the MGC study. PGC fo-

cuses on the problem of classifying percussive tracks. Targets for PGC can be source-separated drum tracks from original audio, or drum-only tracks without using any HPSS techniques. Analyzing drum patterns and sounds is a critical factor to define the style of music including non-western music, such as African, Cuban and Indian music. Audio genre classification using percussive patterns was explored by using clustering algorithms based on timbral audio features [68]. And using percussive patterns and bass lines can be useful to improve musical genre classification task [69]. There is also another instance [62] that HPSS algorithms improve the genre classification model based on Mel-Frequency Cepstral Coefficients (MFCCs).

2.1.8 Timbre

According to the American National Standards Institute, the definition of timbre is that attribute of auditory sensation in terms of which a listener can judge two sounds similarly presented, and having the same loudness and pitch as dissimilar [63]. For another definition, Pratt and Doak suggest that Timbre is the attribute of auditory sensation whereby a listener can judge that two sounds are dissimilar using any criterion other than pitch, loudness, and duration [63].

2.2 Previous Works for Neural Style Transfer

Figure 2.6: Source image

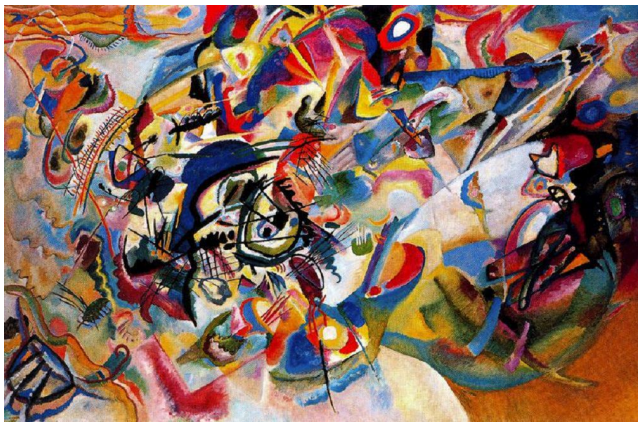


Figure 2.7: Target image.

The neural algorithm for artistic style transfer was first described in a paper that was published in 2015 titled *A Neural Algorithm Of Artistic Style* [24]. The image style transfer task typically requires two images as input: a content (source) image and a target image (or also called the style reference image). The goal of the neural style transfer model is to generate a new image representation, which has the content of the first image rendered using the painting style of the target image. For example, the Wassily Kandinsky example[1] is a typical case of the Neural Style Transfer (NST). Figure 2.6 represents the source image, which will be converted to the style of the target (Figure 2.7) image by Kandinsky. The output image (Figure 2.8) looks mixed version of the source and target image. After the emergence of image style

[1]https://www.tensorflow.org/tutorials/generative/style_transfer

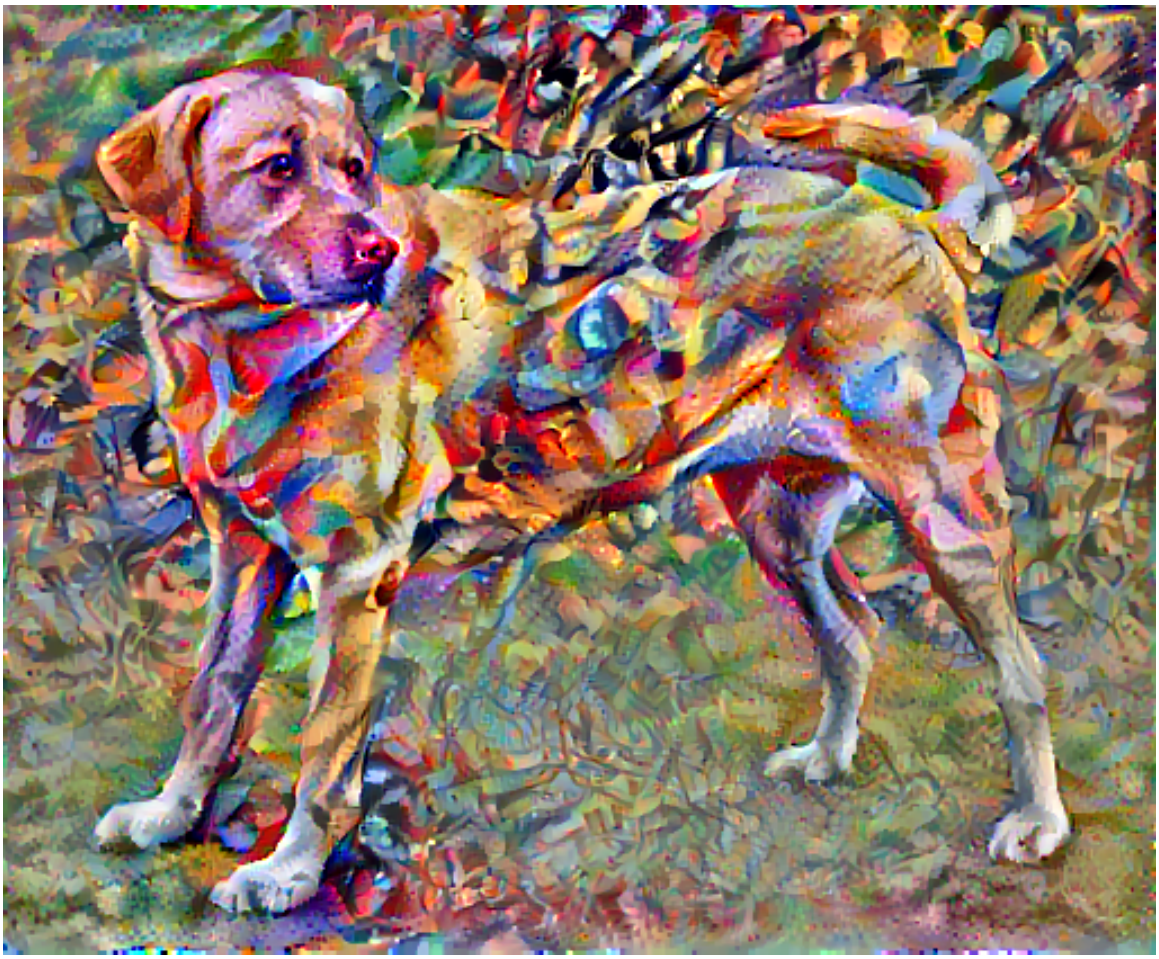


Figure 2.8: Output image using neural style transfer.

transfer, style transfer in audio domain was explored [74, 53, 29, 77]. There is an increasing work for audio style transfer using Generative Adversarial Networks (GANs). Several GAN architectures have been proposed including CycleGAN [82], WaveGAN [18], and MelGAN [40]. Music style transfer in particular is not a well-established topic [17]. One possibility for music style transfer is affecting the perceived genre of a source piece to be similar to a target musical genre. For example,

one can transform a piece of hip-hop music to sound like it is played using heavy metal instrumentation. Another challenge in music style transfer is that music genre does not have a well-defined definition and it is mostly based on the subjective perception.

When it comes to music representations used for audio style transfer, there are two main possibilities: symbolic or audio. For the symbolic music transfer, there were several successful cases mostly using GANs and VAEs (Variational Autoencoders) [11, 9, 10, 16]. With regards to audio-based music genre transfer, there are only a few recent instances [46, 15], since generating audio is typically considered as a more difficult problem, in comparison to generating symbolic patterns especially when using GANs. Specifically, audio-based GAN models tend to require more time to train and are more challenging, as they frequently contain random noise and difficult to achieve high-quality audio.

Chapter 3

Training of Musical Timbre Transfer

In this chapter, the purpose of musical timbre transfer training on source separated drum tracks using MelGAN-VC [57] is discussed. And the training procedure (Figure 3.1) and result (Figure 3.2) of the task are explained.

3.1 Purpose

The purpose for this musical timbre transfer is to achieve the transformation of timbre information from hip-hop (original genre) to metal (target genre) on drum tracks. Outputs are to retain timbral characteristics of the original source genre, while also exhibiting timbral characteristics of the target genre. To evaluate the transfer effective-

ness, I can use the probabilistic output of a trained binary classifier that discriminates between the source and target genre. If the transfer is effective, I expect that the probability of classification of the source genre will be reduced. And the transformed recording will exhibit timbral characteristics of the target genre. This change in probability of classification will depend on the specific drum track that is being transformed and the training epoch of the MelGAN-VC.

3.2 Procedure

The procedure for MelGAN-VC training for musical timbre transfer is shown in Figure 3.1. Musical timbre transfer on drum tracks was trained using the MelGAN-VC [57] architecture, which is based on the Transformation Vector Learning GAN (TraVeLGAN) [2]. The TraVeLGAN architecture, which adds a siamese network to the generator and the discriminator, and trains to preserve vector arithmetic between points in the latent space of the siamese network. Thus, the architecture can preserve semantic information in spectrograms. For experiments, GTZAN dataset[1] [72], which contains 100 tracks (30-second long each) for each 10 genre, was used. Source separated drum GTZAN stems by Spleeter [33] were trained for 4000 epochs to transfer the style from

[1]https://www.tensorflow.org/datasets/catalog/gtzan

hip-hop to metal. The GAN training stopped (4000-epoch in this case) based upon my close listening to the output sounds during testing; however, the proposed supervised pipeline can provide the reasonable number of training epoch. In this scenario, the GAN training would stop when the output reach near 50% metal/50% hip-hop. These two genres were selected, as I expect the drum timbre for them to be distinct. I focus on the drum tracks, as it is a more constrained task for timbre transfer than full music style, which depends on a variety of factors such as vocals, instrumentation, etc. This musical timbre transfer can be viewed as analogous to style interpolation between metal and hip-hop genre. The training procedure (Figure 3.1) can be summarized as following:

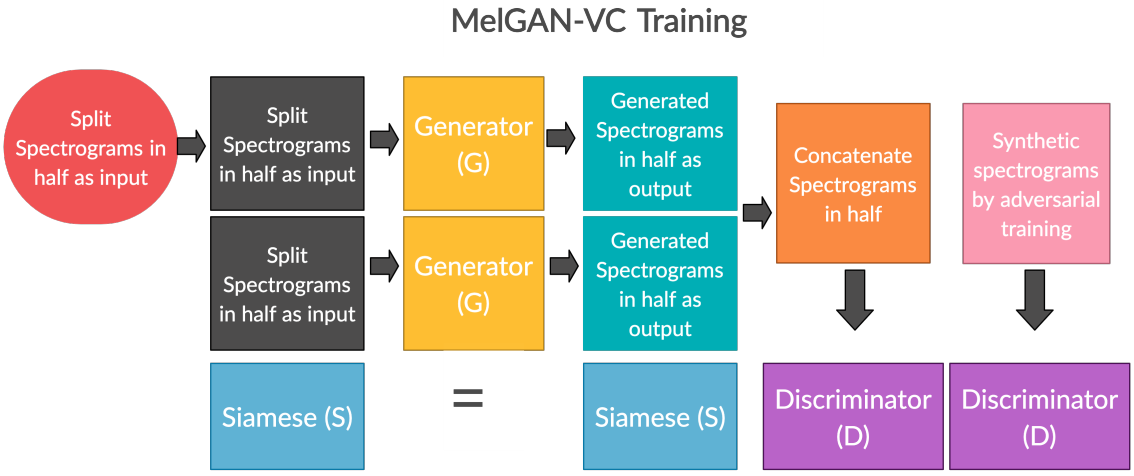


Figure 3.1: MelGAN-VC training procedure.

1. Spectrograms (time-frequency 2-dimensional representation) were

Figure 3.2: MelGAN-VC generator and discriminator loss.

extracted from the drum tracks.

2. The spectrograms were split in half and used as input to the generator (G).

3. The style-translated halves from G were concatenated to the original shape and transferred to the discriminator (D).

4. The Siamese network (S) was added to keep semantic information [2], which captures music style.

5. Adversarial training (4000 epochs): D distinguished metal from hip-hop to improve style-transferred instances generated by G, and

S assisted G training by allowing translations between metal and hip-hop.

6. The missing phase information (spectrograms only include magnitude) was reconstructed by the Griffin-Lim [28] algorithm.

3.3 Training

The rationale for selecting the MelGAN-VC is that it can be trained with only 100 tracks for each genre (100: hip-hop and 100: metal), which is relatively small number of data set. And the MelGAN-VC architecture includes siamese networks, which are helpful to avoid checkerboard artifacts [56], and it worked better than the CycleGAN especially for transferring audio style according to the original paper [57]. The main limitation of the CycleGAN is that it shows lower performance for transferring diverse domains, not similar domains, because it cannot preserve high-level semantic differences [80, 25, 5], such as shapes, types of specific objects, and composition, as opposed to low-level pixel differences, e.g., color, resolution, and lines. On the other hand, MelGAN-VC works better and can keep content information to implement an image style transfer task in diverse domains. In other words, the MelGAN-VC architecture can learn and use high-level semantic differences for style-transferring images, when there are signif-

icant differences between a source image and a target image. As an audio style transfer example, each musical genre has significant differences when both audio waveforms are represented as spectrograms, and this problem is considered diverse domains, so it is challenging to use CycleGAN to implement the audio style transfer from one genre to another genre; however, MelGAN-VC works better in this case, because it can keep content information from the specific musical genre from the spectrogram. In this scenario, the main role of siamese networks is to preserve vectorized semantic information from the spectrogram and keep the timbral information.

The losses of the generator G and the discriminator D for the MelGAN-VC model after training for 4000 epochs are shown in Figure 3.2. First, at least both G and D loss are not decreasing to 0.0, which is considered as a failure mode. Second, the losses are not changing drastically, after 1000 epochs and the values look stabilized in a certain degree with relatively small fluctuations. Lastly, at least human listeners can tell there are differences between training epoch 100 and 2500, and the epoch-2500 sounds improved, compared to the previous early epoch 100. This observation could be strengthened by performing a thorough user study, but the goal is to investigate whether I can support this observation using supervised and unsupervised learning methods.

Chapter 4

Evaluation Methods for Musical Timbre Transfer

Quantitative evaluation methods (Figure 4.1) including the audio feature importance (Figure 4.2) for evaluating style-transferred drum instances from the GAN model are discussed. The methodologies can be divided into two approaches: supervised and unsupervised method. Essentia [4] is used to extract 110 audio features for both methods.

4.1 Audio Feature Selection

Selecting important audio descriptors for drum genre classification is an important task for this evaluation process. The evaluation models trained are dependant on the features and tend to overfit given the size of the data-set when using all 110 audio features. In addition, using

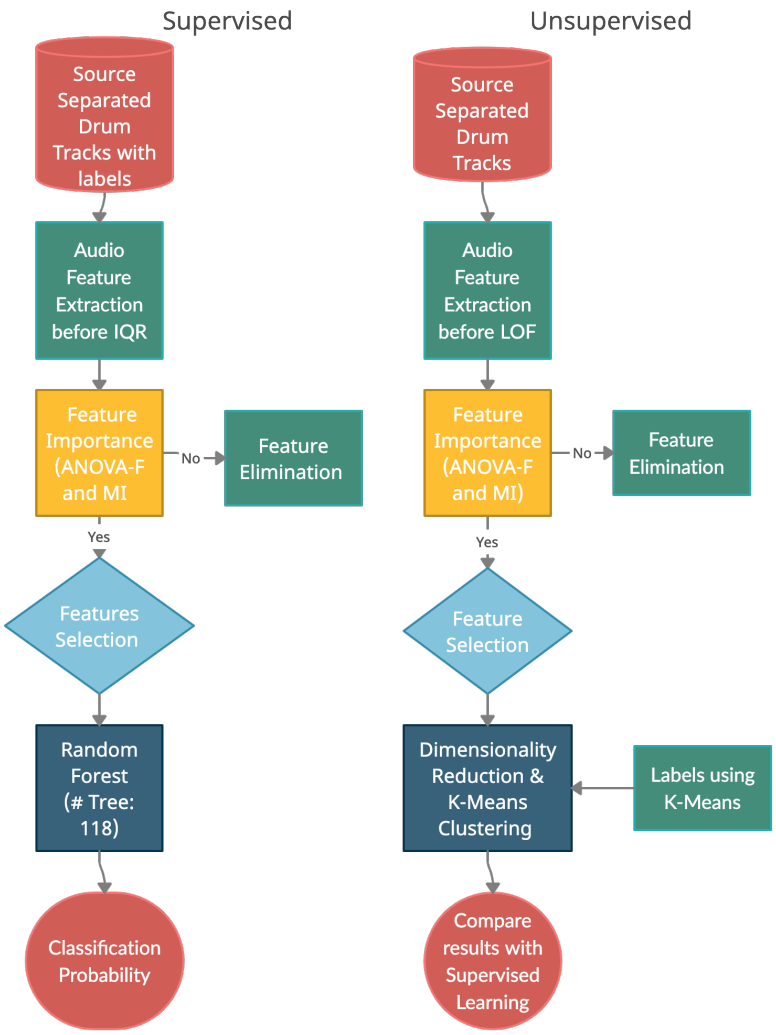


Figure 4.1: Pipelines for quantitative evaluation methods of musical timbre transfer. Left: supervised method, and right: unsupervised method.

all 110 features is not efficient for computing evaluation pipelines. It is worthwhile to explore and get results for the importance of audio features by analyzing the binary music genre classification. Mutual information (MI) and Analysis of Variance F-value (ANOVA-F) were used for estimating the importance of each audio feature. 110 fea-

Top 10 Audio features	ANOVA-F values
lowlevel.erbbands_skewness.stdev	50.89586639
lowlevel.loudness_ebu128.integrated	47.47364044
lowlevel.barkbands_spread.stdev	44.71256256
lowlevel.melbands_spread.stdev	37.2513504
lowlevel.spectral_rms.stdev	35.28242111
lowlevel.melbands_crest.stdev	34.80794144
lowlevel.barkbands_flatness_db.mean	32.10904312
lowlevel.melbands_kurtosis.mean	31.87611771
lowlevel.loudness_ebu128.loudness_range	30.59831238
lowlevel.barkbands_flatness_db.stdev	30.08287048

Top 10 Audio features	MI values
lowlevel.loudness_ebu128.loudness_range	0.2793418142
lowlevel.barkbands_spread.stdev	0.2785261127
lowlevel.pitch_salience.mean	0.2643023024
lowlevel.loudness_ebu128.momentary.stdev	0.2589722888
lowlevel.erbbands_skewness.mean	0.2476374769
rhythm.bpm_histogram_second_peak_weight	0.2160613488
lowlevel.loudness_ebu128.integrated	0.2113145765
lowlevel.melbands_crest.mean	0.1974811933
lowlevel.spectral_kurtosis.mean	0.1923467722
lowlevel.barkbands_skewness.mean	0.1883028758

Figure 4.2: Top 10 audio feature importance for percussive genre classification. Top: ANOVA-F and bottom: Mutual information.

tures (come from Essentia Music Extractor[1], excluding temporal features and metadata, to evaluate holistically) were extracted from three genres (hip-hop, metal, and rock from source separated drum tracks from GTZAN dataset) and each top 10 features (Figure 4.2) were se-

[1]https://essentia.upf.edu/reference/std_MusicExtractor.html

Weight	Feature
0.0870 ± 0.0094	percussive_var
0.0508 ± 0.0068	percussive_mean
0.0465 ± 0.0067	harmony_mean
0.0383 ± 0.0044	mfcc4_mean
0.0325 ± 0.0030	chroma_stft_mean
0.0274 ± 0.0054	harmony_var
0.0262 ± 0.0050	rms_var
0.0190 ± 0.0024	mfcc11_mean
0.0186 ± 0.0045	mfcc9_mean
0.0164 ± 0.0029	tempo

Figure 4.3: Permutation feature importance (audio features based on Librosa library [50] are used for this experiment.) with original GTZAN data set. The result of importance weight is noticeable that percussive variance and percussive mean features are significant factors to classify musical genre, even if it is not a percussive genre classification task.

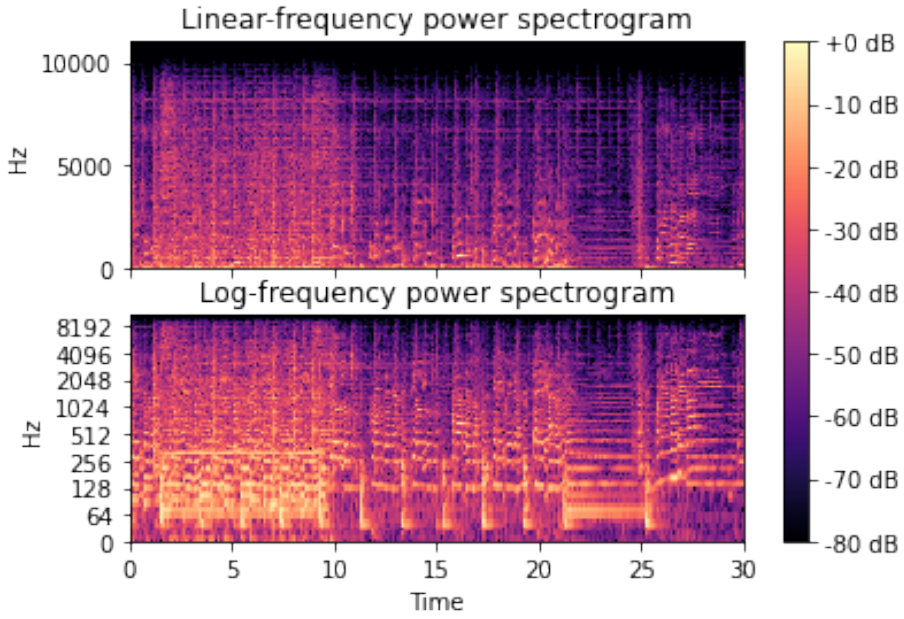


Figure 4.4: Spectrograms for original GTZAN rock audio.

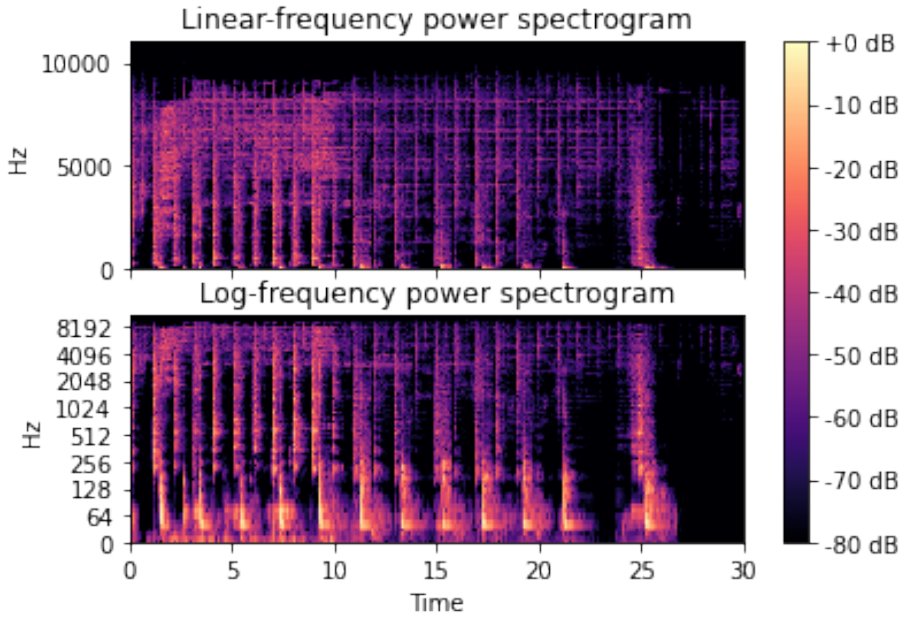
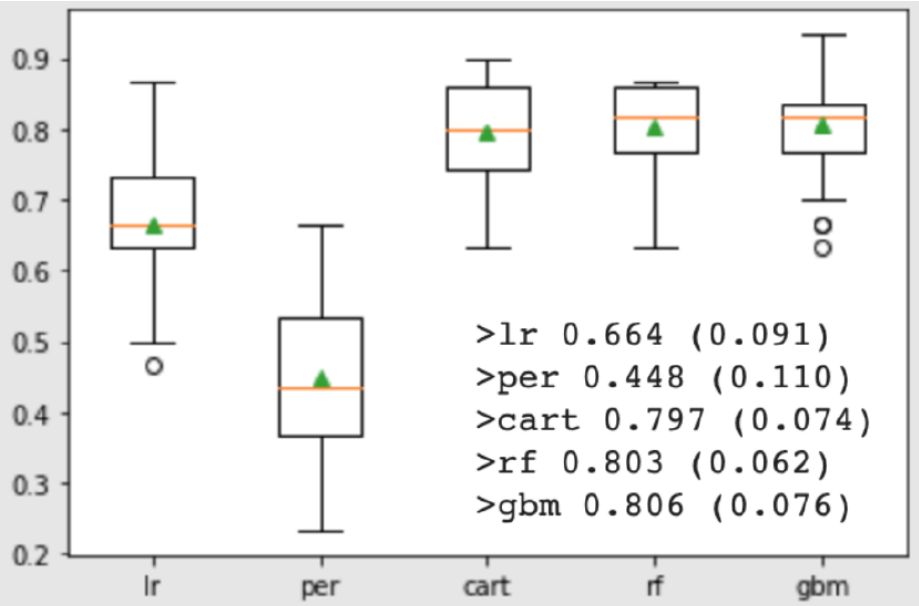


Figure 4.5: Spectrograms for GTZAN rock source separated drum audio.

lected based on MI and ANOVA-F, respectively. Therefore, for the supervised learning experiments, 17 features were selected. However, for the unsupervised learning experiments, I found that the ideal number was 6, based on the accuracy of the number of features between 1 and 20. And top 6 features in Figure 4.2 were selected using ANOVA F-values, because it shows better accuracy when using ANOVA-F. A noticeable point is that only one rhythm-related feature (as shown in Figure 4.2, when using MI) was discovered according to the feature importance weight, although other audio low-level descriptors could be directly or indirectly related to rhythm.

Permutation feature importance for genre classification with original

GTZAN data set is implemented as an additional experiment. Permutation feature importance (audio features based on Librosa library [50] are used for this experiment.) with original GTZAN data set is shown in Figure 4.3. The permutation feature importance is extracted from audio features based on 30-second long and the .csv file is available on Kaggle[2]. The result of feature importance is important to note that percussive variance and percussive mean are significant features to classify musical genre, even if it is not a percussive genre classification task. The percussive related features are extracted by harmonic-percussive source separation algorithms[3]. Therefore, analyzing audio signal problems using source separated drum or percussive tracks can be useful to approach musical timbre transfer task. In addition, drum tracks contain less timbral information, in comparison to original tracks (contain vocal, drum, bass, etc) at least based on timbral information (as shown in Figure 4.4 and 4.5), so it is very challenging to implement musical timbre transfer, due to its complexity of timbre in the original track.




```
model = RandomForestClassifier(n_estimators=118)
model = model.fit(X, y)
print(model)
cv = RepeatedStratifiedKFold(n_splits=5, n_repeats=3, random_state=1)
scores = cross_val_score(model, X, y, scoring='accuracy', cv=cv, n_jobs=-1)
print('%.3f (%.3f)' % (mean(scores), std(scores)))
```

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='gini', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=118,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
0.910 (0.049)
```

Figure 4.6: Comparison of supervised learning classifiers: logistic regression (lr); perceptrons (per); classification and regression trees or decision trees (cart); random forest (rf); and gradient boosting machines (gbm).

4.2 Supervised Learning Method

4.2.1 Training

The supervised learning method for analyzing musical timbre transfer was trained with 17 audio features. There was a trend that ensemble

[2]https://www.kaggle.com/andradaolteanu/gtzan-dataset-music-genre-classification
[3]https://librosa.org/librosa_gallery/auto_examples/plot_hprss.html

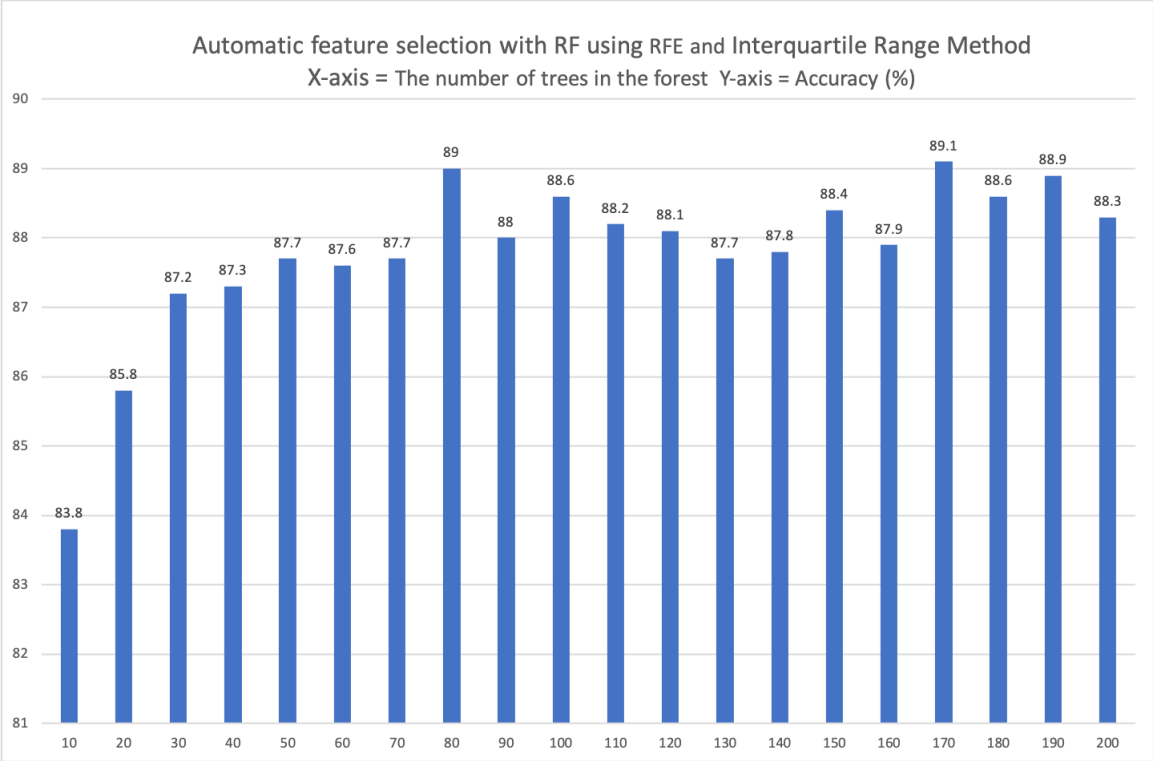


Figure 4.7: Accuracy for random forest classifier using recursive feature elimination (RFE) and interquartile range method (IRM).

methods, such as random forest (RF) and gradient boosting algorithm, have higher classification accuracy in comparison to logistic regression and multi-layer perceptrons as shown in Figure 4.7. Therefore, the RF algorithm was chosen as the estimator for this case with the configuration of 118 trees in the forest and trained with two genres: hip-hop (label: 0) and metal (label: 1). As a result, the accuracy score turned out to be 91.0% for drum genre classification on hip-hop and metal for the source separated GTZAN drum tracks. Note that the drum stems from Spleeter are used and therefore this is purely based on source sepa-

rated drum tracks. Figure 4.6 explains details about the experiment for comparing multiple supervised machine learning classifiers with accuracy by using a boxplot and the below code is for implementing random forest classifier with the configuration of 118 trees in the forest. The number 118 is achieved GridSearchCV [4] function in Sci-kit learn library, which selects the best hyperparameter for the model based on the accuracy. Figure 4.7 is another experiment with random forest classifier by applying recursive feature elimination (RFE) and interquartile range method (IRM). In this case, 80 trees in the forest (better than 170 or 190, because of less computation time) is the best parameter. RFE is to reduce the number of features by iterating and calculating feature importance. With IRM, outliers can be eliminated by setting the range when you are pre-processing data sets. For example, if you set 10/90 for the range, 90 percent of the data set will be used and 10 percent will be excluded. RFE and IRM are mostly useful, when you deal with bigger data sets. In this scenario, only GTZAN data set is used for experiments and the experiment with RFE and IRM did not show meaningful improvements regarding accuracies of classifiers.

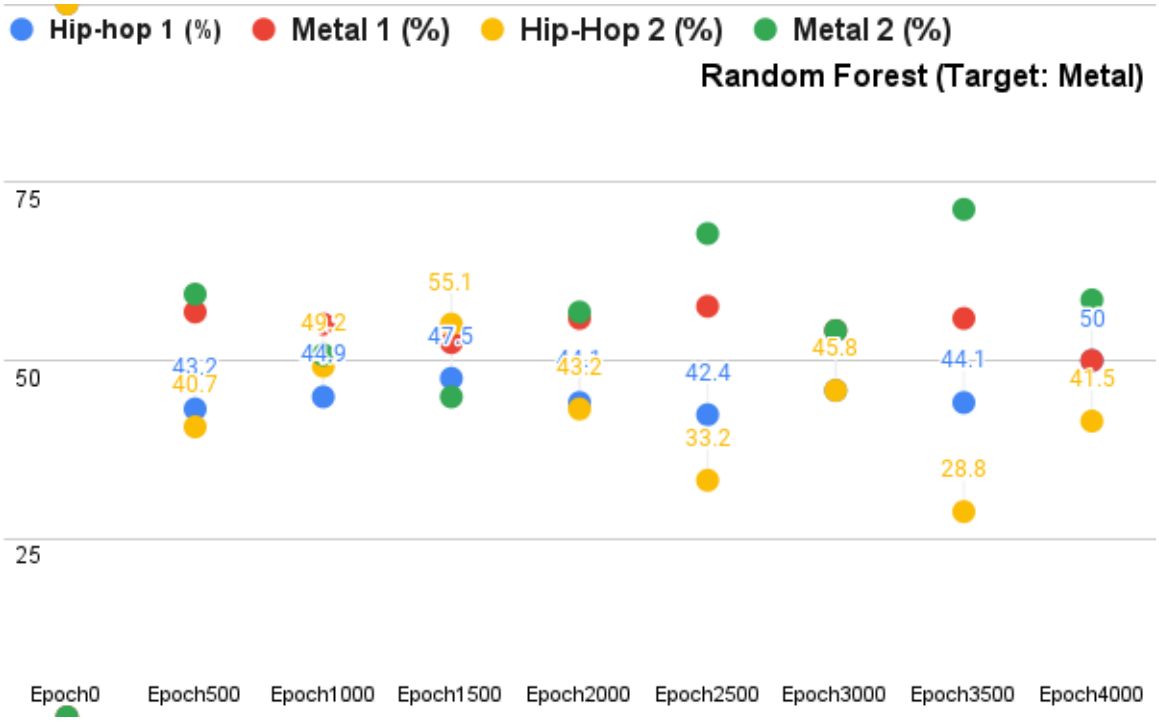


Figure 4.8: Supervised method results: track 1 & 2.

4.2.2 Testing

I tested the evaluation method with five randomly chosen instances from the style-transferred GTZAN drum tracks and one random hip-hop style drum-only track (without source separation) from outside of the GTZAN dataset. The results were analyzed at intervals of 500 epochs, within the range of 500 to 4000 epochs from hip-hop to metal. The main purpose for this method is to reflect the trend for the progress of musical timbre transfer. Figure 4.8, 4.9, 4.10, and 4.11 demonstrate evaluation results with the supervised learning pipeline for classifying

[4]https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

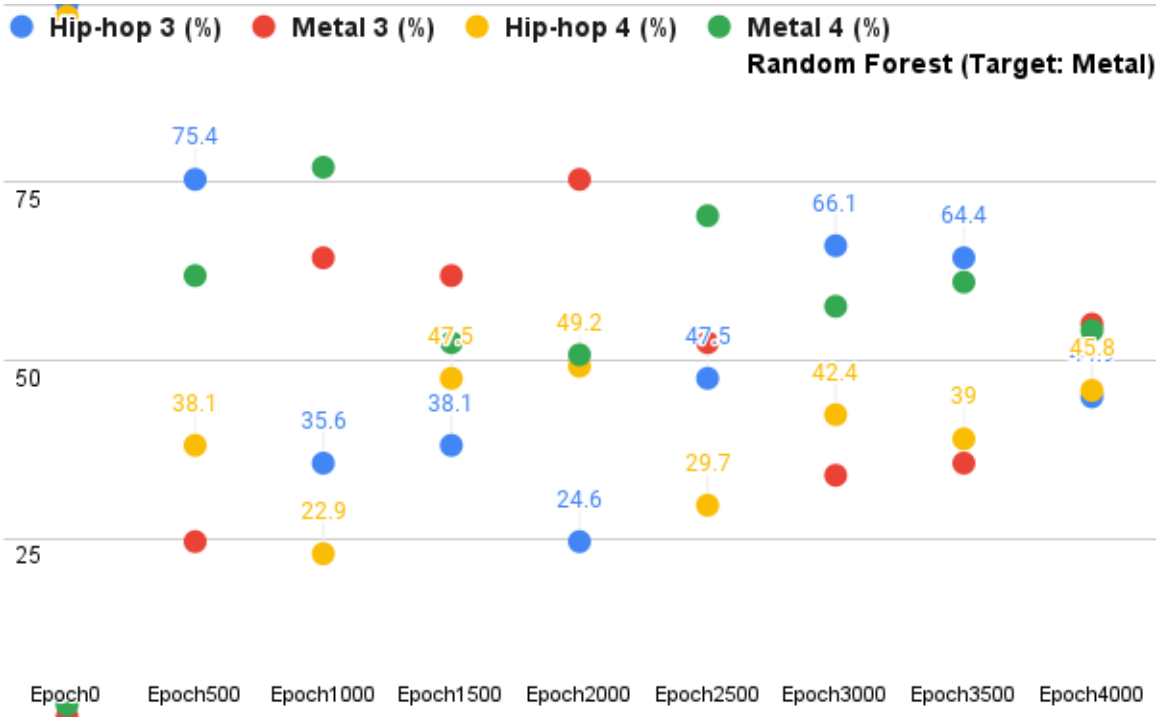


Figure 4.9: Supervised method results: track 3 & 4.

six style-transferred tracks based on the RF classification probability. The ideal case for this result is where hip-hop and metal is near 50%, respectively, which proves the achievement of style interpolation between hip-hop and metal. The interpolated line graphs in Figure 4.11 are demonstrating that the MelGAN-VC training is working because the testing instances are nearly converging to 50/50% as the number of training epochs increase. All line graphs are oscillating for both the discriminator and the generator loss. Top left scatter plot is showing the results for two style-transferred instances. The first test instance (hip-hop1/metal1) is the hip-hop style drum-only track and it

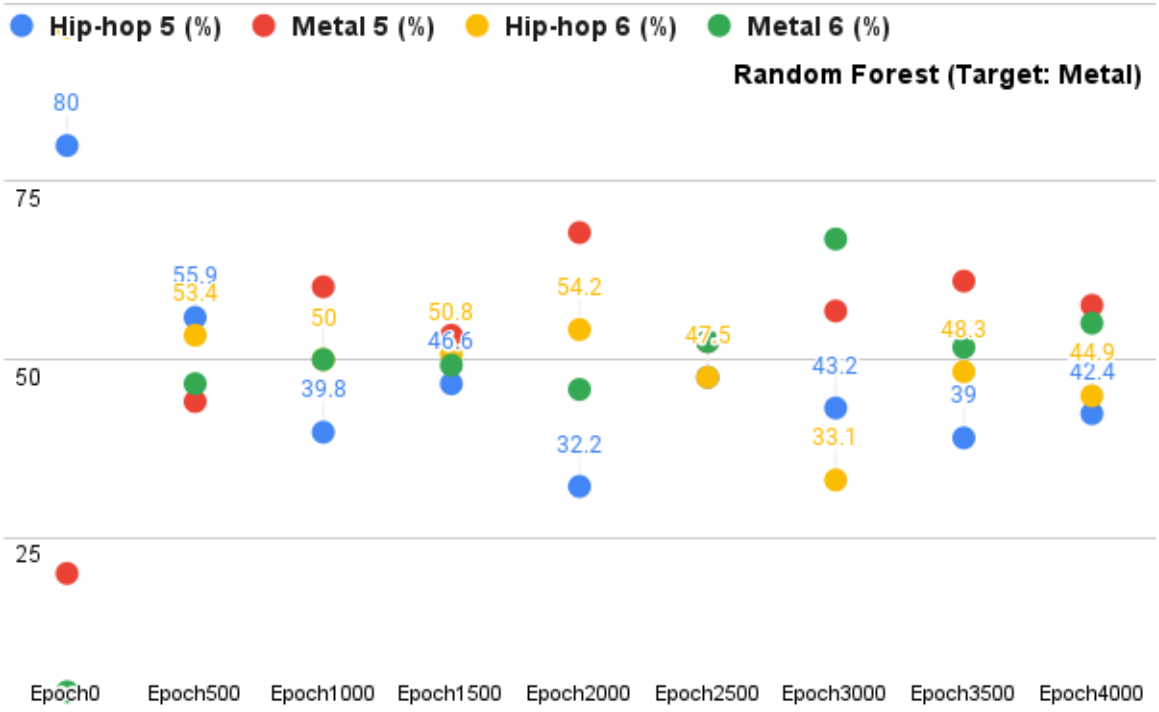


Figure 4.10: Supervised method results: track 5 & 6.

was calculated hip-hop (50%) and metal (50%). The other five examples (hip-hop2-6/metal2-6) are more fluctuating compare to the first one. As I expected, the musical timbre transfer works better with the drum-only piece (track 1) because the source separated GTZAN drum stems still contain some portion of other stems, such as vocal and bass. When interpreting the interpolated line graphs, it is reasonable to ignore the intersection point (50/50%) before epoch 1500 because it is the early stage of the MelGAN-VC training, and tends to include more random noise. The testing interval was 500 epochs (0, 500, 1000, etc) and the results between them, e.g., 300 and 700, were interpolated

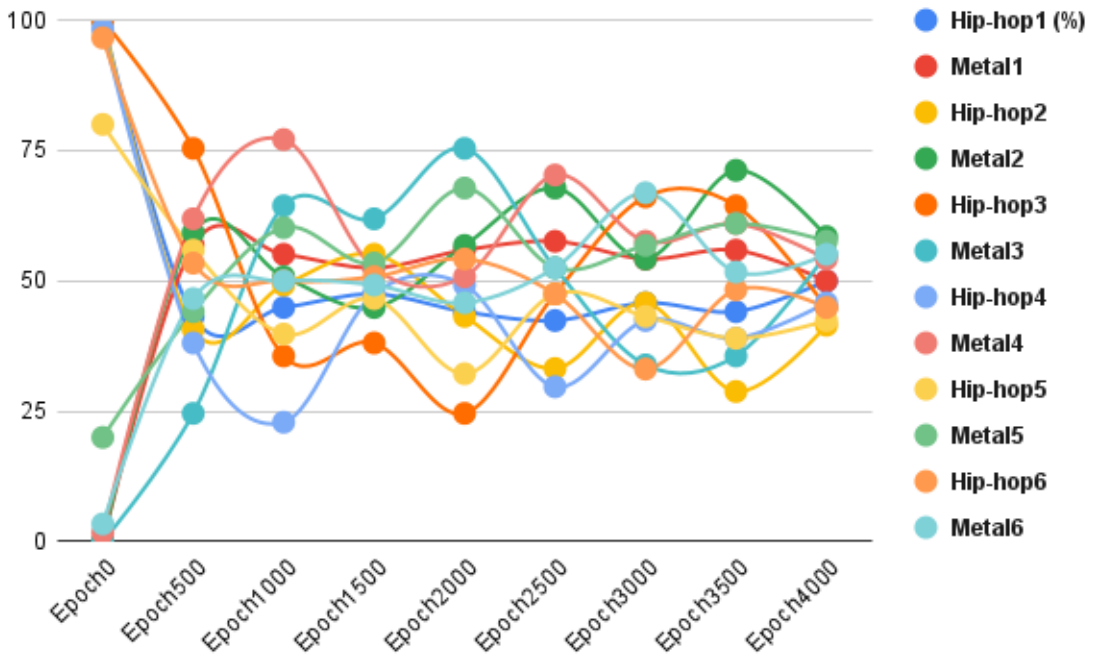


Figure 4.11: All tracks trend using interpolation.

as shown in Figure 4.11; however, this could be solved by simply reducing the testing interval with the same method. According to the supervised evaluation pipeline, the best example instances[5] are listed in sequence: hip-hop/metal 1) epoch4000, 2) epoch3000, 3) epoch2500, 4) epoch2000, 5) epoch2500, and 6) epoch3500.

4.3 Unsupervised Learning Method

To investigate further how the style transformed instances related to the original source and target genre, I employ clustering and visual-

[5]https://drive.google.com/drive/folders/1dpn8NL0fhtXJ4JhzeAeIlDyxUzQWHttC?usp=sharing

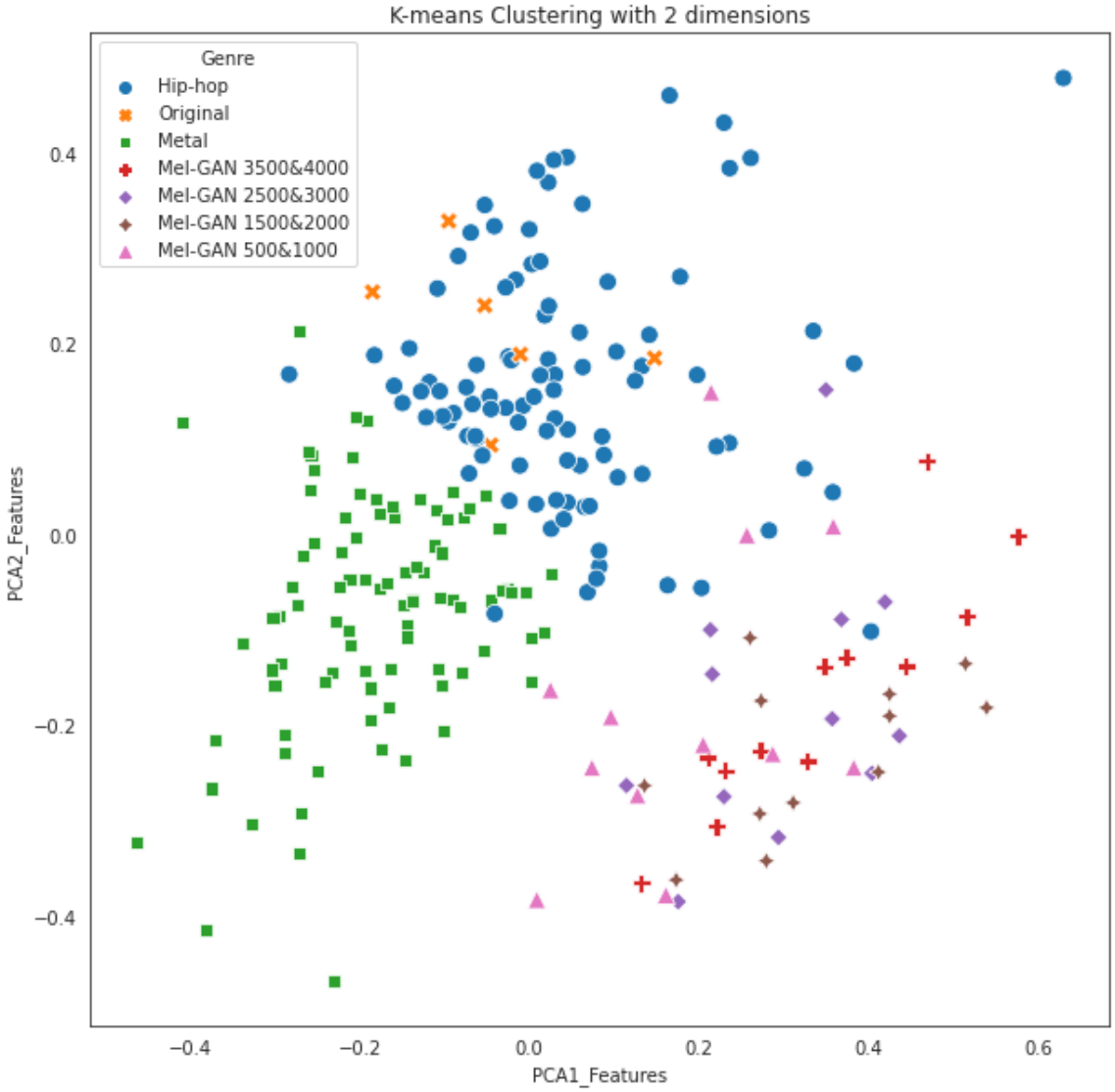


Figure 4.12: Unsupervised method clustered by K-means and visualized by PCA. Style-transferred outputs by MelGAN-VC is demonstrated depending on the training epoch (the original legend denotes epoch-0 outputs).

ization techniques, which are part of unsupervised machine learning algorithms. The K-means algorithm for discovering 2 clusters was used with the 200 GTZAN drum stems (100 metal and 100 hip-hop each) based on audio features. Top 15 audio features for consideration were

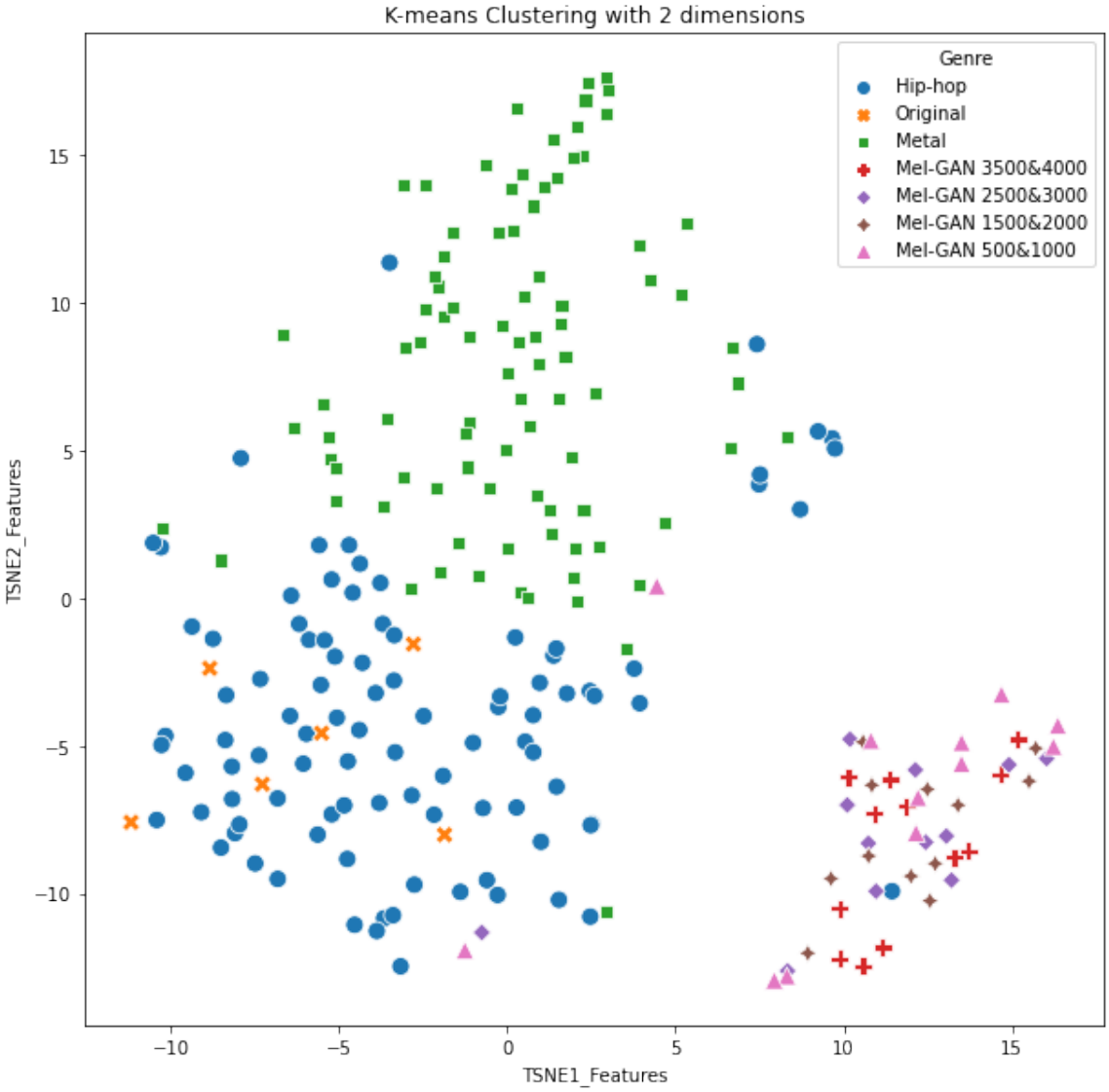


Figure 4.13: Unsupervised method clustered by K-means and visualized by t-SNE. Style-transferred outputs by MelGAN-VC is demonstrated depending on the training epoch (the original legend denotes epoch-0 outputs). t-SNE works better than PCA, and the outputs are clustered independent of original hip-hop and metal clusters.

selected by extracting feature importance using ANOVA-F and MI, respectively. The clustering was evaluated by examining how well it captures the original two genres. The K-means algorithm with ANOVA-F

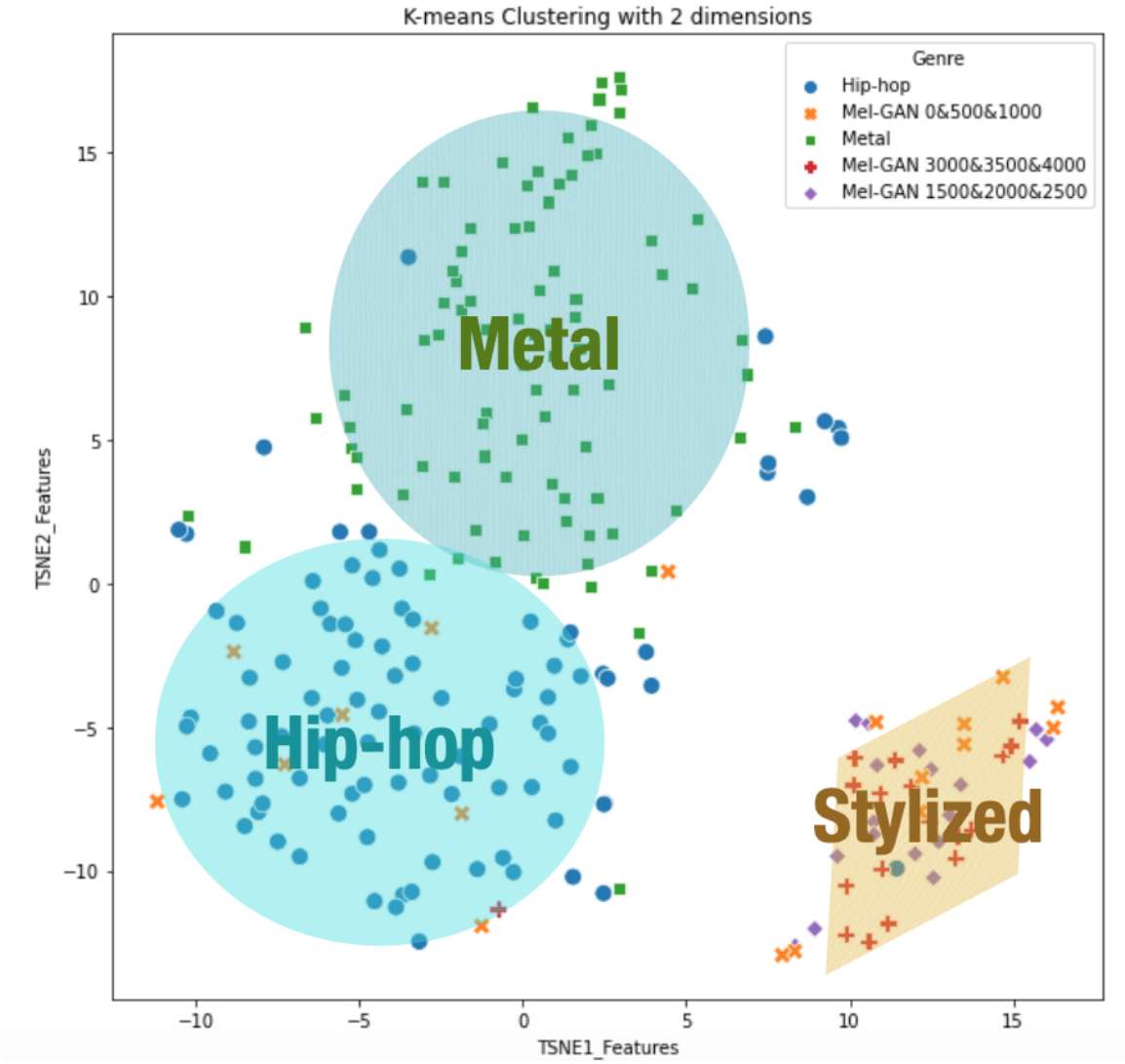


Figure 4.14: Unsupervised method clustered by K-means and visualized by t-SNE. Three big clusters (Metal, Hip-hop, and Stylized) can be observed from the experiment.

created clusters, where top 5 features were selected. 112 instances were classified as metal and 88 examples were labelled hip-hop. When K-means produced clusters with MI, it classified better than ANOVA-F. Top 6 features (in Figure 4.2 MI table) were selected and the two genres

were labelled almost evenly: 102 for hip-hop and 98 for metal. As a result, I decided to select MI with K-means based on these six audio features.

In order to better understand the musical timbre transfer, dimensionality reduction (PCA & t-SNE) is utilized. Principal component analysis (PCA) and t-distributed Stochastic Neighbor Embedding (t-SNE) were used to visualize 200 GTZAN drum stems along with 48 style-transferred instances: 6 transferred instances by MelGAN-VC at 8 training epoch points, 500, 1000, 1500, etc. Figure 4.12 and 4.13 are demonstrating that three big clusters can be observed based on hip-hop, metal and Mel-GAN instances. t-SNE components worked slightly better because it generated more robust clusters based on the scatter plots in Figure 4.13. And it is clearly representing that the Mel-GAN instances are moving away from the original dots. Mel-GAN instances where the epoch range between 2500 and 4000 are located outside of hip-hop and metal clusters. Moreover, the t-SNE plot provides that the instances of that particular epoch range (stronger instances) are clustered more densely compared to weaker instances (Mel-GAN 500 & 1000). Furthermore, the stronger instances (when greater than 2000 epochs) are style-transferred better based on results. Three big clusters (Metal, Hip-hop, and Stylized) can be observed from the experiment

(Figure 4.14). The relation among Metal, Hip-hop, and sytlized can be visualized and compared with supervised evaluation results, although especially several Hip-hop instances are located outside of its cluster. Because Hip-hop style tends to be diverse and has a number of sub-genres: East coast, West coast, G-Funk, Trap, etc. In Appendix A, all testing instances are compared using spectrograms for checking the results.

Chapter 5

Conclusions & Future Work

In this chapter, conclusions and limitations for MelGAN-VC generative model and computer evaluation methods are discussed. And directions of future work are suggested to improve the current model and methods.

5.1 Conclusions

MelGAN-VC can be used for musical timbre transfer on drum tracks between two genres, particularly achieving style interpolation, which is the output sounds similar to an in-between genre of hip-hop and metal drum tracks. Evaluation methods for the musical timbre transfer are introduced by using supervised and unsupervised learning with visualization. It is a reasonable first step to explore the potential of existing state-of-the-art GAN model. Computer evaluation methods with visualization are an important part, because a number of previous audio

style transfer experiments analyzed results by simply comparing spectrograms for original inputs and style-transferred outputs or conducting user studies. Therefore, designing novel evaluation pipelines for computational approaches could be regarded as an important milestone to explore and improve results of audio style transfer in general, and in this case, musical timbre transfer on drum tracks.

5.2 Limitations

The first limitation for this musical timbre transfer is that it is difficult to be used in real-time applications, such as digital audio workstation, because of lower audio quality compared to commercial audio quality, and the timbre transfer task requires long training time (around 2 weeks for training 4000 epochs with Google Colab Pro GPU), although it is a valuable first step to explore and find possibilities in this research area. And one of the main limitations for this research is that it does not consider temporal information, which is a critical factor to create rhythms and each genre has different tempo on average. And another drawback is the current model uses short-time fourier transform (STFT) spectrograms to extract time and frequency information from audio files, and phase information cannot be estimated and reconstructed perfectly by using the Griffin-Lim algorithm [28]. Thus, it can lead to lower audio

quality, and inaccurate generation of musical timbre transfer. In addition, the STFT method has lower resolution in low-range frequencies [35], which is typically the frequency range for kick (or bass) drum and floor tom in most cases. Bass drum is top 3 drum components along with hi-hat and snare for creating rhythm in music, so the lower resolution for the bass drum part could be problematic, especially when you are implementing style transfer using timbre. And the last limiting point is the current generative model needs to consider tempo, orchestration, and syncopation to generate realistic rhythm. Besides previous points, the computational evaluation model is required to integrate with the generative system that the system can be trained efficiently, and the model only evaluate musical genre holistically with 30-second long pieces or longer, so it may have a lower performance with shorter pieces (3 seconds). Both generative and evaluation model need to be experimented with bigger data sets to achieve better performance in many scenarios, not limited to specific cases. The limitations can be summarized as follows:

1. The current model has difficulties for real-time applications due to low audio quality, and long training time.

2. It does not consider temporal information which is important for creating rhythm depending on the music genre.

3. Use of STFT and its limitation of phase reconstruction and lower resolution in low-range frequencies, particularly for bass drums.

4. The current generative model (MelGAN-VC) needs to consider tempo, orchestration, and syncopation to generate realistic rhythm along with transferring timbre.

5. The current evaluation methods are required to integrate with the generative model and can make the MelGAN-VC training process more efficient.

6. Both generative and evaluation model need to be experimented with bigger data sets to achieve better performance in many scenarios, not limited to specific cases.

5.3 Future Work

For future work, limitations of the current model should be tackled by proper measures or appropriate solutions. And further experiments need to be implemented to make the model robust and explore potentiality. Long training time can be solved by simply training 4 bars of rhythm, instead of training a full-length of a drum track. Experimenting with different genre of drum loops could be better for training time and audio quality, rather than using source separated drum

tracks. Low audio quality and the limitation of using STFT can be solved by using Constant-Q transform (CQT) [8], because CQT provides higher resolution especially for low-range frequencies [35]. And it may improve the result of musical timbre transfer on drum tracks. Timbretron [35] achieved better performance for musical timbre transfer, when using CQT. However, using CQT and MelGAN still require to estimate phase information, which would not be achieved perfectly by a current method. Thus, recent deep learning based approaches can be utilized in this scenario to circumvent the phase reconstruction problem. There are deep learning based models to improve audio quality of speech, such as Tacotron2 [64], and WaveGlow [59]. To consider temporal and rhythmic information, use of a beat synchronous representation [66] would be helpful, although reconstructing output audio from the beat synchronous representation can be challenging and needs to be explored. Consideration of drum orchestration and syncopation directly in audio domain can be challenging with audio GAN models, but it might be possible by training patterns of separated drum sounds (hi-hat, bass, and snare) for each music genre. Separated drum sounds can be obtained by NMF toolbox [44] from drum tracks. An example of drums generation in the symbolic domain is [47]. Comparisons of multiple audio GAN architectures using the same methodology, and

expansion of current examples to additional pairs of genre style transfers would be explored with bigger data sets to improve robustness of the model. And integrating current generative and evaluation model would be useful for training GAN by considering both GAN loss values, and supervised learning evaluation results. For example, if the output audio of generative model achieves desired results, the generative does not need to keep training. Applying the same technique with other source separated stems (vocals and bass) would be interesting as well. Future work can be summarized as follows:

1. Training with shorter length of drum tracks.

2. Experiments with CQT or deep learning based approaches to circumvent phase estimation procedure.

3. Use of a beat synchronous representation to consider temporal and rhythmic information.

4. Experiments with separated drum sounds (hi-hat, bass, and snare) by NMF toolbox [44] for consideration of drum orchestration and syncopation.

5. Comparisons of multiple audio GAN architectures using the same methodology and expansion of current examples to additional pairs of genre style transfers.

6. Integration of current generative and evaluation model would be useful for training GAN by considering both GAN loss values, and supervised learning evaluation results.

7. Application of the same technique with other source separated stems, such as vocals and bass.

Appendix A

Spectrograms of Results

According to the supervised evaluation pipeline, the best example instances[1] are listed in sequence: hip-hop/metal 1) epoch4000, 2) epoch3000, 3) epoch2500, 4) epoch2000, 5) epoch2500, and 6) epoch3500. The best instances are compared using spectrograms for checking the results. Linear-frequency power and log-frequency power spectrograms are plotted for comparison between original source separated drum stems and style-transferred drum tracks.

[1] https://drive.google.com/drive/folders/1dpn8NL0fhtXJ4JhzeAeIlDyxUzQWHttC?usp=sharing

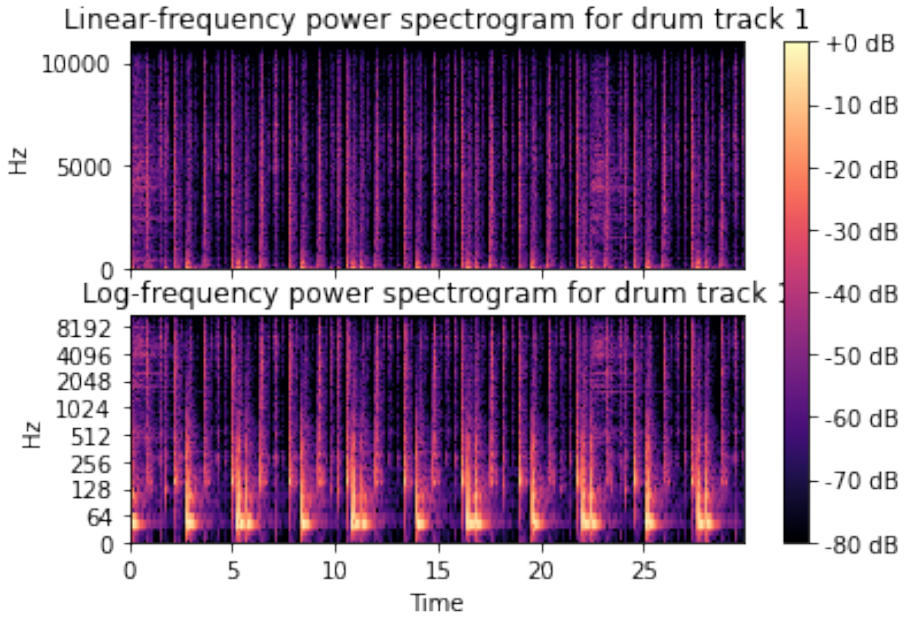


Figure A.1: Spectrograms of original drum track 1.

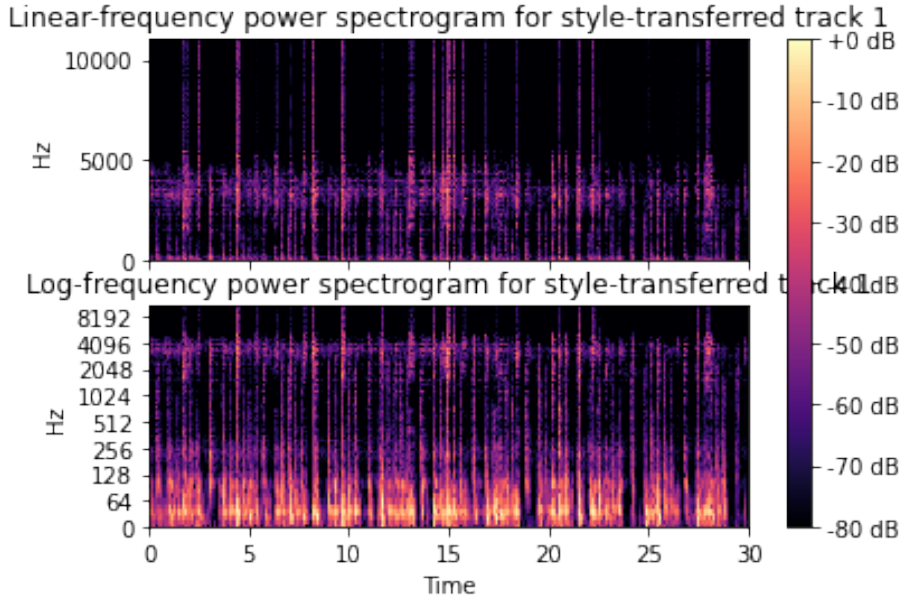


Figure A.2: Spectrograms of style-transferred drum track 1.

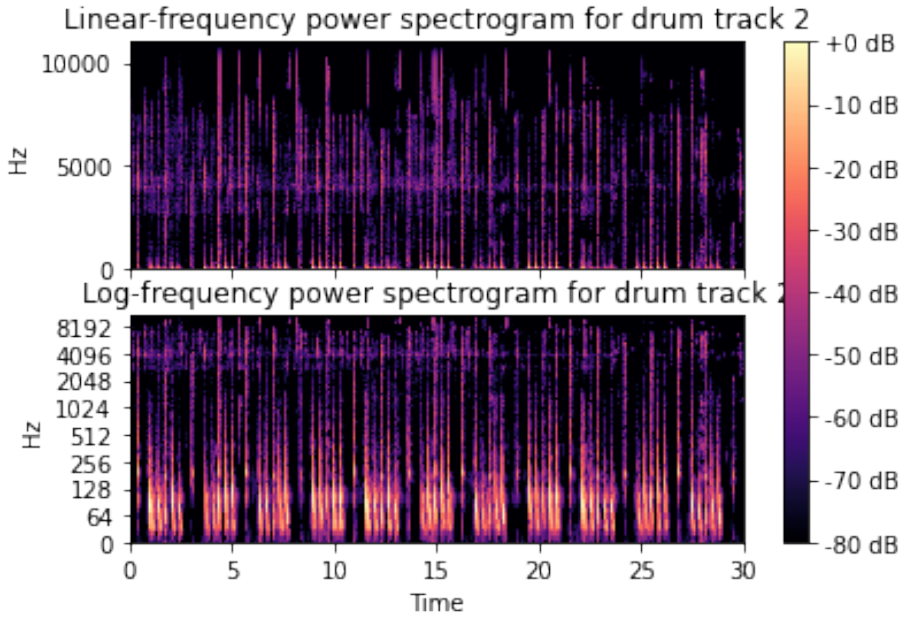


Figure A.3: Spectrograms of original drum track 2.

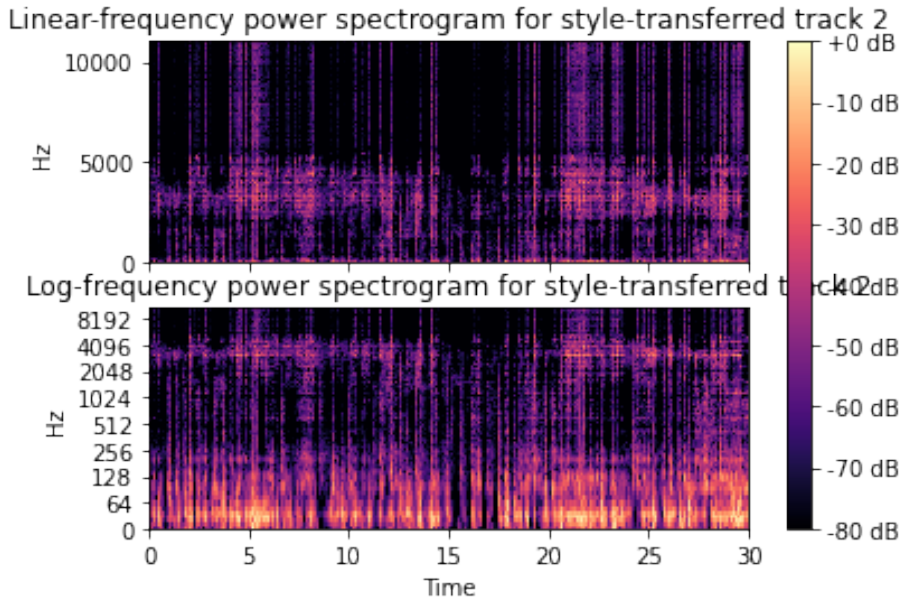


Figure A.4: Spectrograms of style-transferred drum track 2.

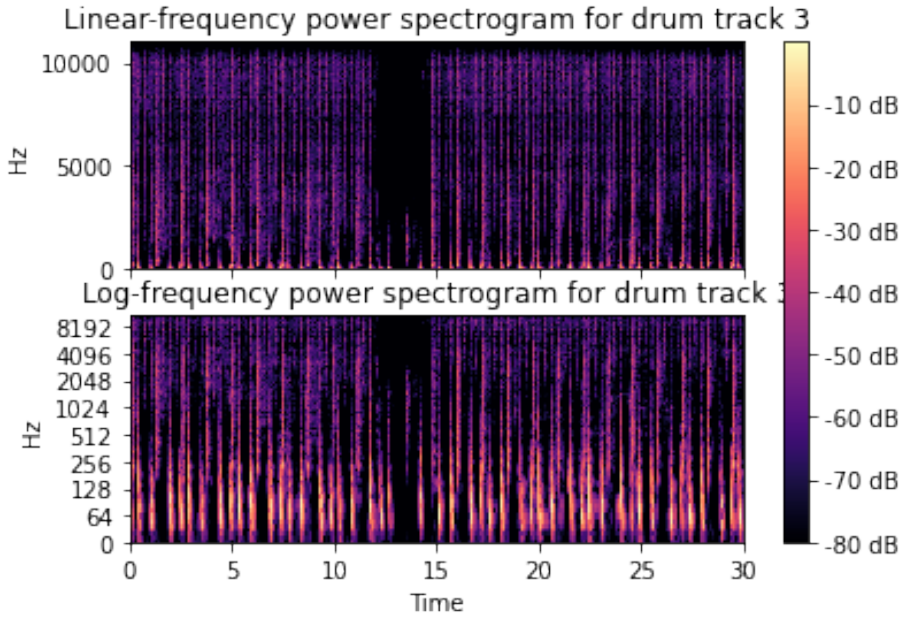


Figure A.5: Spectrograms of original drum track 3.

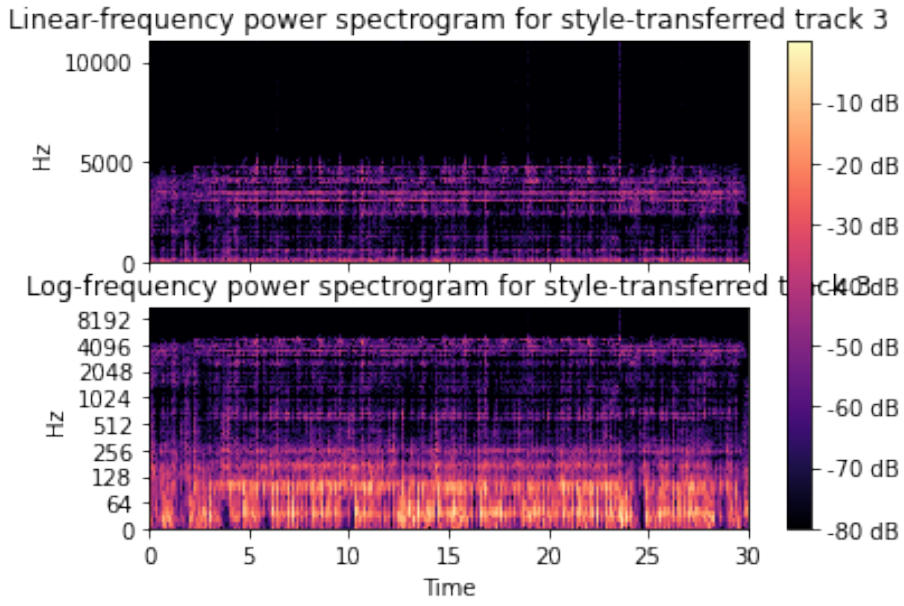


Figure A.6: Spectrograms of style-transferred drum track 3.

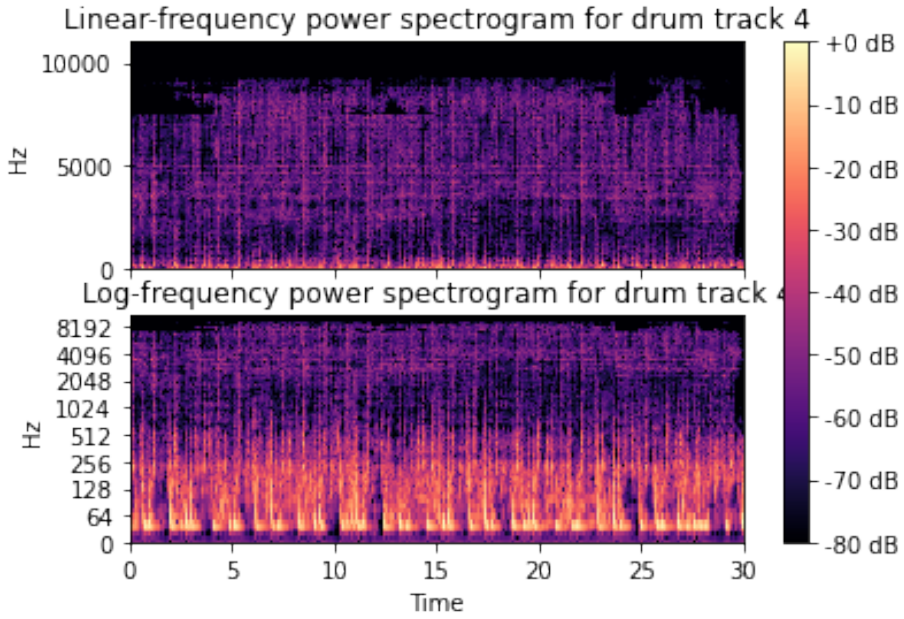


Figure A.7: Spectrograms of original drum track 4.

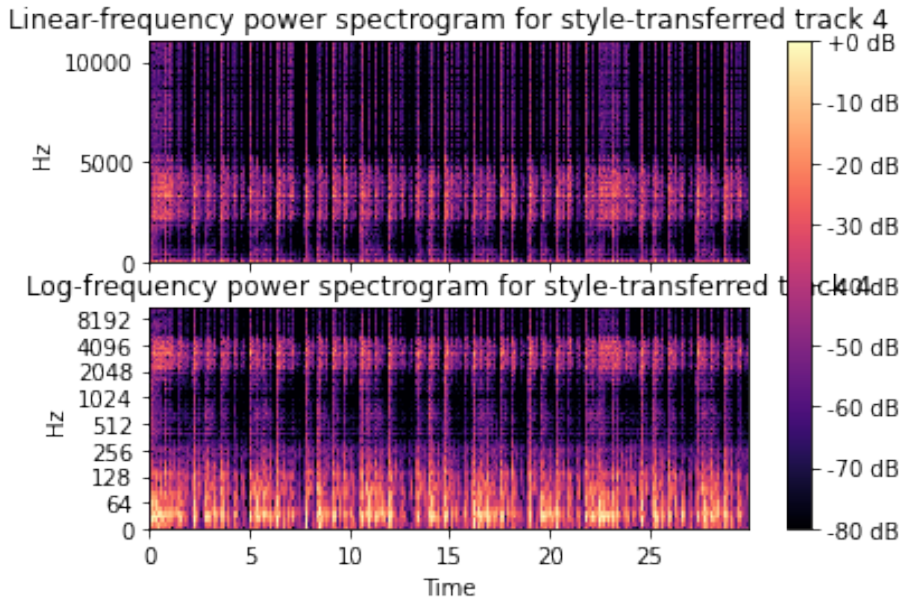


Figure A.8: Spectrograms of style-transferred drum track 4.

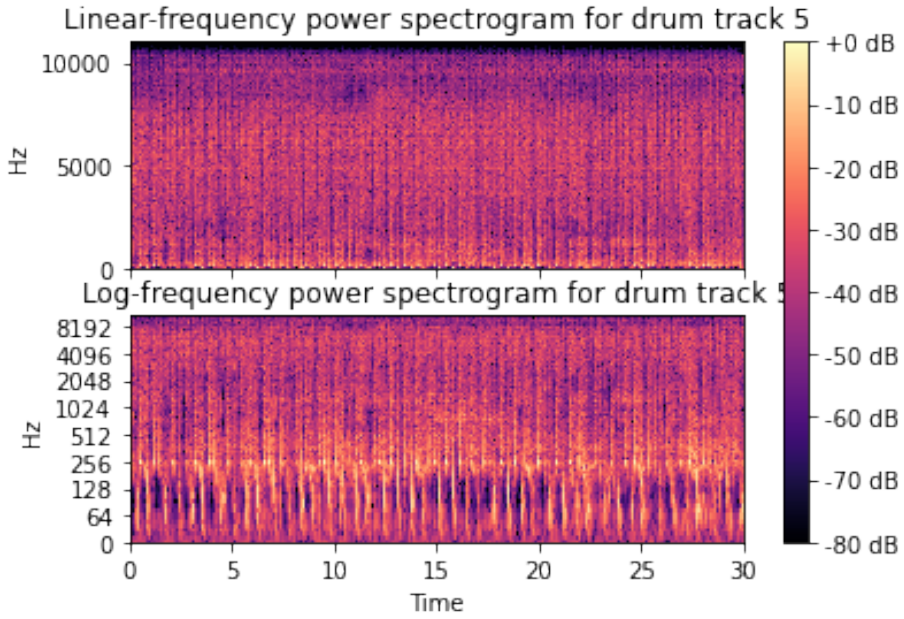


Figure A.9: Spectrograms of original drum track 5.

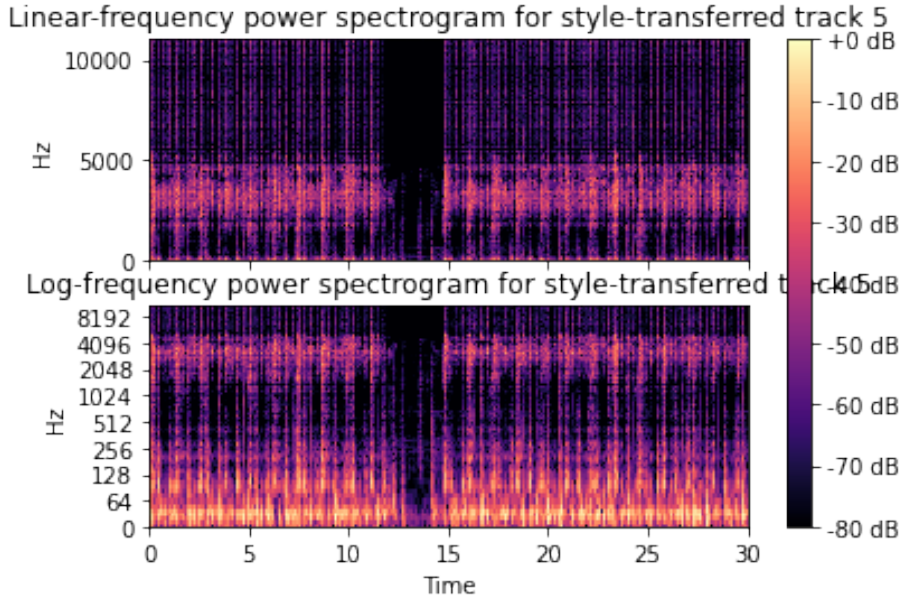


Figure A.10: Spectrograms of style-transferred drum track 5.

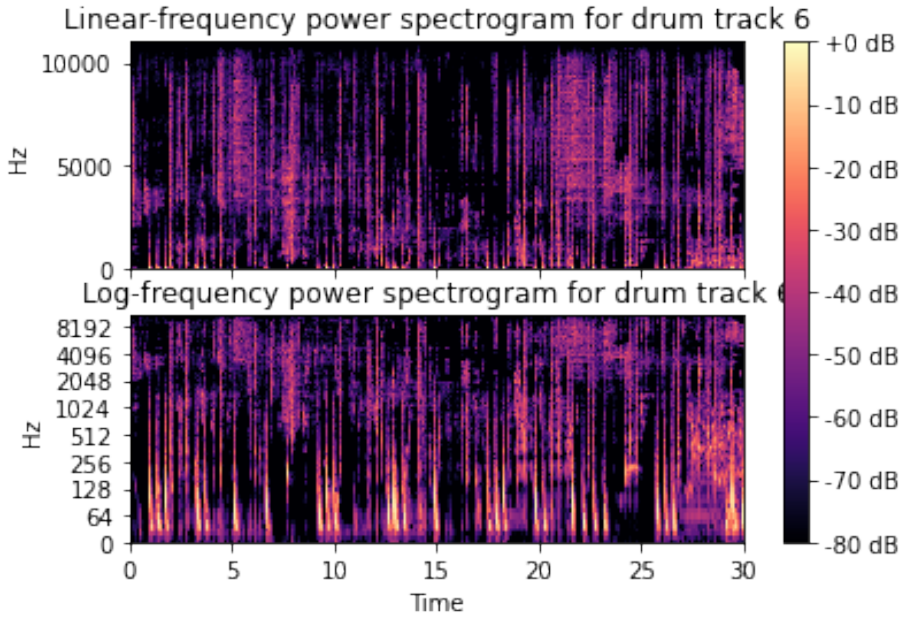


Figure A.11: Spectrograms of original drum track 6.

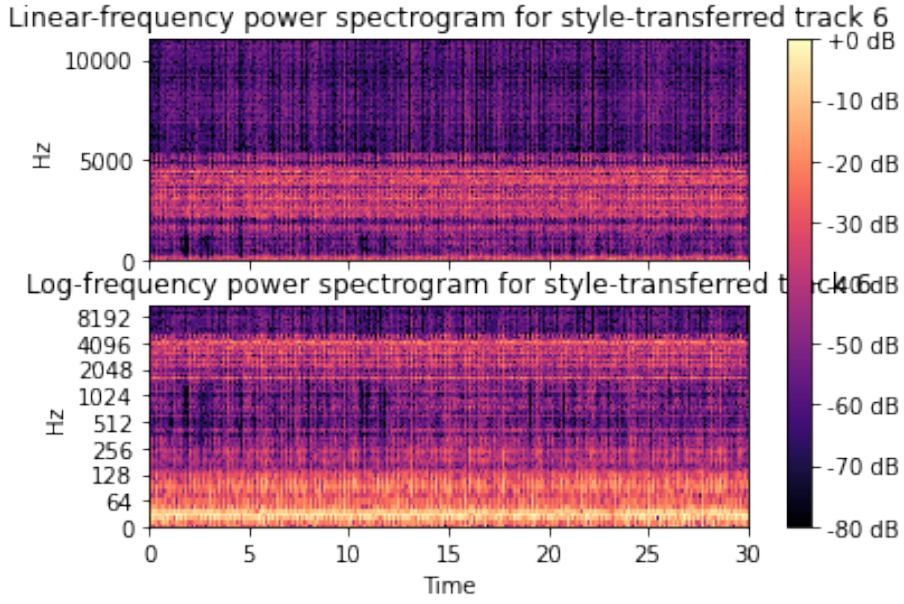


Figure A.12: Spectrograms of style-transferred drum track 6.

Bibliography

[1] Hervé Abdi and Lynne J Williams. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459, 2010.

[2] Matthew Amodio and Smita Krishnaswamy. Travelgan: Image-to-image translation by transformation vector learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8983–8992, 2019.

[3] Adrien Bitton, Philippe Esling, and Axel Chemla-Romeu-Santos. Modulated variational auto-encoders for many-to-many musical timbre transfer. *arXiv preprint arXiv:1810.00222*, 2018.

[4] Dmitry Bogdanov, Nicolas Wack, Emilia Gómez Gutiérrez, Sankalp Gulati, Herrera Boyer, Oscar Mayor, Gerard Roma Trepat, Justin Salamon, José Ricardo Zapata González, Xavier Serra, et al. Essentia: An audio analysis library for music information retrieval. In *Britto A, Gouyon F, Dixon S, editors.*

14th Conference of the International Society for Music Information Retrieval (ISMIR); 2013 Nov 4-8; Curitiba, Brazil.[place unknown]: ISMIR; 2013. p. 493-8. International Society for Music Information Retrieval (ISMIR), 2013.

[5] Konstantinos Bousmalis, Nathan Silberman, David Dohan, Dumitru Erhan, and Dilip Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3722–3731, 2017.

[6] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[7] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. *Advances in neural information processing systems*, 6:737–744, 1993.

[8] Judith C Brown and Miller S Puckette. An efficient algorithm for the calculation of a constant q transform. *The Journal of the Acoustical Society of America*, 92(5):2698–2701, 1992.

[9] Gino Brunner, Andres Konrad, Yuyi Wang, and Roger Wattenhofer. Midi-vae: Modeling dynamics and instrumentation

of music with applications to style transfer. *arXiv preprint arXiv:1809.07600*, 2018.

[10] Gino Brunner, Mazda Moayeri, Oliver Richter, Roger Wattenhofer, and Chi Zhang. Neural symbolic music genre transfer insights. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 437–445. Springer, 2019.

[11] Gino Brunner, Yuyi Wang, Roger Wattenhofer, and Sumu Zhao. Symbolic music genre transfer with cyclegan. In *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 786–793. IEEE, 2018.

[12] Lee Callender, Curtis Hawthorne, and Jesse Engel. Improving perceptual quality of drum transcription with the expanded groove midi dataset. *arXiv preprint arXiv:2004.00188*, 2020.

[13] Chris Cannam, Christian Landone, and Mark Sandler. Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1467–1468, 2010.

[14] Girish Chandrashekar and Ferat Sahin. A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1):16–28, 2014.

[15] Ondřej Cífka, Alexey Ozerov, Umut Şimşekli, and Gael Richard. Self-supervised vq-vae for one-shot music style transfer. *arXiv preprint arXiv:2102.05749*, 2021.

[16] Ondřej Cífka, Umut Şimşekli, and Gaël Richard. Groove2groove: One-shot music style transfer with supervision from synthetic data. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 28:2638–2650, 2020.

[17] Shuqi Dai, Zheng Zhang, and Gus G Xia. Music style transfer: A position paper. *arXiv preprint arXiv:1803.06841*, 2018.

[18] Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. *arXiv preprint arXiv:1802.04208*, 2018.

[19] Jesse Engel, Lamtharn Hantrakul, Chenjie Gu, and Adam Roberts. Ddsp: Differentiable digital signal processing. *arXiv preprint arXiv:2001.04643*, 2020.

[20] Jeff Ens and Philippe Pasquier. Quantifying musical style: Ranking symbolic music based on similarity to a style. *arXiv preprint arXiv:2003.06226*, 2020.

[21] Jeffrey Ens and Philippe Pasquier. Caemsi: A cross-domain analytic evaluation methodology for style imitation. In *ICCC*, pages 64–71, 2018.

[22] Derry Fitzgerald. Harmonic/percussive separation using median filtering. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, volume 13, 2010.

[23] Derry FitzGerald, Eugene Coyle, and Matt Cranitch. Using tensor factorisation models to separate drums from polyphonic music. In *Proceedings of the International Conference on Digital Audio Effects (DAFX09), Como, Italy*, pages 1–4, 2009.

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

[25] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2414–2423, 2016.

[26] Ian Goodfellow. Nips 2016 tutorial: Generative adversarial networks. *arXiv preprint arXiv:1701.00160*, 2016.

[27] Ian J Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *arXiv preprint arXiv:1406.2661*, 2014.

[28] Daniel Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on acoustics, speech, and signal processing*, 32(2):236–243, 1984.

[29] Eric Grinstein, Ngoc QK Duong, Alexey Ozerov, and Patrick Pérez. Audio style transfer. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 586–590. IEEE, 2018.

[30] Brian Gygi, Gary R Kidd, and Charles S Watson. Similarity and categorization of environmental sounds. *Perception & psychophysics*, 69(6):839–855, 2007.

[31] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

[32] Simon Hendry and Joshua D Reiss. Physical modeling and synthesis of motor noise for replication of a sound effects library. In *129th AES Convention, San Francisco*, 2010.

[33] Romain Hennequin, Anis Khlif, Felix Voituret, and Manuel Moussallam. Spleeter: a fast and efficient music source separation

tool with pre-trained models. *Journal of Open Source Software*, 5(50):2154, 2020.

[34] Matthew D Hoffman and Perry R Cook. Feature-based synthesis: A tool for evaluating, designing, and interacting with music ir systems. In *ISMIR*, pages 361–362. Citeseer, 2006.

[35] Sicong Huang, Qiyang Li, Cem Anil, Xuchan Bao, Sageev Oore, and Roger B Grosse. Timbretron: A wavenet (cyclegan (cqt (audio))) pipeline for musical timbre transfer. *arXiv preprint arXiv:1811.09620*, 2018.

[36] Gregory Koch, Richard Zemel, and Ruslan Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML deep learning workshop*, volume 2. Lille, 2015.

[37] Alexander Kraskov, Harald Stögbauer, and Peter Grassberger. Estimating mutual information. *Physical review E*, 69(6):066138, 2004.

[38] Max Kuhn and Kjell Johnson. *Feature engineering and selection: A practical approach for predictive models*. CRC Press, 2019.

[39] Max Kuhn, Kjell Johnson, et al. *Applied predictive modeling*, volume 26. Springer, 2013.

[40] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brébisson, Yoshua Bengio, and Aaron Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. *arXiv preprint arXiv:1910.06711*, 2019.

[41] Olivier Lartillot and Petri Toiviainen. A matlab toolbox for musical feature extraction from audio. In *International conference on digital audio effects*, volume 237, page 244. Bordeaux, 2007.

[42] Tao Li, Mitsunori Ogihara, and George Tzanetakis. *Music data mining*. CRC Press, 2011.

[43] Zhu Liu, Yao Wang, and Tsuhan Chen. Audio feature extraction and analysis for scene segmentation and classification. *Journal of VLSI signal processing systems for signal, image and video technology*, 20(1):61–79, 1998.

[44] Patricio López-Serrano, Christian Dittmar, Yigitcan Özer, and Meinard Müller. Nmf toolbox: Music processing applications of nonnegative matrix factorization. In *Proceedings of the 2019 International Conference on Digital Audio Effects (DAFx-19)*, 2019.

[45] 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. *Proceedings of the AAAI Conference on Artificial Intelligence*, Jul. 2019.

[46] 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. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 1061–1068, 2019.

[47] Dimos Makris, Maximos Kaliakatsos-Papakostas, Ioannis Karydis, and Katia Lida Kermanidis. Combining lstm and feed forward neural networks for conditional rhythm composition. In *International conference on engineering applications of neural networks*, pages 570–582. Springer, 2017.

[48] Bangalore S Manjunath, Philippe Salembier, and Thomas Sikora. *Introduction to MPEG-7: multimedia content description interface*. John Wiley & Sons, 2002.

[49] Josh H McDermott and Eero P Simoncelli. Sound texture perception via statistics of the auditory periphery: evidence from sound synthesis. *Neuron*, 71(5):926–940, 2011.

[50] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and

music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, pages 18–25. Citeseer, 2015.

[51] Cory McKay, Ichiro Fujinaga, and Philippe Depalle. jaudio: A feature extraction library. In *Proceedings of the International Conference on Music Information Retrieval*, pages 600–3, 2005.

[52] Martin McKinney and Jeroen Breebaart. Features for audio and music classification. 2003.

[53] Parag K Mital. Time domain neural audio style transfer. *arXiv preprint arXiv:1711.11160*, 2017.

[54] Dalibor Mitrović, Matthias Zeppelzauer, and Christian Breiteneder. Features for content-based audio retrieval. In *Advances in computers*, volume 78, pages 71–150. Elsevier, 2010.

[55] David Moffat, David Ronan, and Joshua D Reiss. An evaluation of audio feature extraction toolboxes. 2015.

[56] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016.

[57] Marco Pasini. Melgan-vc: Voice conversion and audio style transfer on arbitrarily long samples using spectrograms. *arXiv preprint arXiv:1910.03713*, 2019.

[58] Geoffroy Peeters. Rhythm classification using spectral rhythm patterns. In *ISMIR*, pages 644–647, 2005.

[59] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3617–3621. IEEE, 2019.

[60] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015.

[61] Yi Ren, Jinglin Liu, Xu Tan, Zhou Zhao, Sheng Zhao, and Tie-Yan Liu. A study of non-autoregressive model for sequence generation. *arXiv preprint arXiv:2004.10454*, 2020.

[62] Halfdan Rump, Shigeki Miyabe, Emiru Tsunoo, Nobutaka Ono, and Shigeki Sagayama. Autoregressive mfcc models for genre classification improved by harmonic-percussion separation. In *ISMIR*, pages 87–92. Citeseer, 2010.

[63] William A Sethares. *Tuning, timbre, spectrum, scale*. Springer Science & Business Media, 2005.

[64] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yux-

uan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

[65] Ryan Stables, Sean Enderby, Brecht De Man, György Fazekas, and Joshua D Reiss. Safe: A system for extraction and retrieval of semantic audio descriptors. 2014.

[66] Adam M Stark, Matthew EP Davies, and Mark D Plumbley. Real-time beat-synchronous analysis of musical audio. In *Proceedings of the 12th Int. Conference on Digital Audio Effects, Como, Italy*, pages 299–304, 2009.

[67] Kıvanç Tatar, Daniel Bisig, and Philippe Pasquier. Latent timbre synthesis. *Neural Computing and Applications*, 33(1):67–84, 2021.

[68] Emiru Tsunoo, George Tzanetakis, Nobutaka Ono, and Shigeki Sagayama. Audio genre classification using percussive pattern clustering combined with timbral features. In *2009 IEEE International Conference on Multimedia and Expo*, pages 382–385. IEEE, 2009.

[69] Emiru Tsunoo, George Tzanetakis, Nobutaka Ono, and Shigeki Sagayama. Beyond timbral statistics: Improving music classification using percussive patterns and bass lines. *IEEE Trans-*

actions on Audio, Speech, and Language Processing, 19(4):1003–1014, 2010.

[70] George Tzanetakis and Perry Cook. Marsyas: A framework for audio analysis. *Organised sound*, 4(3):169–175, 2000.

[71] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302, 2002.

[72] George Tzanetakis, Georg Essl, and Perry Cook. Automatic musical genre classification of audio signals, 2001.

[73] George Tzanetakis, Randy Jones, and Kirk McNally. Stereo panning features for classifying recording production style. In *ISMIR*, pages 441–444. Citeseer, 2007.

[74] Dmitry Ulyanov and Vadim Lebedev. Audio texture synthesis and style transfer. 2016. *URL https://dmitryulyanov. github. io/audio-texture-synthesis-and-style-transfer*, 2016.

[75] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[76] Michel Verleysen and Damien François. The curse of dimensionality in data mining and time series prediction. In *Interna-*

tional work-conference on artificial neural networks, pages 758–770. Springer, 2005.

[77] Prateek Verma and Julius O Smith. Neural style transfer for audio spectograms. *arXiv preprint arXiv:1801.01589*, 2018.

[78] Mika Westerlund. The emergence of deepfake technology: A review. *Technology Innovation Management Review*, 9(11), 2019.

[79] Chih-Wei Wu, Christian Dittmar, Carl Southall, Richard Vogl, Gerhard Widmer, Jason Hockman, Meinard Müller, and Alexander Lerch. A review of automatic drum transcription. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(9):1457–1483, 2018.

[80] Zili Yi, Hao Zhang, Ping Tan, and Minglun Gong. Dualgan: Unsupervised dual learning for image-to-image translation. In *Proceedings of the IEEE international conference on computer vision*, pages 2849–2857, 2017.

[81] Kenneth Yip and Gerald Jay Sussman. Sparse representations for fast, one-shot learning. 1997.

[82] Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. Unpaired image-to-image translation using cycle-consistent adversar-

ial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2223–2232, 2017.