NTNU
Norwegian University of
Science and Technology

Master's thesis

2020

Tyler McAllister

Master's thesis

NTNU
Norwegian University of
Science and Technology
Faculty of Information Technology and Electrical
Engineering
Department of Computer Science

Tyler McAllister

Generating Remixed Music via Style Transfer

Using Constant-Q Transform Spectrograms

June 2020

NTNU
Norwegian University of
Science and Technology

NTNU
Norwegian University of
Science and Technology

Generating Remixed Music via Style Transfer

Using Constant-Q Transform Spectrograms

Tyler McAllister

Master of Science in Informatics
Submission date: June 2020
Supervisor: Björn Gambäck

Norwegian University of Science and Technology
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

MSIT Informatics Postgraduate Master's Thesis

Generating Remixed Music via Style Transfer

Using Constant-Q Transform Spectrograms

McAllister, Tyler

June 1, 2020

Abstract

Selective remixing refers to altering an existing musical composition to create something new. The process of remixing audio is commonly intertwined with having a fundamental understanding of music, or music production software — such as digital audio workstations. As research in the roles machine learning can have in audio related transformation and generation tasks continues, there is an indication that systems aiming to remix all types of music without prior musical knowledge from the user could be an effective means of creating content. Existing machine learning research focused on music related generation and transformation is commonly concerned with targeting single instrument or single melody music. As such, five genres of music are used throughout this thesis with the goal being to achieve selective remixing by using image-based domain transfer methods on spectrogram images of music.

With this in mind a system with a pipeline architecture comprised of two independent generative adversarial network models was created. The first model in the pipeline, CycleGAN (Zhu et al. 2017) is responsible for performing style transfer on constant-Q transform spectrogram images. CycleGAN applies features from one of five genres to the spectrogram and passes its result to the next process in the pipeline, CQTGAN which is a modified MelGAN (Kumar et al. 2019) model. The spectrogram output by CycleGAN is turned into a real-value tensor representing a spectrogram and is approximately reconstructed back into audio. Four seconds of music are output by the system in WAV format, and can be concatenated together to recreate a full length music track.

To evaluate the system a number of experiments and a survey are conducted, each concerning the intelligibility of the music and the sufficiency of the style transfer performed. In both cases the audio quality output from the system was considered to be low quality. This was determined to be due to the increased complexity involved in processing high sample rate music with homophonic or polyphonic audio textures. Despite the low quality results, the style transfer performed by the system did appear to perform noticeable selective remixing on most of the music tracks used for evaluation.

Twenty-five unique examples are provided on https://mcallistertyler95.github.io/music-comparison.html, it is recommended to listen to them before reading the rest of this report. Additionally, the code for the implemented system is hosted at https://github.com/mcallistertyler95/genre-transfer-pipeline along with run and training instructions.

Preface

This Master's thesis report is completed for the Master of Science in Informatics programme at the Norwegian University of Science and Technology (NTNU) provided by the Department of Computer Science (IDI).

The thesis was written from September 2019 to June 2020 and is an open project based on computational creativity in music, supervised by Professor Björn Gambäck.

I would like to thank Professor Gambäck for his supervision and feedback throughout the academic year. I would also like to thank Alexander for providing me with helpful advice regarding music theory, Nadia for giving tips for image-based evaluation methods and my family for supporting me throughout my time at NTNU.

Tyler Scott McAllister
Trondheim, June 1, 2020

Contents

List of Figures

List of Tables

Chapter 1: Introduction

Artificial intelligence (AI) and deep learning have been applied to the generation of content in numerous artistic fields, with some of the most prominent results being within the generation of imagery. Comparatively, the generation of audio — in particular music — has had less popularity. Deep learning has proven to be an effective tool for the generation of artistic content such as in Zhu et al. (2017) and Gatys et al. (2016). However, music generation via machine learning has only recently reached significant development — signalled by Benzi et al. (2016) who note that tasks involving audio are restrained by the limited number of freely available audio datasets. Of course, this does not imply the field of music generation is immature or that progress being made is inadequate. Audio generation using machines has progressed greatly since the initial computer generated compositions from the late 50s, as mentioned by Hiller & Isaacson (1979). Deep learning has been applied to diverse music generation tasks (Briot et al. 2017) using a variety of different approaches. However there has been little in-depth research on the creation of effective music remixing systems via deep learning, which is where this thesis aims to contribute.

1.1. Motivations

With the advent of digital audio workstation (DAW) software, such as Ableton Live (Ableton 2019) and GarageBand (Apple Inc 2019), and digital platforms such as YouTube (*Youtube* 2005) and SoundCloud (*SoundCloud* 2007) allowing for anyone to upload their own video content, the creation of musical remixes has increased in popularity dramatically (Fagerjord et al. 2010). The process of remixing music can be defined as altering, or adding content, to an already existing musical composition. This newly created remix shares similarity to its initial composition but sounds audibly distinct. A number of musical genres exist that are heavily built on remixing existing music such as vaporwave, electronic or lo-fi (low fidelity) genres. This leads into the motivations for this thesis, which are to investigate how well less conventional, yet prevalent musical genres that rely on remixing music content can be generated using deep learning. The benefits of having such a system would allow for amateur music artists to effortlessly generate audio waveforms of their own remixes using nothing but audio waveform of an already existing composition. This means users of such a system would not require extensive music theory knowledge or have to learn how to operate DAWs. Additionally, generation of audio waveforms would also contribute to the research of computer music generation, in which audio waveform generation is still a widely researched topic (see chapter 4).

1.2. Project Goal and Research Questions

The overall goal of the thesis was to implement and evaluate a system capable of taking the audio waveform of a song as input and output a transformed audio waveform that represents a differing genre. In essence the system should be capable of remixing audio from one genre to another, with a focus on generating genres of music that are most well known for remixing existing songs. To give a more contextual meaning to ‘music remixing' the definition proposed by (Navas 2010, p. 4) is used, who defines it as: "a reinterpretation of a pre-existing song, meaning that the 'spectacular aura'[1] of the original will be dominant in the remixed version". He goes further to say that there are three types of music remixing - *extended*, *reflexive* and *selective*. Extended remixing is defined as a "longer version of the original composition" containing long instrumental sections. Reflexive remixing is defined as keeping the original track intact but "challenges" the original by introducing its own concepts. For this thesis *selective remixing* was focused on, which is defined as: "adding or subtracting material from the original composition". This definition is broad in scope so instead it was shortened down to achieving a new composition via genre transfer methods (mentioned in section 4.3). Due to the subjective nature of music, the system was be evaluated with well researched methods, such as audio fingerprinting (item 6.4.1), PEAQ analysis (subsection 4.5.3) and mean opinion scores (subsection 4.5.2), inspired by work done by other researchers. For the goal to be satisfied research questions were derived from it, which lead the literature review of the thesis. Additionally, the project goal of the thesis was formalised as:

Project Goal

Create a deep learning system capable of remixing and creating high quality samples of modern genres of music.

To evaluate parts of the project goal separately throughout the experimentation and evaluation stages, it was split into several conditions (Table 1.1) that are required to be met to consider the project goal completed. From the project goal research questions can be inferred that instigated the initial research of the thesis, and acted as the basis for implementation and evaluation that attempted to satisfy the project goal. Table 1.2 shows the research questions, followed by a more detailed analysis of each question.

Conditions	
C1.	Deep learning must be one of the key characteristics of the implemented system.
C2.	The system must output audio waveform.
C3.	Selective remixing via genre transfer must be performed on audio.
C4.	The genres of music used must be modern and outwith the standard genres used in existing state-of-the-art systems.
C5.	The audio generated must be high quality.

Table 1.1.: Conditions to be met throughout development

[1] Navas explains that the spectacular aura of music is its cultural value created via its contribution and sensory impact to the listener.

Research Questions	
R1.	How are raw audio waveforms generated in other music generation systems?
R2.	How can selective remixing be performed using deep learning?
R3.	Can high quality convincing remixed music generated via deep learning methods be reasonably evaluated?

Table 1.2.: Research questions

Research Question R1

How are raw audio waveforms generated in other deep learning music generation systems?

R1 is concerned with how existing state-of-the-art solutions create raw audio waveform, an aspect which is necessary to satisfy conditions **C1** and **C2**. This question also closely ties into the succeeding research question - **R2**, as if selective remixing can be performed then it must be performed in a way that allows for the generation of audio waveform via deep learning.

Research Question R2

How can selective remixing be performed using deep learning?

R2 also highlights a key area of this thesis and is associated with conditions **C3** and **C4**. As deep learning is the primary focus of the system significant research was performed to discover which deep learning methods are currently being used for the creation of remixed music. Current methods of creating music were also highly applicable to this system, including those that did not involve selective remixing as a research goal. Ultimately this research question is concerned with the availability of existing software or theoretical solutions, and how these could be integrated into the system proposed by the thesis. Therefore adequate research into the state-of-the-art is performed, as shown in chapter 4 and chapter 7 which aim to answer both **R1** and **R2**, and supplement a system that can be evaluated according to **R3**.

Research Question R3

Can high quality convincing remixed music generated via deep learning methods be reasonably evaluated?

R3 is related to condition **C5** as finding an evaluation method for music generated by deep learning systems allows for the effectiveness of the system to be determined. First, the meaning of “convincing music” is chosen. One of the Oxford English Dictionary’s definitions of “convince” is (Oxford University Press 2020*a*):

> “To cause (a person) to admit, as established to his satisfaction, that which is advanced in argument; to bring to acknowledge the truth of; to satisfy or persuade by argument or evidence. In passive, To be brought to, or to have, a full conviction; to be firmly persuaded.”

As such, convincing music is defined as music presented to a person that becomes reasonably persuaded that the content they were presented with (in this case listening to the sound generated from a computer system) could be classified as music. Additionally “high quality” music is defined as music that is clear of audible imperfections, such as artefacts.

1.3. Contributions

The most noteworthy contributions this thesis makes are:

1. The creation of a system capable of outputting genres of music that have been selectively remixed.
2. An investigation into the generation of audio waveform via deep learning.
3. An in-depth evaluation of state-of-the-art deep learning music generation systems.
4. Insight and investigation into how generated music can be objectively and subjectively evaluated.

1.4. Thesis Structure

Essential background material into music and signal processing and deep learning are present in chapter 2 and chapter 3. Information from these chapters contextualises many of the aspects discussed in the succeeding chapters.

Chapter 4 discusses the current state-of-the-art literature concerning audio waveform generation, genre style transfer, spectrogram reconstruction methods and evaluation approaches for genre classification and audio quality.

In chapter 5 an exploration and evaluation of available datasets was performed, with one being chosen to train the system.

Chapter 6 presents the basic system architecture that was created to plan the development of the implemented system. Additionally, the software tools utilised throughout development and experimentation are discussed.

Chapter 7 describes experiments performed on the system that were used to build its final architecture and evaluate its performance. Similarly, chapter 8 presents the results from a survey used to determine the audio quality and success of genre transfer.

Following this, in chapter 9 the system in its entirety is evaluated based on the results from the survey and experiments, along with the limitations present in the current system.

Finally, chapter 10 concludes the thesis by describing the work performed and advising how the work could be improved in future iterations.

Chapter 2: Music and Signal Processing

An understanding of basic music theory and audio signal processing was necessary for the creation of the system and reviewing existing research (chapter 4). Throughout this section an overview of each of the relevant music theory is discussed. Concepts that directly relate to the implementation of the system are also touched on lightly within this chapter and the subsequent chapter - chapter 3. In particular, the information present in subsection 2.5.2 and subsection 2.5.3 details elements that were implemented in the system described in chapter 6.

2.1. Basic Music Theory

Basic theoretical concepts of music are necessary to understand sound processing. While the basis for the implemented system's inputs and outputs were raw-audio waveforms of produced musical scores, there were necessary elements within the audio signal processing field that had to be understood rather than music theory itself. Regardless, the fundamentals of how musical scores can be represented and performed is helpful to explain various elements of the system. Timbre (subsection 2.1.4) and musical texture (subsection 2.1.5) are among some of the key concepts referred to in future sections.

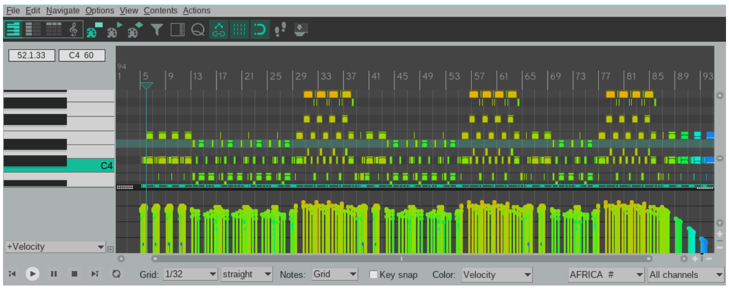
2.1.1. Pitch

Pitch describes how high or low the frequency of a note is. A high pitch is described as a sound wave with a high frequency and short wavelength while the opposite is true for a low pitch. In the field of music theory the degree of pitch a sound has is commonly represented as from letters A to G which make up the natural, sharp and flat notes used in musical staves and between octaves. Pitch is vital to one's perception of music, speech and sound source segregation. In the field of music pitch can be described as:

> "inherent to the concepts of melodies and chords, and is what allows us to perceive a sound as musical". (Oxenham (ed.) & Oxenham 2005, p.1)

2.1.2. Musical Notation

Musical score, or musical notation, of audio can be achieved in various ways. Globally, sheet music using the pitch notes mentioned in subsection 2.1.1 has become the most popular way of transcribing music. Although there are others, such as tablature (Weiss & Taruskin 2007) that have gained popularity for instruments such as guitars online (Chesney 2004). Piano-roll styled notations have also gained popularity in digital music based programs, to allow users to become familiarised with music without needing to learn sheet music notation. Such programs commonly show a virtual keyboard on-screen which display notes and the duration they should be played on a rolling tape. Figure 2.1 shows an example of a MIDI file being edited using a piano-roll type display in the Reaper digital audio workstation (Cockos 2020).



Grid: 1/32 straight
Notes: Grid
□ Key snap
Color: Velocity
AFRICA #
All channels

Figure 2.1.: Piano Roll in Reaper

2.1.3. Melody

Closely related to pitch, melody in its most simplest description is:

> “A series of single notes arranged in a musically expressive or distinctive sequence” (Oxford University Press 2020*b*)

Such notes are described as being in the ‘melodic line’. Not all notes can be described as being part of the melody, others can be added in tandem to the melody to bring extra complexity to the composition outside of this melodic line.

2.1.4. Timbre

All information within sound, outside of the pitch, duration and volume can be described as the timbre (Abbado 1988, p.2). Timbre is a key aspect within all music and can be understood via a comparison between two instruments. Playing a note from a piano and the same note from a guitar for the same length of time at a same intensity will still have very audible differences produced from each instrument. This unique property of sound describes the concept of timbre. Ultimately, timbre allows a listener to distinguish different types of sound outside of their pitch, volume and duration. Within the field of deep learning modifying the timbre of audio has been a well researched topic (Briot et al. 2017).

2.1.5. Musical Texture, Density and Range

Describing the complexity of a musical composition is commonly done by referring to the concept of range, density and texture. Density and range refer to the high level features of a composition. For example, a composition’s density can be described as “thick” if there are multiple instruments or voices present while it would be regarded as “thin” if only one instrument were playing throughout the composition. Range refers to the interval between the highest and lowest tones within the composition, a composition can be described as having a “narrow” range if it has small intervals and “wide” if they are large.

Musical texture furthers the concepts of range and density by giving more well-defined categories for music to fit into. Four texture types for compositions are described by

Benward & Saker (2009):

- Monophonic - In which only a single melodic line is played in the composition. Typically one instrument or singer is present.
- Polyphonic - Two or more melodies that are independent of each other but are being played at the same time.
- Homophonic - A composition consisting of a melody that is intended to be the most prevalent sound within the composition that is supported by an associated accompaniment.
- Homorhythmic - A composition with similar rhythmic material in all parts.

Homophonic is the most common texture used in modern popular music (Benward & Saker 2009), although texture can change throughout a composition.

2.2. Music Remixing and Remix Culture

After rising in popularity in the seventies, (Navas 2010) the concept of 'remixing' was popularised within the music industry, in which existing musical compositions were altered and presented as brand new content. Music remixing can be defined in simple terms as "a reinterpretation of a pre-existing song" from the larger definition stated in section 1.1. With the advent of the internet, availability of remixed music via media sharing websites like YouTube, has caused remix culture (Cheliotis & Yew 2009) (also referred to as sampling culture) to become a significant phenomenon. Remix culture can be defined as:

> "global activity consisting of the creative and efficient exchange of information made possible by digital technologies" (Navas 2010, p. 3):

Remix culture does not solely encompass remixed music. Video content and artistic imagery are some of the few creations that remix culture has lead to within the current generation of the internet (Fagerjord et al. 2010). Due to the initial hurdle learning a new piece of software can impart on a user, improving the ease of participating in remix culture was a primary motivation for this thesis.

2.3. Digital Audio

Audio stored digitally has a number of differences from its analogue form. Since the proposal for this thesis' goals and research question concerns audio quality some focus should be given into detailing how digital audio is stored and how quality is achieved.

Analogue audio that is stored digitally is created via a digital audio encoder that transforms the analogue signal into a digital format which is then decoded back into analogue audio upon being played. Audio encoding is a challenge of maintaining quality of the original signal while reducing the amount of information needed to represent the original signal to reduce processing time and complexity (Bosi & Goldberg 2002, p. 6). All audio encoding is done by sampling frequencies from the original input signal at specific times. Analogue sound is a continuous-time signal which must be transferred into a discrete-time signal. To do this a sampling-rate is chosen for audio encoding that defines the number of samples to be recorded from an continuous signal. In general, a higher sampling rate will result in more accurately captured audio quality. Typically compact disk (CD) format audio stores audio as a stereo signal with a sampling rate of 44.1kHz which is stated by (Bosi & Goldberg 2002, p. 8) to be "adequate to preserve frequency content of up to 22.05kHz".

Digital music can be stored in a variety of formats to be played by software. Common formats include MPEG Audio Layer-3 (MP3), Waveform Audio File Format (WAVE) and Free Lossless Audio Codec (FLAC) which all support multiple audio channels and sample rates.

2.4. Evaluating Music and Genre Classification

As a subjective art form, qualitative analysis may be considered the most applicable type of evaluation suited to music, although objective measures do exist and have been applied to sound quality (subsection 4.5.3). Other objective measurements can be made for music itself (Romney et al. 2016) but none have revealed how the sound can be perceived and understood on an artistic level by the listener. As stated by research question **R3**, the evaluation of musical genres is paramount to reaching the goal of this thesis. Most evaluation methods for music are performed to categorise them into a genres via the use of music information retrieval (MIR) or AI systems systems.

Berenzweig et al. (2003) proposed a number of artificial intelligence based similarity measures, which they named acoustic measures, for comparing the music of multiple artists to create similarity matrices. These measures are:

- Using a neural network trained on mel-frequency cepstral coefficients derived from short segments of audio to identify 12 different genres and the gender of the singer.
- Applying a Gaussian mixture model (Reynolds 2009) to short segments of audio represented as data points within the model to cluster the data into artists with similar songs.

They also investigated using subjective sources to create similarity matrices between artists. Some of the most prominent measures proposed by the authors were:

- Surveys in which participants were given a target artist and were asked "Which of these artists is most similar to the target artist?" and given a selection of ten artists to choose from. The authors noted that despite having 22,000 responses only 7.5% of artists were directly compared as being similar.
- Expert opinions from collections of related artists from the music review website `www.allmusic.com` were used as an alternative to a large scale survey. This method allowed for more efficient data collection and managed to reach 87.4% artist pairs in comparison to the previous method.
- Co-occurrence of songs within a publicly available online user created playlist were utilised under the impression that songs within these playlists would be closely related in terms of genre.

Overall Berenzweig et al. (2003) concluded that the their subjective measures were more effective than the acoustic measures due difficulties in representing temporal structure information within their AI models.

Lefaivre & Zhang (2018) investigated adapting the a priori association algorithm (Toivonen 2010) to use music, represented as vectors, containing acoustic features from mel-frequency cepstral coefficients (MFCCs) defined by Xu et al. (2005). By using the MFCC representations they were able to use the algorithm to attempt to identify music tracks into one of six genres. Competitive results were created by this method of classification but the authors note that genres like pop music were frequently misclassified.

Seyerlehner et al. (2010) compared genre classification algorithms to human performance using the same dataset for both methods. With human participants they performed a listening test where each listener was asked to classify 190 songs into one of nineteen genres from thirty second segments of each song. Around 55% of the participants were able to classify songs correctly from the dataset. Most mistakes came from confusion between definitions of genres. For example blues and jazz music were often confused, or country and folk music. Five machine learning based methods (two nearest neighbour classifiers and three SVM classifiers) were used for comparison to the human participants. Ultimately, they noticed that human participants were at least 10% more accurate at making correct decisions than the machine learning methods. However the authors did mention that the ground truth definitions of the data could have impacted classification accuracy. This highlights some of the issues surrounding genre classification. The authors state that:

> “there will always exist some annotation errors due to the inconsistency of the genre taxonomy itself” (Seyerlehner et al. 2010, p.11)

Meaning music genre is not a well defined taxonomy for classification. It is not possible to fit all songs into a genre because genre taxonomy is loosely defined and ever-changing. There will always be a degree of erroneous labelling when attempting to fit a song into one, or many, genres.

2.5. Digital Audio Signal Processing

Digital audio signal processing (DSP) is the process of using computational methods to make modifications to sound signals. Music, speech and environmental sound processing are some of the numerous signal processing tasks that have become more widely researched with the emergence of deep learning (Purwins et al. 2019). In addition, a number of traditional DSP methods (Gold et al. 2011) aid greatly in deep learning related tasks that focus on these types of signal processing. Fourier transforms, sound recognition and audio synthesis are all DSP methods commonly used in state-of-the-art research for audio (chapter 4), they are covered in this section.

2.5.1. The Discrete Fourier and Short-Time Fourier Transforms

Short-time Fourier transforms (STFT) are complex-valued transforms between frequency representations of signals and time domain representations. The discrete Fourier transform (DFT) algorithm produces a finite spectrum of a continuous finite signal, as defined in Heideman et al. (1985):

$$X_k = \sum_{n=0}^{N-1} x_n e^{-i2\pi \frac{kn}{N}} \tag{2.1}$$

To compute the DFT of a signal it is first run through a window function to be represented as a periodic signal and is processed via the DFT in which discrete samples of the signal are captured. A spectrum is the result produced by the DFT, making it an incredibly useful algorithm for signal processing. Despite this initial effectiveness, the DFT algorithm is computationally slow taking N^2 operations due to its brute force nature, so another algorithm named the Fast Fourier Transform (FFT) is applied. FFT is similar to DFT but represents the input signal as a two-dimensional array rather than one-dimensional, as in DFT. As a result the FFT algorithm is $Nlog(N)$ in complexity, a vast improvement compared to DFT. STFT extends this algorithm even further by applying FFT using a window sliding method which produces a 2D matrix of the frequency against time - this representation is known as a spectrogram.

2.5.2. Spectrogram

Initially known as the sound spectrograph when proposed by Koenig et al. (1946) and is occasionally referred to as the magnitude STFT. A spectrogram is the squared magnitude of the STFT signal and contains the intensity plot of frequencies over time:

$$Spectrogram = |X(k)|^2 \tag{2.2}$$

A colour intensity is typical given to show the volume of the frequency at a given time. Due to being the magnitude of the STFT, spectrograms are a lossy transformation because they lose the phase information that is preserved by the STFT representation. This can make accurate signal reconstruction impossible via inversion methods. Instead reconstruction algorithms such are used to recreate the initial audio signal from a magnitude spectrogram. Applications of spectrograms include speech recognition, audio synthesis, pitch modulation and seismology. An example of a spectrogram with frequency on the x-axis and time on the y-axis can be seen in Figure 2.2, note that the representation looks very sparse which can make it difficult to use in some computer vision tasks. Due

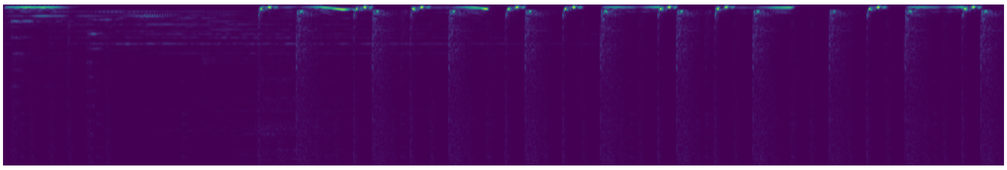


Figure 2.2.: Spectrogram Representation

to being two-dimensional images, spectrograms are popular with image-based machine learning techniques such as those mentioned in section 3.6. However, magnitude spectra do not show much information visually in comparison to other spectrogram types. Instead it is common to alter the magnitude spectrum to represent a different scale on the y-axis of the spectrogram, such as decibels.

Log-scaled Spectrogram

Log-scaled spectra, or spectrograms, are used to display more human readable and machine interpretable information compared to magnitude spectra by representing the frequency in the decibel scale. The formula:

$$y_{db} = 20log_{10}(y) \tag{2.3}$$

represent decibels on their y-axis. Figure 2.3 shows a log-scaled spectrogram created from the magnitude spectrogram in Figure 2.2.

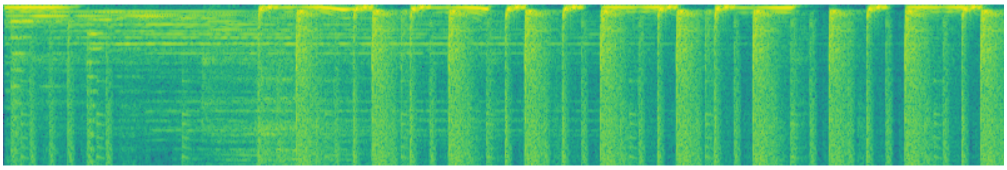


Figure 2.3.: Log-scaled Spectrogram Representation

Mel-scaled Spectrogram

Mel-scaled spectrograms, or mel-spectrograms, are another representation of magnitude spectrograms, similar to log-scaled spectrograms. Instead of using decibels, these spectrograms use the mel scale - a scale initially proposed by Stevens (1937) that describes a subjective scale of equal pitch distance measures decided upon by groups of human listeners. The formula proposed by O'Shaughnessy (1987) to convert a frequency f to mels is:

$$m = 2595log_{10}(1 + \frac{f}{700}) \tag{2.4}$$

Mel-spectrograms are visually similar to magnitude STFT spectrograms, as described in Figure 2.5.2, but instead of representing decibels on the y-axis they represent the mels of a signal instead. Within recognition and audio synthesis tasks this form of spectrogram has gained a large amount of popularity (Prenger et al. 2018, Briot et al. 2017, Vasquez & Lewis 2019, Shen et al. 2017, Wang et al. 2017). While no extensive comparison of spectrogram representations has been done with audio synthesis tasks, researchers working with audio recognition have found that mel-spectrograms perform much better than their counterparts (Huzaifah 2017, Choi et al. 2017) meaning there may be some justification to using them for audio synthesis tasks. Despite their widespread use within the deep learning field, this form of spectrogram cannot be inverted back into a waveform using typical DSP methods when compared to a magnitude STFT spectrogram.

Figure 2.4 shows a mel-spectrogram representation of the same song used in Figure 2.2 and Figure 2.3

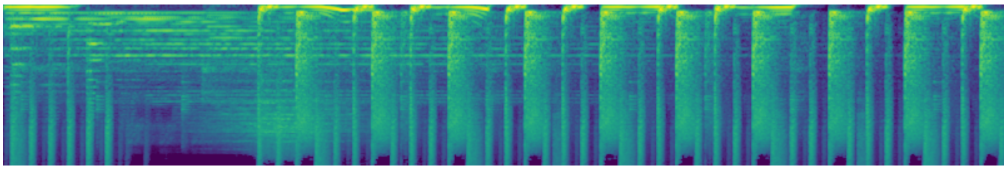


Figure 2.4.: Mel-scaled Spectrogram Representation

Constant-Q Transform Spectrogram

The constant-Q transform (CQT) is another transform (Brown 1991) that is focused on use for musical audio signals. While closely related to discrete Fourier transforms, the CQT differs in that its creation was motivated by finding a better way to represent music signals. In a comparison to DFT, Brown (1991) states that DFT:

> “yields components which do not map efficiently to musical frequencies”. (Brown 1991, p.425)

In contrast to this, the CQT transfers an audio signal into a time-frequency scale with logarithmically spaced centre frequencies for each frequency bin, while DFT uses a constant spacing between its frequencies. By spacing frequencies in this way the CQT is capable of being mapped closely to the equal temperament scale used in western music, and allows for improved frequency resolution at lower frequencies while giving improved time resolution at higher frequencies. The equation used to create the CQT of a frequency signal is defined as follows (Brown 1991, p.427):

$$X[k] = \frac{1}{N[k]} \sum_{n=0}^{N[k]-1} W[k,n]x[n]exp^{\frac{-j2\pi Qn}{N(k)}} \tag{2.5}$$

This calculation differs from Equation 2.1 in that the frequency of a kth component is $(2^{1/24})^k f_{min}$ (Brown 1991) where f_{min} represents the smallest frequency to be mapped in the transform. And $2^{1/24}$ gives quarter-tone spacing, allowing for simultaneous notes to play within the twelve-tone equal temperament scale of western music.

Essentially, musical frequencies are more accurately represented by the CQT when compared to mel and DFT related transforms. Despite this noticeable improvement for musical representation, the CQT has not been a popular transform for use in signal processing fields. Schörkhuber & Klapuri (2010) state that there are three reasons why the CQT has not been favoured in comparison to DFT in such fields:

1. Compared to the DFT it is more computationally expensive to calculate.
2. It cannot be reconstructed back into audio as easily as the DFT, which can be inverted back into a perfect reconstruction of its original input signal.
3. Its data structure is more complex to work with in comparison to DFT.

However, Schörkhuber & Klapuri (2010) go on to present an invertible CQT transform that can be reconstructed back to an approximated audio signal. Despite its lack of popularity, the CQT has still seen use in some projects involving audio and speech transfer as shown in subsection 4.3.5. An image of a CQT log-magnitude spectrogram is shown in Figure 2.5.

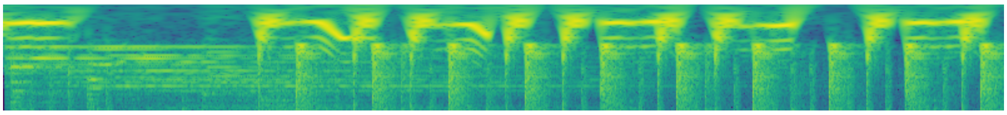


Figure 2.5.: CQT Spectrogram Representation

2.5.3. Signal Reconstruction and Audio Synthesis

For systems that aim to work with time-frequency representations, to reconstruct an audio signal to generate a waveform there must be some decision made on how to invert these representations back into audio that sounds intelligible and produces an expected sound. The inverse of STFT representations (Figure 2.2), are fully invertible back to their original signal because they retain frequency, amplitude and phase information of the signal. STFT is the basis for most other types of spectrogram, in particular the mel spectrogram and the log-magnitude STFT spectrogram. These types of spectrogram represent the original signal on a different scale. For example the log-magnitude of an STFT spectrogram represents the decibel scale. Spectrograms that have changed their scale in these ways lose the phase information of the signal, and thus cannot be easily inverted back into their original audio. Instead of inversion, other methods are used to create an approximate signal from these types of spectrogram. This can lead to significant quality loss in the audio if done naïvely, although some loss in quality should be expected because a perfect reconstruction is not possible.

Griffin-Lim

The Griffin-Lim algorithm, from Griffin & Jae Lim (1984), iteratively creates artificial phase information which is derived from a magnitude spectrum (Figure 2.3). The algorithm will converge towards the estimated phase, creating an approximated phase layer that can be used to reconstruct a waveform from the spectrum. Griffin-Lim is known to give intelligible results and has been used throughout research involving audio synthesis that use spectrograms as an intermediate representation. The original Griffin-Lim algorithm can be applied to any type of spectrogram but is never guaranteed to create intelligible audio, primarily being used on magnitude STFT spectrograms to achieve good results.

Inversion via Deep Learning

Spectrograms are a well documented intermediate representation for audio, used in various deep-learning related tasks. STFT spectrograms are the only representation that can be perfectly reconstructed back into its original audio, meaning high quality reconstructions are possible. For all other representations, Griffin-Lim is a viable solution but is often not enough on its own to reconstruct audio with sufficient quality. Furthermore, mel and CQT spectrograms contain much richer features compared to STFT, and assort sound against scales that are more representative of human hearing. For these reasons they are often the preferred representation for audio in tasks involving audio recognition due to their higher performance over STFT (Huzaifah 2017).

Audio synthesis tasks that use spectrograms can reach a hurdle in which audio reconstruction is needed but algorithms like Griffin-Lim are not enough to create high quality audio. Due to this a number of deep-learning models have been proposed for spectrogram reconstruction, namely the creation of models representing voice vocoders, and generative adversarial networks. Spectrogram reconstruction methods used in deep learning are specified in greater detail in chapter 4.

Chapter 3: Deep Learning

Deep learning encompasses a variety of applications, from computer vision, business analysis and recommendation systems to natural language processing. No differently from other fields, deep learning has also found a place within the audio and music domain. Music and speech synthesis, recognition and domain transfer have all had deep learning play a role in their development. This chapter focuses on explaining deep learning techniques and models that were explored for use within the implemented system, or covered in the literature review. The reader is assumed to have a basic understanding of deep learning concepts.

3.1. Audio Synthesis

Closely linked to what was discussed in subsection 2.5.3. Approximate inversion of mel spectrogram representations was achieved via Google DeepMind's WaveNet (van den Oord et al. 2016), which was used as a vocoder that can be trained on audio paired with their spectrogram equivalent (Shen et al. 2017).

Nvidia's WaveGlow (Prenger et al. 2018) is another deep learning flow-based network that was created to generate high quality audio from log-scaled and mel-scaled spectrograms, in both speech and music synthesis tasks with a similar training method.

These models have distinct popularity within audio synthesis tasks (chapter 4) as they allow for good training data (richer features present in log-magnitude and mel spectrograms) without compromising audio quality.

3.2. Feedforward Neural Networks

Well known within artificial intelligence research, feedforward neural networks (or simply, neural networks) were among the first of the main connectionist models (McCulloch & Pitts 1943) that have been used for numerous applications since their inception. Their creation also paved the way to the development of other connectionist models, such as generative adversarial networks and convolutional neural networks. Modelled after the biological process of neurons firing within the human brain, a neural network is composed of activation nodes, also known as neurons, that are fully connected via weights. Figure 3.1 below shows the structure of a feed-forward neural network. Neural networks are trained iteratively on large amounts of data that allows their weights to converge to the desired solution.

3.3. Convolutional Neural Networks

While primarily used for computer vision related tasks, convolutional neural networks (LeCun et al. 1989), also known as CNN, have also been applied to audio in a range of audio related projects (Briot et al. 2017, Huzaifah & Wyse 2019). Architecture within a CNN differs from a traditional feedforward neural network, in that they introduce the concepts of convolutional and pooling layers. Convolutional layers perform a process called filtering which involves 'running', or convolving, a matrix of weights across the

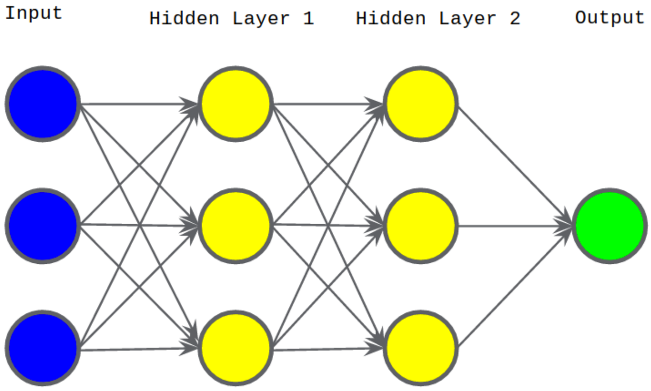


Figure 3.1.: Basic Feedforward Network Architecture, modified from Google's Machine Learning Crash Course, released under CC BY 4.0

input data (e.g. an image). The matrix is multiplied against the input data and summed together to create a single value. This process continues until the entirety of the input data has been convolved and a new output has been created, known as a feature map. This feature map is passed onto a non-linear activation layer which uses an activation function like those found in feed-forward neural networks. Typically, the ReLU activation function (Agarap 2018) is used to eliminate negative values from the feature map and to mitigate the vanishing gradient problem (Hu et al. 2018), which negatively impacts training. Finally, a pooling layer can be used to reduce the dimensions of the data to improve computation time and reduce complexity without compromising the quality of the network training.

3.4. Generative Adversarial Networks

Focusing on the creation of new content, generative adversarial networks (GANs), first proposed by Goodfellow et al. (2014), are a suitable choice in the deep learning field for use in content creation. These networks are capable of creating entirely new data by utilising two network models within their structure that compete against each other in a minimax-like game. Mathematically, a GAN can be represented by the following (Goodfellow et al. 2014, p.3):

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}\,(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z} \sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))] \tag{3.1}$$

Which represents the minimax game between G (the generative model) and D (the discriminative model) with the value function $V(D,G)$ in which the generator G aims to minimise the function while the discriminator D aims to maximise it. $D(\boldsymbol{x})$ is the discriminator's probability that the provided data x is real. $\mathbb{E}_{\boldsymbol{x} \sim p_{\text{data}}(\boldsymbol{x})}$ is the expected value over all instances of data. $G(\boldsymbol{z})$ is the output of the generator G, which is given random noise as input - z, while $D(G(\boldsymbol{z}))$ is D's probability estimate of how 'real' a given output of the generator is. $\mathbb{E}_{\boldsymbol{z} \sim p_z(\boldsymbol{z})}$ represents the expected value over all fake outputs of $G(\boldsymbol{z})$.

Evidently, the purpose of the generator is to create fake data that successfully fools the discriminator into believing it is part of the training data. Consequently the discriminator's purpose is to give an estimation as to whether a sample has been created by the generator or is part of the training data in an attempt to minimise the generator's successful samples. Theoretically, after suitable training the GAN should converge to the point where neither the discriminator nor generator are capable of reducing the loss of the other, meaning the network would be capable of outputting fake data that is convincingly similar to that in the training data. A high level overview of typical GAN architecture is shown in Figure 3.2.

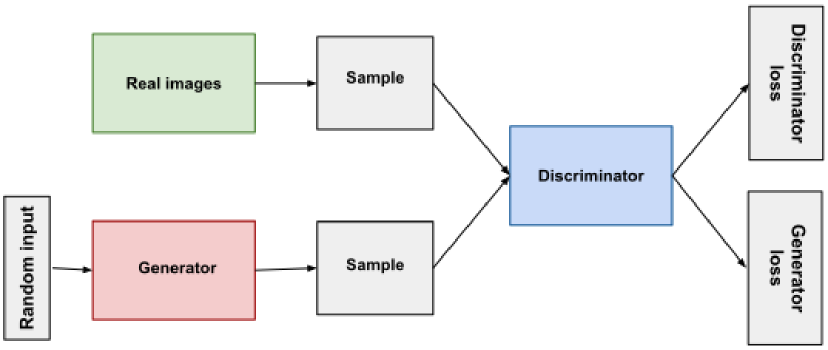


Figure 3.2.: Basic GAN Architecture from Google's Machine Learning Crash Course, released under CC BY 4.0

Since their inception GANs have seen usage for a number of engaging tasks, such as the generation of fake, yet accurate, looking faces of non-existent people (Karras et al. 2018), upscaling images while reducing the loss of quality (Karras et al. 2017) and domain transfer and translation tasks (Zhu et al. 2017).

3.5. Conditional Generative Adversarial Networks

From section 3.4, the type of GAN described is unconditional, meaning it is incapable of 'controlling' which data is generated. For example an unconditional GAN capable of generating different letters of the alphabet would not be able to oversee which letter is generated. Because of this, the creation of a conditional GAN (cGAN) was shortly conceived after their unconditional variation by Mirza & Osindero (2014). Such GAN models use labeled data to aid in training and controlled generation of examples.

By introducing y as additional information, such as a class label, to the GAN the original GAN model (Equation 3.1) can be adapted to (Mirza & Osindero 2014, p.3):

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x}|\boldsymbol{y})] + \mathbb{E}_{\boldsymbol{z}\sim p_z(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z}|\boldsymbol{y})))] \quad (3.2)$$

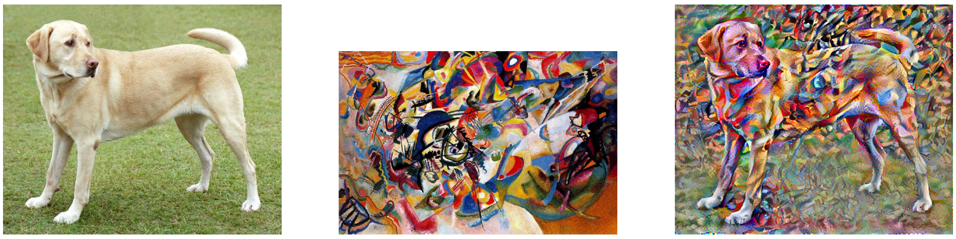
This new information y is represented as an additional layer to both the generator and discriminator which makes both model a probability distribution containing the class label. Conditional GANs have also contributed to various domains, with most of the networks mentioned at the end of section 3.4 making use of conditional generation.

3.6. Neural Style Transfer

Neural style transfer (NST, or style transfer) is a technique that begun as an approach to using convolutional neural networks to extract the *style* and *content* from an image and apply the extracted style onto another image (Gatys et al. 2016). To understand the concept of NST the concept of style and content needs to be discussed, as well as how these representations are inferred from an input image. The content of an image can be defined as the objects, or 'physical' properties and scenery that are present within it, while the style can be defined as the colours, basic shapes and texture contained within an image. Overall, the process of NST works by taking an input image p and a style image a. The network will take the input image p and extract the content representation of the image - $C(p)$. Similarly, the style image a is also fed through the network and its style representation is extracted - $S(a)$. To create a new output image x with the content representation of p with the style representation a such that $C(x) = C(p)$ and $S(x) = S(a)$. The loss function to be minimised by the network is defined by (Gatys et al. 2016, p.2419) as:

$$L_{total}(p, a, x) = \alpha L_{content}(p, x) + \beta L_{style}(a, x) \tag{3.3}$$

where α and β are the weighting factors for content and style reconstruction.



(a) 'Yellow Labrador Looking' from Wikipedia Commons

(b) Wassily Kandinsky's Composition VII

(c) Result from Google's Tensorflow Tutorials, released under CC BY 4.0

Figure 3.3.: Neural Style Transfer Example

Figure 3.3 shows how the information extracted from the separate image domains is obtained using the network model created by Gatys et al. (2016). The content of the image (a) is the dog which is retained by the model, while in image (b) Wassily Kandinsky's Composition VII the style is derived (colour, shapes and texture). After training the final result, image (c), is created as a combination of the content features from image (a) having a filter like effect containing the style of image (b) being applied.

3.7. Audio Style Transfer

The concepts of NST are specified as extracting content and style information from two separately given inputs, and applying the extracted style to the content to create a new element. Such concepts have been applied to audio in various ways, although the style and content are less well defined due to the number of differences between image and audio data. Grinstein et al. (2018) state that the style of music could be defined as:

> "the timbres of musical instruments and musical genre"

and the content as:

> "some global musical structure (including, e.g., the score played and rhythm)" (Grinstein et al. 2018, p.587)

Within state-of-the-art research audio style transfer is most commonly performed by utilising images of spectrogram representations of audio (subsection 2.5.2) to allow for the use of image-based style transfer methods.

Dai et al. (2018) argues that the applying image-based style transfer methods to audio is an issue, because audio contains a large amount of features that cannot be simply separated into two categories. Instead they suggest that there are three different techniques of style transfer that can be applied to music — timbre style transfer, performance style transfer and composition style transfer.

Timbre Style Transfer

From Dai et al. (2018), timbre style transfer focuses on treating the timbre (subsection 2.1.4) as the style of the music and the performance control information as the content. By learning the timbre representation of one instrument, or music track, that timbre can be applied to another piece of audio's performance control. With this style of audio style transfer it would be possible to transform a song being played with a drum to one with the same expression but with a guitar if the timbre representation of the guitar was learned, and the drum's was removed. This form of style transfer is the most common one used within state-of-the-art music and genre style transfer systems (chapter 4).

Performance Style Transfer

Described as an unexplored field within audio style transfer, the performance style transfer method described by Dai et al. (2018) involves separating the performance control as the style and the implicit score of the audio as the content. An example of this form of style transfer would involve finding an artist's specific 'style of playing' and applying this to other songs. For example an artist may commonly alter the pitch of their guitar using a whammy bar despite this not being part of the score of the song. A performance style transfer system may pick up on this habit and apply it to other music tracks.

Composition Style Transfer

The melody contours of audio are treated as the style in composition style transfer while the content are the score features. This method of style transfer can be used for reharmonisation or improvisation as it would learn the melody contour of a music track but be capable of making significant alterations to rhythm, pitch and other score features of the audio. The difficulty in modelling the composition of audio is the main hurdle for this type of style transfer as little research has been put into formalising the structure of music accurately enough for features like chord progression to have a consistent grammar.

3.8. Image-to-Image Translation

Comparably to neural style transfer, image-to-image translation is another *style transfer-like* technique of image modification that has be used via convolutional neural networks and generative adversarial networks. An image-to-image translation describes the process of learning a mapping $G : X \rightarrow Y$ from the source domain X to the target domain Y such that the images of $G(X)$ are indistinguishable from Y (Zhu et al. 2017).

3.8.1. Conditional Adversarial Networks

Isola et al. (2016) implemented a conditional GAN (cGAN) capable of performing image-to-image translation tasks. By using paired sets of images it was capable of learning the underlying mapping function between these domains and apply them to any input. Figure 3.4 shows examples of outputs from the model.

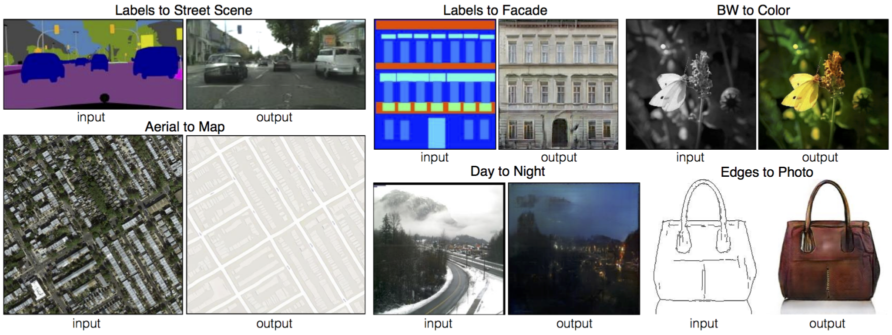


Figure 3.4.: Image-to-Image translation examples from Pix2Pix, from Isola et al. (2016)

The authors make a number of unique changes to cGAN architecture, such as implementing a generator similar to U-Net (Ronneberger et al. 2015) and a discriminator they name 'PatchGAN' respectively.

Typically for image-to-image translation an encoder-decoder network is used to downsample source images to a bottleneck layer upon which they are then upscaled to the target domain. Isola et al. (2016) claim this can cause low-level information to be lost in the downsampling process, such as edges or colours. As such their non-standard generator architecture allows the network to share low-level information between layers by 'skipping' across to other layers without downsampling information in the image.

Their discriminator, PatchGAN, uses an architecture that "penalizes the structure at the scale of patches" (Isola et al. 2016, p.2), meaning the network will classify $N \times N$ sections of an image (patches) as real or fake, rather than using the entirety of the image. Similar to a ConvNet, this defined patch size convolves across the image returning a confidence value, the discriminator's response is the average of all the responses from each patch of a singular image. Compared to other discriminators, PatchGAN was capable of modeling high-frequency structures, allowing for more crisp images with the capability of being applied to images of arbitrary size.

A variety of experiments were performed by Isola et al. (2016). Most involving using a variety of different datasets to test the applicability of the model. Some standouts from their experimentation were: translation from greyscale pictures to colour, daytime images to night, vector graphic maps to real life aerial map pictures and semantic labels of cityscapes to real life pictures of cityscapes.

For their evaluation they determined that 'plausibility to a human observer' was the goal for most of the tasks performed by their network. Because of this they test their map generation and image colourization results with human participants in a 'real vs. fake' test and compare it to similar methods of image-to-image translation, namely the CNN network from the 'Colorful Image Colorization' project by Zhang et al. (2016) and their own encoder-decoder network using L1 loss (L1). Furthermore they use an image recognition system to determine the semantic interpretability of their cityscape generation. Using their map generation network they found that 18.9% of participants believed the images generated from the network were real when transferring from vector graphic map to real life representation. The inverse of this translation (real life to map) only managed to fool 6.1% of participants. Regardless their network performed much better than the standard L1 model which only had 2.8% and 0.8% respectively. For colour generation 22.5% of participants thought the generated images were genuine but worse looking than images from the CNN created by Zhang et al. (2016), although Isola et al. (2016) note that the 'Colorful Image Colorization' model is specifically made for generating colours while their model is more general.

While the results of the proposed model by Isola et al. (2016) are commendable, the greatest weakness of the model is that it must be trained on pairs of images to find a mapping function. Sourcing such data can be difficult in domains where identifiable mappings cannot be determined by a human being, or if the available data for such mappings is sparse. Although Isola et al. (2016) state that even small datasets can lead to acceptable results.

3.8.2. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks

Addressing the main flaw of paired sets of data being a hard requirement for all training in the model presented by Isola et al. (2016). Zhu et al. (2017) created the CycleGAN model, which learns the mapping function required for image-to-image translation without the need for paired source to target examples. With this model the authors aimed to tackle the issue of many datasets of paired domains either being non-existent or difficult to create.

Differing from the previous model, CycleGAN learns the mapping function on the set level rather than for an individual element, such that:

> "$G : X \rightarrow Y$ where the output $\hat{y} = G(x), x \in X$ is indistinguishable from $y \in Y$ by an adversary trained to classify $\hat{y}$ apart from y"
> (Zhu et al. 2017, p.2)

With this mapping function $\hat{y}$ is capable of matching the empirical distribution of the target dataset. However a number of issues are raised with this mapping, specifically that it does not guarantee that an individual element of set X or Y is meaningfully mapped. Additionally, mode collapse was a common occurrence when training models using this type of mapping function. To circumvent these issues an imperative component was added to the structure of the CycleGAN model in the form of a cycle consistency loss measure, inspired by Zhou et al. (2016), that enforces the constraint — $F(G(x)) \approx x$ and $G(F(y)) \approx y$, formalised as the following loss equation (Zhu et al. 2017):

$$\begin{aligned} \mathcal{L}_{\text{cyc}}(G, F) = & \ \mathbb{E}_{x \sim p_{\text{data}}(x)} [\|F(G(x)) - x\|_1] \\ & + \mathbb{E}_{y \sim p_{\text{data}}(y)} [\|G(F(y)) - y\|_1] \end{aligned} \tag{3.4}$$

An identical evaluation as in subsection 3.8.1 was performed by the authors and compared against that model (which they refer to as 'pix2pix') trained on paired image datasets and four other models capable of image-to-image translation. They found that the CycleGAN model was capable of outperforming all of the models with the exception of pix2pix which had more accurate classification against image recognition systems, although its results were not presented for the test involving human participants.

Ultimately Zhu et al. (2017) considered the CycleGAN model to be effective at colour and texture translation but contained more failure cases for certain domains compared to pix2pix. They remark that the quality of training data could askew certain results heavily, as they describe a case in which they were capable of translating between images of horses and zebras but were incapable of accurately translating new images containing people riding horses. While they considered pix2pix to be much more effective at translating domains between image sets, the capability of using unpaired training data makes CycleGAN useful due to the simplified data sourcing and processing needed to make use of it compared to pix2pix.

In the following chapter we can see that image-to-image translation using CycleGAN has served as the basis for a number of audio style transfer related works.

3.8.3. StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation

Further expanding the field of image-to-image translation, the StarGAN model by Choi et al. (2018) concentrated on overcoming the hurdle of performing image-to-image translation between more than two domains, something that required multiple generators to be trained if the previously discussed models were used. As such they put forward a model capable of learning mappings between multiple domains of unpaired training data with only a single generator to learn the mapping function $G(x, c) \to y$ where G is the generator, x is an input image from the set of source images X, c is the target domain label of the target set of images Y and y is the output image representing an element of Y.

By adding a label specifying the domain to their training data Choi et al. (2018) were capable of using three loss functions:

- Adversarial - A standard for loss function used in generative adversarial networks section 3.4
- Domain classification - In which an auxiliary classifier is used, which optimises the discriminator to classify real images belonging to multiple domains and the generator is optimised to generate images that are accurately classified by the discriminator.
- Reconstruction - A cycle consistency loss in a similar vein to subsection 3.8.2 in which the generator creates an image from the source domain to the target domain and verifies that it can also take the inverse of this transformation.

To evaluate their model a number of experiments were performed in which StarGAN was compared to DIAT (Li et al. 2016), CycleGAN (Zhu et al. 2017) and IcGAN (Perarnau et al. 2016) on transferring between seven different domains involving various hair colours, genders and ages. Choi et al. (2018) created multiple models trained between two different domains for the previously mentioned models, while StarGAN was trained on all of the domains using one model. In a qualitative analysis they found that the quality of images generated by StarGAN was much higher quality than the aforementioned models. They attribute this to StarGAN's capability to regularise when being trained on many different domains, lowering its likelihood of overfitting between domains. A quantitative analysis was also performed where participants were asked to pick the best generated image based on realism, quality of transfer and similarity to the original celebrity's visage. A transfer between two different domains (e.g. $X \to Y$) and multi attribute transfers (e.g. $X \to Y \to Z$) was also performed. In all experiments StarGAN greatly outperformed the other models showing that it was effective in single and multiple transfers between domains.

Chapter 4: Literature Review

A number of systems related to generating audio and genre transfer exist within current research. Many systems have been created that allow for some kind of musical generation, whether this be symbolic or raw audio. All literature within this literature review will cover audio generation and domain transfer systems that are closely related to the system implemented in this thesis. Section 4.1 begins with describing systems involved in the generation of raw audio waveform without the use of intermediate representations, such as spectrograms. Following this section 4.2 explores symbolic music generation and how it relates to the creation of audio generation. Expanding on the concept of genre style transfer — section 4.3 discusses state-of-the-art research involved in using style transfer techniques (section 3.6) to perform genre modifications to existing music tracks. Discussion regarding spectrogram reconstruction is presented in section 4.4, which involves three deep learning models that were evaluated during experiments (chapter 7). Objective and subjective evaluation approaches for audio are then presented and discussed in section 4.5. Finally, an overview of the literature review is given in section 4.6 which describes how the research discussed in this chapter influenced decisions made for the implemented system.

4.1. Raw Waveform Generation

The term 'raw waveform' or 'raw audio' is often used to describe an audio signal displayed across time. This type of data is typically paired with metadata, such as the artist name and song title, and stored digitally as an audio file format such as WAV (*.wav*) or MPEG (*.mp3*) at a specific sampling rate — typically 44.1kHz. Directly using waveform data within the deep learning field is commonly done (as will be shown throughout this literature review) but it is less popular than other methods, such as using spectrogram or MIDI data.

4.1.1. WaveNet: A Generative Model for Raw Audio

Google DeepMind's WaveNet (van den Oord et al. 2016) is described as a deep convolutional neural network model for generating raw audio that was initially used to improve text-to-speech (TTS) systems. The defining factor of WaveNet that separates it from typical CNNs were its use of dilated causal convolution layers which are well suited for time-series data. Causal convolutions ensure that the output at a chosen point in time is only created using data from time-steps occurring before that time. Data that occurs at time-steps after the chosen point in time are not responsible for influencing outputs at a previous time-step. A chosen dilation rate is then used to exponentially increase at each layer to skip over inputs when connecting between layers, this prevents the network from using the entirety of the history available to reduce complexity. Figure 4.1 shows an example of dilated causal convolutional layers. Note that no dilation rate impacts the number of nodes to be skipped and that the output of each layer cannot be dependent on data from the previous layers that occur at a future time-step. While convincing speech synthesis was the primary goal of the paper, the model showed some capability at effectively synthesising music. Described as, being able to generate "any kind of audio,

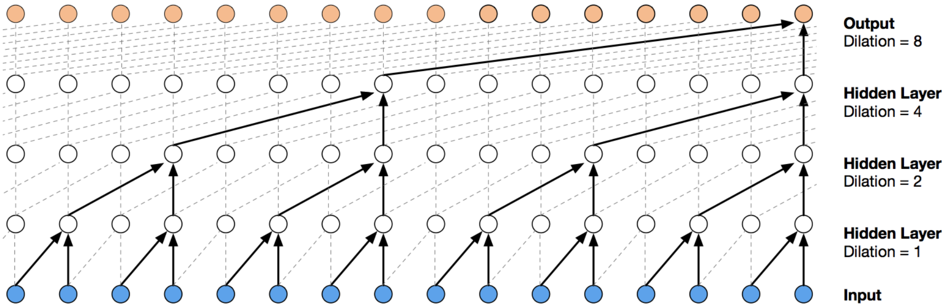


Figure 4.1.: Visualisation of dilated causal convolutional layers, from van den Oord et al. (2016) with permission

including music" by van den Oord et al. (2016), they support this statement by training the model on a YouTube piano dataset containing 60 hours of solo piano music, and the MagnaTagATune dataset (chapter 5) containing 200 hours of music of various genres. Overall van den Oord et al. (2016) described WaveNet as "a generic and flexible framework for tackling many applications that rely on audio generation". Although responsible for creating and training on time-series data, WaveNet was capable of generating more convincing audio than other popular text-to-speech synthesis systems that used models considered to be superior to causal convolutional networks, such as long short-term memory (LSTM) recurrent neural networks. Despite this, the researchers admitted that evaluating the results of the WaveNet model was difficult to do quantitatively but noted that the audio generated was "often harmonic and aesthetically pleasing" but was not capable of keeping a consistent volume, instrumentation or genre when generating audio unconditionally.

Some significant limitations within WaveNet were noted by the authors, namely that it could take a non-trivial amount of time to generate just one second of audio. However, an extended version of WaveNet utilising a parallel feed-forward network by van den Oord et al. (2017), aptly named Parallel WaveNet, parallelises the audio generation and claims to generate audio twenty times faster. As such WaveNet and its derivatives are a promising look into the generation of raw audio and strongly contribute to research questions R1 and R3. The effectiveness and impact of WaveNet within audio generation can also be seen by its use in many research papers aiming for tasks involving audio generation.

4.1.2. Adversarial Audio Synthesis

Donahue et al. (2018) created WaveGAN and SpecGAN — models capable of generating raw audio for use in sound effect generation. The most noticeable difference between the models being that WaveGAN is trained using raw audio as input, while SpecGAN uses short-time Fourier spectrograms Figure 2.5.2 as its input. Both networks were capable of generating up to 1 second of audio and could generate convincing drum, speech and piano audio. Both models could also generate audio faster than WaveNet. Their implementation was based on DCGAN (Radford et al. 2015), a GAN that uses convolutional layers to aid in the generation of images. To process raw audio as input the WaveGAN model flattens the filters used in DCGAN from 5x5 to 1x25 to increase the receptive field of the filter due to audio containing significantly more periodic patterns compared to natural images. SpecGAN uses a spectrogram representation of the audio input that is "well suited to GANs and can be approximately inverted". The paper stated that both networks were trained on the SC09 speech dataset and were compared and evaluated. Human judges were used to evaluate the quality of the audio via mean opinion score (subsection 4.5.2) by separately rating the sound quality, ease of intelligibility and speaker diversity on a scale of 1 to 5 (higher is better). Additionally, participants were asked to identify generated audio of digits between 1 to 9 and label it.

Overall, WaveGAN was rated higher than SpecGAN from their subjective audio quality evaluation in terms of speaker diversity and sound quality. Although SpecGAN was labelled more accurately than WaveGAN when it came to listeners identifying digits.

After their evaluation, the authors concluded that SpecGAN was likely better at capturing the variance in the underlying data compared to WaveGAN, while WaveGAN had superior sound quality due to the Griffin-Lim inversion process being lossy, which was used in SpecGAN.

SpecGAN and WaveGAN show usefulness in the unconditional generation of audio and put the focus away from speech generation by giving more attention to sound effect generation. Their short audio generation capacity and unconditional nature make it quite different from what the goal of this thesis. For the generation of music there should be some importance given to the length of audio that can be created from a system and unconditional generation could be unsuited if existing music tracks need to be transformed.

4.2. Symbolic Music Generation

Generation of symbolic music, such as sheet music, Music Instrument Digital Interface (MIDI) or piano roll, has also seen noteworthy research (Briot et al. 2017). Symbolic representations of audio allow for an unambiguous well-defined representation of music. While symbolic representations are not a focus of this thesis there are notable projects that utilise many similar methods that could also be taken into consideration for the creation of raw audio.

Dong et al. (2018) created MuseGAN, a network capable of generating original piano-rolls when trained on a the LPD dataset (Raffel 2016) which contains a large number of multitrack piano rolls. The authors proposed three different models based on how music can be performed; the first based on a group of musicians improvising without planning any harmonic structure (colloquially known as "jamming"), another based on a composer creating music via planning a composition beforehand, and a third hybrid model that combined aspects of both of the aforementioned models. The system proposed by Dong et al. (2018) made some unique contributions to music generation in that it was one of the first to create multi-track sequences of music.

Brunner et al. (2018) used CycleGAN to achieve genre transfer (also covered in section 4.3) using MIDI representations of music, which is very similar to the goal of this thesis. Data in MIDI format can be used in digital synthesisers or audio workstation software to be played as a stream of musical notes with a digital instrument.

4.3. Genre Style Transfer

State-of-the-art work that aims to perform domain transfer on audio is discussed in this section. Throughout this section genre transfer's meaning can be interpreted in various ways, and thus implementation details, results and evaluation can differ depending on the researcher's understanding. Primarily, genre transfer is always expressed as a type of neural style transfer task in which the domain is portrayed as the the researcher's understanding of genre. Throughout this section the terms 'genre transfer' and 'style transfer' may be used interchangeably, with the term style transfer being used to refer to systems that could be applicable to the transfer of genre between music but may not have musical genre transfer as a key element of their research. The viability and analysis of genre transfer contributes to research question **R2** in which selective remixing is achieved via the use of genre transfer.

4.3.1. Audio texture synthesis and style transfer

One of the earliest contributions to domain transfer using audio was the convolutional neural network created by Ulyanov & Lebedev (2016). The authors describe their process for texture synthesis and style transfer in the audio domain. Similar to many other style transfer tasks involving audio, they convert raw audio into STFT spectrogram and use this as their input data for a model, acting as an intermediate representation. The spectrogram output from the network was then inverted using the Griffin-Lim algorithm. A one-dimensional convolutional neural network was used to learn the style and content representation of the audio. Although they apply the style of audio to the content of other audio examples they do not describe the the process as style transfer, instead they state that style transfer is purely an image-based technique. Despite being one of the earliest works involving domain transfer of audio, Ulyanov & Lebedev (2016) expressed that the music and audio generation community were likely to improve on the concepts of style transfer and audio synthesis. This can be seen in the remainder of research within this section, which commonly use spectrograms as an intermediate representation for use in genre style transfer.

4.3.2. Neural Style Transfer for Audio Spectrograms

The research provided by Verma & III (2018) was the first relevant academic research detailing style transfer among audio. Similar to subsection 4.3.1 the authors used a STFT log-magnitude spectrogram with a CNN, in this case AlexNet (Krizhevsky et al. 2012) with a smaller receptive size. They state that larger receptive sizes lead to poor localisation in the audio reconstruction, which resulted in artefacts within the audio. Here, the authors referred to their style transfer as timbre transfer (section 3.7). Results from their experimentation involved transferring the sound of a harp to the style of a tuning fork and the sound content of singing being transferred to the style of a violin. Methods for evaluating the audio quality or effectiveness of their style transfer attempts were not mentioned despite their experimentation. Regardless, the work provided by the authors was meaningful but was largely conceptual compared to what was achieved by Ulyanov & Lebedev (2016) in terms of what could be possible with style transfer in the audio domain.

4.3.3. Applying Visual Domain Style Transfer and Texture Synthesis Techniques to Audio

Huzaifah & Wyse (2019) performed a study to determine the most effective methods and challenges involved when performing visual inspired domain style transfer on audio data. Multiple types of CNNs were evaluated by the authors while importance was put heavily on the viability and issues surrounding representing audio as a spectrogram for style transfer related tasks. Log-magnitude spectrograms were described as being a poor choice for CNNs because of dilation, shift, rotation and mirroring techniques that are utilised in the visual domain, but when applied to spectrograms can entirely remove the time domain information of the representation. This results in the resynthesised audio being dramatically altered in terms of time structure. Following this the authors mentioned that the "most pertinent" issue in using spectrograms in visual domain related tasks was their inherent asymmetry of axes. Altering a section of a spectrogram's frequency across the y-axis would change both its pitch and timbre, meaning the sound would have entirely changed from its original characteristics. To lessen this issue Huzaifah & Wyse (2019) recommend mel-frequency and constant-Q transform (CQT) spectrograms over log-magnitude spectrograms. Both mel and CQT spectrograms were shown to increase performance for classification tasks, suggesting that these scales are more effective. CQT spectrograms were noted by the authors to be well-suited due to preserving the harmonic structure and keeping the positions of harmonics the same even when the frequency has been altered.

A number of experiments performed by the authors showed various CNN architectures being used to perform style transfer using CQT and STFT spectrograms. Both types of spectrogram were then inverted back into audio using the Griffin-Lim algorithm. When comparing their hypothesis that CQT are more effective for translation invariance (superior to STFT when resynthesised back into audio) they find that it does perform better than STFT, although the timbre is still not completely preserved. They also discuss that when synthesising voice data STFT spectrograms generated a single voice, while inverted CQT spectrograms of the same data generated multiple voices at different pitches. They go on to say that this is possibly due to multiple harmonic groups being present within CQT, which is not the case for STFT transformations.

From their experiments, Huzaifah & Wyse (2019) concluded that defining 'style' and 'content' within audio is difficult to comprehend and not well defined. As such they stated that from a high level of abstraction, the style of music may be its genre while the content would be the lyrics. While a low level abstraction would represent style as the timbre of the instruments and the content would be the notes and rhythm. This low level abstraction is described in multiple research papers covered within this thesis (subsection 4.3.4, subsection 4.3.5, subsection 4.3.7) and is representative of timbre style transfer as defined in section 3.7 by Dai et al. (2018).

4.3.4. Symbolic Music Genre Transfer with CycleGAN

As mentioned in section 4.2 genre transfer was achieved by Brunner et al. (2018) who utilised the CycleGAN, a generative adversarial network designed for image-to-image translation and capable of style transfer. The authors arrived at the conclusion that their genre transfer can be detected by a classifier, and to the "untrained" human ear (although they offer no human evaluation results). Transferring from jazz to classical music was mentioned as being the "most noticable" out of all of the genre transfers they attempted involving jazz, classic and pop music. To evaluate their system the authors also created a binary classifier trained on jazz and classic genres of music to identify one of the genres. The use of a classifier to evaluate the genre of music used in their system shows one objective way of measuring the effectiveness of genre transfer.

4.3.5. TimbreTron

Continuing the use of CycleGAN as a key aspect for genre transfer, Huang et al. (2018) created TimbreTron which utilised a complex pipeline for achieving timbre style transfer — which they described as manipulating "the timbre of a sound sample from one instrument to match another instrument while preserving other musical content, such as pitch, rhythm, and loudness". This form of genre transfer was achieved using a combination of WaveNet and CycleGAN utilising CQT spectrograms as their representation of audio data. Unique from some of the other research reviewed in this thesis, the authors did not opt to use the Griffin-Lim algorithm for audio reconstruction, instead using a conditional WaveNet vocoder. They note that reconstruction quality will always be limited within audio style transfer tasks if Griffin-Lim is used for the synthesis of audio. Their use of a conditional WaveNet involved training the network with pairs of CQT representation and a waveform. Then, when presented with a new CQT spectogram it was capable of generating an approximated raw audio waveform that could potentially have higher quality than Griffin-Lim for audio reconstruction. Although for this to be the case the WaveNet model must be trained on audio and spectrograms that closely met the same timbre, pitch and rhythm as the input. This method of audio reconstruction was inspired by the TacoTron 2 model created by Shen et al. (2017) who were capable of using the model to create audio from mel-spectrograms using WaveNet for use in text to speech synthesis.

A high level overview of the TimbreTron pipeline can be seen in Figure 4.2. Three

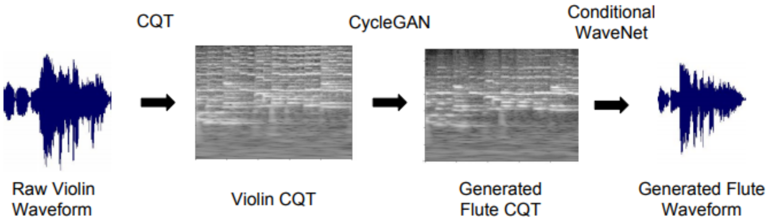


Figure 4.2.: High Level Overview of the TimbreTron Pipeline, used with permission from Huang et al. (2018)

steps are detailed by the pipeline, with the first being the transformation of the raw audio waveform into a CQT spectrogram. Similarly to Huzaifah & Wyse (2019), CQT spectrograms were chosen as being superior to other spectrogram representations due to being more effective at preserving harmonic structure, as well as having higher frequency resolution in comparison to other representations which allows for instruments such as the cello or trombone to be reconstructed at a higher quality. The second step of the pipeline used the CycleGAN model to perform genre transfer in a similar vein to systems described in subsection 4.3.4 and subsection 4.3.7. Finally, the WaveNet conditioned on CQT spectrogram data creates an approximated waveform of the newly created timbre shifted spectrogram.

Results from the research were promising, a number of human evaluations were performed to determine whether the transfer of timbre was audibly clear to users. A questionnaire was conducted to compare audio clips of an original piece of audio recorded with one instrument and its timbre transferred counterpart using a different instrument. Participants were asked to quantify the differences in sound of instrument, the similarity in note structure and audio quality. Overall 67.5% of participants noted that the music pieces were both almost identical in structure while 71.7% noticed that the instruments used for the timbre transferred creations of the audio still had similarities to the instrument used in the original version of the audio. Huang et al. (2018) also mention that most participants were capable of identifying the correct, or a very similar sounding, instrument used in the newly generated version of the audio. Additionally, the authors evaluated the audio quality of TimbreTron against two other pipelines: one using STFT with their WaveNet vocoder (now trained on STFT) and another using STFT with Griffin-Lim for audio reconstruction. Against both pipelines they found that participants were more receptive towards TimbreTron, with over 50% of them finding it to be superior to the STFT pipelines. Overall TimbreTron was considered to be capable of performing convincing timbre transfer and allowed for higher quality audio generation than what has been used before in previous audio style transfer tasks, thanks to its use of CQT spectrograms and audio reconstruction using WaveNet.

There are some differences in the work completed by this project and that proposed in this thesis. As stated most of the audio domains utilised by Huang et al. (2018) involved singular instruments, or instruments that are very typical for classical, older genres of music such as pianos, flutes and violin, which are indicitive of monophonic audio textures (subsection 2.1.5).

4.3.6. A Universal Music Translation Network

Mor et al. (2018) presented "a method for translating music across musical instruments, genres, and styles" with their Universal Music Translation Network which had some significant differences to most of the models described within this literature review. They used a single WaveNet encoder that was capable taking a variety of different inputs (e.g. multiple different instruments) and still produce style transfer from one piece of audio to another. They achieved this by training the encoder on multiple different instruments and eliminate any domain specific information from being encoded in the encoder by using a 'domain confusion network' that produces an adversarial signal to the encoder. As such the encoder was capable of receiving a variety of inputs, including instruments it had not been trained on, while still producing effective results. Raw audio was used as input to the network, making this stand out from other style transfer models that use visual based methods for style transfer. Surprisingly, the authors did not consider their work to be related to style transfer and state that two pieces of music cannot display similarities other than audio texture. Despite this the creators of the network do claim that translation between genres is capable with the network although their results indicate that classical styled music was the domain where most of their experiments were conducted.

Regardless, the network produces notable results in terms of timbre style transfer between instruments such as harpsichord and piano. To evaluate their network they made the network compete against three professional musicians who were tasked to translate music from three domains of music to piano. The domains chosen were harpsichord, orchestra and a variety of domains unseen in the network's training. The mean opinion scores of a crowd sourced group of volunteers was used to determine the quality of the music and how effectively it was translated from its original domain. Against all three professional musicians, the network placed last in both categories but was notably capable of equalling their results when it came to translating from unseen domains to piano. While the network eliminates domain specificity, the results of the network still show timbre style genre transfer, which has been achieved by other projects. Testing results on only three domains also puts into doubt how well such a network would perform on commercial music containing multiple instruments with much more complex beats and melodies compared to classical-styled piano music, which shows a trend in a number of research papers in this chapter.

4.3.7. A CycleGAN for style transfer between drum & bass subgenres

Differing from the previous research, Vande Veire et al. (2019) used CycleGAN for genre transfer between liquid and dancefloor music, which are both sub-genres of the drum & bass genre of music. This marks one of the few papers used on music fitting into a modern genre outside of the commonly used singular instruments use other in audio transfer work. The authors used mel-spectrogram representations for their music and performed image-to-image translation using CycleGAN. Rather than using Griffin-Lim to reconstruct the audio they used the phase information from the original STFT spectrogram before it was transformed into a mel-spectrogram, resulting in an audio recreation that did not suffer from the loss in audio quality that Griffin-Lim can cause when used on mel-frequency spectrograms.

From the results of Vande Veire et al. (2019), they signify that the unique elements of the liquid genre to dancefloor changed the sound of the snare drums to be louder and harder, and the frequency of the hi-hats to more suited those used in dancefloor drum & bass music. Conversely when going from dancefloor to liquid drum and bass they noticed that the hi-hats become lower in volume while the snare drums were less pronounced. The authors mentioned that the similarity in genres may have been what yielded the differences to be more pronounced and recognisable, although the results were not evaluated by any group other than the authors themselves, meaning some bias may be involved. Regardless the authors mention that their work could be the "first step to automated remix generation"(Vande Veire et al. 2019, p.2), which is likely an accurate claim as this is one of the few studies utilising modern music for use in genre transfer.

4.4. Spectrogram Reconstruction

Audio quality is a crucial element to generating convincing music and has been one of the defining factors in numerous evaluations conducted within the research analysed in section 4.1. Because spectrograms cannot be perfectly inverted to raw audio there is a focus within audio generative research on finding effective methods to generate high quality audio from spectrogram data. In context of this thesis, research questions **R1** and **R2** are most dependent on the generation of audio and its quality, meaning that if spectrograms are to be used within the system implemented then effective audio generation from spectrogram is a critical aspect that must be analysed.

While the Griffin-Lim algorithm (subsection 2.5.3) is a common way to reconstruct audio from spectrograms it is limited when reconstructing audio from CQT and mel-spectrograms since phase information is lost in these representations. This leads to an impasse where, despite mel and CQT spectrograms keeping harmonic structure more effectively than STFT (thus being more suited to speech and music respectively) their audio quality suffers when reconstructed with Griffin-Lim compared to STFT. To circumvent this there have been some advancements within deep learning to reconstruct audio without the use of Griffin-Lim, with three models in particular being some of the most recent.

4.4.1. Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions

Text-to-speech systems are where audio reconstruction from spectrograms have seen the most research. Shen et al. (2017) created Tacotron2 a text-to-speech synthesis system that utilised a modified version of the WaveNet (subsection 4.1.1) model conditioned on mel-spectrograms to generate waveforms. This modified version of the model was more lightweight, and required less "significant domain expertise" due to the complexity of the inputs required for the original version of WaveNet while still being capable of producing high quality audio. End to end text-to-speech synthesis was the aim of the Tacotron2 model so its architecture consisted of a predictive spectrogram generative model, multiple predictive LSTM networks and a five-layer CNN until any data reaches their WaveNet model. To relate to this thesis only the WaveNet model created by the authors will be focused on. The model exists as a modular part of their overall system architecture meaning it is not reliant upon other components within their system apart from being supplied spectrograms and raw audio. Training on the WaveNet model is done via pairs of mel-spectrogram frames and their equivalent audio waveform representations. In their paper Shen et al. (2017) state that a US English dataset containing 24.6 hours of speech from a female speaker was used to train the model.

From their evaluation the authors found that their system "significantly outperforms all other TTS systems and results in a MOS comparable to that of the ground truth audio"(Shen et al. 2017, p.3). Using a human rating service, the authors of Tacotron2 created 100 random samples of audio where each sample was given to up to 8 participants. Audio generated from five models (all trained on the same data) and the ground truth audio were used within the evaluation. These models include, a model previously created by the authors - Tacotron (Wang et al. 2017) which utilised Griffin-Lim and STFT spectrograms, WaveNet trained on linguistic data, a parametric model and concatenative model, which are both older style text-to-speech systems that were once used by Google. The participants rated audio on a 1-5 scale with 0.5 point increments to determine the quality of the audio in terms of how natural and human-like it appears. Ultimately the ground truth data performed the best, with a mean opinion score of 4.582 ± 0.053 while Tacotron 2 came second with 4.526 ± 0.066, which is impressive. Overall the usefulness of a WaveNet model conditioned on spectrograms shows a promising direction for audio synthesis from spectrogram representations, and provides support to all research questions in the context of visual style transfer systems such as those proposed by Huang et al. (2018).

4.4.2. WaveGlow: A Flow-based Generative Network for Speech Synthesis

WaveGlow (Prenger et al. 2018), a flow-based generative network also showed promising results on generating high quality speach from mel-spectrograms by combining elements from the flow-based generative model Glow (Kingma & Dhariwal 2018) and WaveNet. Simplicity of training and implementation while still producing high quality speech audio were some of the main contributions the authors claim.

From their own evaluation the network was compared to the Griffin-Lim algorithm, a WaveNet vocoder and ground truth data with the criteria for human participants being to rate the audio based on 'pleasantness' on a five-point scale. Overall the ground truth audio gained the highest score with 4.270±0.1340, with WaveGlow being second with 3.961±0.1343, and WaveNet and Griffin-Lim reaching 3.885±0.1238 and 3.823±0.1349 respectively. The authors note that the subjectivity of audio could have impacted the scores. Despite WaveGlow not significantly outperforming Griffin-Lim and WaveNet the authors highlight that the simplicity in training and speed of generation were its main advantages, being able to synthesise ten seconds of speech at 520kHz compared to WaveNet's 0.11kHz. Music generation is not mentioned by the authors, only speech, meaning that there may be some experimentation needed to test the network's viability with music generation.

4.4.3. MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis

Both of the models previously mentioned have their own drawbacks. The WaveNet vocoder is capable of generating convincing audio but takes a significant amount of time to synthesise one second of audio. WaveGlow in comparison is considerably faster at audio synthesis but is hampered by its intense training process with the authors stating that it was "trained on 8 Nvidia GV100 GPU's"(Prenger et al. 2018, p.3) putting into doubt the viability of the model in terms of reproducability and practicality.

With one of their contributions being lessened difficulties in training and synthesis time, Kumar et al. (2019) created MelGAN, a GAN capable of generating waveform from mel-spectrogram that simplifies training and synthesis time without significantly sacrificing audio quality. The authors achieved this by making alterations to typical GAN architecture, with some insight from other audio related GAN projects.

The architecture of MelGAN involves the generator being represented as a convolutional neural network taking the mel-spectrogram and raw audio waveform as input. Differing from other GANs, MelGAN does not take a global noise vector because the authors found that it does not make a noteworthy difference to their results. Within their generator they avoid the generation of checkerboard patterns that can be created in deconvolutional steps as described by Odena et al. (2017) by ensuring their kernel size and stride are thoughtfully chosen. In addition their use of weight normalisation (Salimans & Kingma 2016) was stated as an alternative to typical normalisation techniques, which made a significant difference in audio quality.

Three discriminators were used by MelGAN's creators to capture different frequency ranges of audio. This was done based on the fact that audio data typically has different structures at different levels. With three distinct discriminators working at different audio frequencies the authors were able to capture frequency ranges more accurately without having one discriminator being responsible for all frequencies - thus improving the training quality.

Since one of their contributions was to create a model capable of quick inference compared to other available solutions, Kumar et al. (2019) MelGAN's inference speed on CPU and GPU to WaveNet (subsection 4.1.1), WaveGlow (subsection 4.4.2) and Clarinet (Ping et al. 2018). In both cases MelGAN severly outclassed all of the models, being over ten times faster than WaveGlow and Clarinet when using GPU and twenty-five times faster on CPU. A quality evaluation using mean opinion scores was also performed against WaveGlow, the Griffin-Lim algorithm and WaveNet with all being given mel-spectrograms to invert into raw audio. MelGAN placed third in the evaluation (achieving a score of 3.61) much higher than Griffin-Lim's 1.57 score, and was a comparabled result to WaveNet and WaveGlow which achieved 4.05 and 4.11 respectively.

Use in music domain translation was also mentioned by the authors in which they showed that MelGAN was modular enough to replace existing mel-spectrogram to audio models used in other systems. They replaced the autoregressive model in subsection 4.3.6 and state that it achieves "decent quality", although no further evaluation on the music was given, likely due to the focus of the experiment being the modularity of MelGAN. Overall the Kumar et al. (2019) provided a model that seemed robust and lightweight in comparison to other state-of-the-art methods in spite of its poorer audio quality.

4.5. Evaluation Approaches

An evaluation of a system capable of generating remixed audio was the first research question proposed to meet the goal of this thesis, while one capable of generating quality audio that can be evaluated was another. Two evaluation methods based on these research questions were investigated for this thesis. The first, focusing on how to evaluate the genre of music, and the second focusing on how to determine the quality of the generated audio. Without an evaluation covering both genre classification and audio quality evaluation there would be an inability to determine whether the system created is robust enough to truly generate selective remix style music genres. Throughout the systems described in section 4.3 and section 4.4 the use of mean opinion score is frequently used to measure the subjective quality of the audio while genre classification methods have also been touched on by studies such as section 4.2.

4.5.1. Genre Classification

Classification of genre is important, in that to convincingly transfer music from one genre to another there needs to be some metric to represent how successful the transfer was. Due to this some studies have opted to use external systems capable of classifying music into genres as part of their evaluation. Brunner et al. (2018) created a binary classifier using a convolutional neural network that was capable of distinguishing between two genres of music via confidence values. From this model the authors were able to analyse how closely the music output from their GAN was similar to a specific genre. This approach is novel but relies on their classifier being well trained and evaluated for their analysis to be reliable.

4.5.2. Mean Opinion Scores

Commonly, within the previously mentioned studies mean opinion scores (MOS) have been the evaluation criteria for determining the quality of the audio created. The International Telecommunication Union (2016) proposed MOS as a method for evaluating the audio quality of telecommunication systems in which opinion scores are taken from a group of participants and the mean is calculated. Their definition of an opinion score in the context of telephone transmissions is as follows:

> "The value on a predefined scale that a subject assigns to his opinion of the performance of the telephone transmission system used either for conversation or for listening to spoken material."
> (International Telecommunication Union 2017, p.23)

Three categories of MOS model are defined - Subjective, Objective and Estimated - which are further divided into groups depending on the frequency of the audio. While initially created to evaluate speech quality over telecommunication systems they state that "general audio signals, such as music or mixed speech and music" can be used with subjective models of MOS but not with objective models. This is true for a number of the studies covered in this chapter which used a five-point MOS scale for subjective testing of perceived audio quality with test participants. The five-point scale recommended by the International Telecommunication Union (2016) is shown in Table 4.1. While similar to a Likert-scale in presentation, taking the mean of results from a MOS evaluation is considered to be mathematically valid.

Value	Opinion
1	Bad
2	Poor
3	Fair
4	Good
5	Excellent

Table 4.1.: Mean Opinion Score Scale

4.5.3. Objective Measurement Metrics for Audio Quality

In an effort to find objective metrics for audio quality a number of algorithmic models have been created that aim to replicate human hearing to rank the quality of audio. Objective qualities of music audio quality are discouraged by International Telecommunication Union (2016) in their presentation of MOS. Despite this these methods are detailed due to studies that have evaluated their use on music(Table 4.5.3).

Perceptual Evaluation of Audio Quality

Initially designed as an alternative to human listening tests, Perceptual Evaluation of Audio Quality (PEAQ) was created by Thiede et al. (2000) in an attempt to develop an objective measurement for the quality of audio. PEAQ calculates an *objective difference grade* (ODG) for audio signals that succinctly describe their quality in comparison to a ground truth reference audio signal. Table 4.2 shows the 'judgement of impairment' that ODG measures. If an audio signal receives a 0 ODG value then it can be considered indistinguishable from the reference signal, while a value of -4 means the audio signal is significantly worse in quality in comparison to the reference signal.

ODG	Judgement of impairment
0	Imperceptible
-1	Perceptible but annoying
-2	Slightly annoying
-3	Annoying
-4	Very annoying

Table 4.2.: Audio reconstruction comparison

Perceptual Objective Listening Quality Assessment

Perceptual Objective Listening Quality Assessment (POLQA) created by Beerends et al. (2013) for use in telecommunications speech audio quality was noted by the authors as an "ideal tool for all speech quality measurements from low end to HD voice communication" (Beerends et al. 2013, p.401) and note that it outperformed other popular objective speech quality systems. However no details about its suitability for music were mentioned.

Virtual Speech Quality Objective Listener

Another objective audio quality model, Virtual Speech Quality Objective Listener (ViSQOL) (Hines et al. 2015) was created for evaluating speech quality by modelling human speech quality perception. The model was further extended into ViSQOLAudio (Gillen et al. 2015) to allow for objective metrics to be created for music encoded at low bitrates. Overall the authors found it to be superior at measuring the quality of music compared to PEAQ and POLQA.

4.6. Overview

From a variety of the systems created by authors there is a large emphasis on image based domain transfer techniques, implemented via some kind of deep-learning architecture. Generative adversarial networks such as CycleGAN seem to have the capability to produce interesting genre transfer results (subsection 4.3.5 and subsection 4.3.7). However it should be noted that the music used in style transfer and audio generation tasks commonly have monophonic or homophonic textures and were commonly restricted to a single instrument playing. This puts into doubt the effectiveness of CycleGAN for genre transfer of more complex musical tracks, although it was shown to give reasonable results in genre transfer between two very similar genres in the case of subsection 4.3.7.

Additionally, spectrogram reconstruction has seen a majority of its research focused on the quality of speech rather than music. While both are audio domains, a single person speaking is entirely a monophonic audio domain meaning it is much simpler in comparison to most genres of music. It is common for modern genres of music to use a variety of music textures from polyphonic to homophonic. The use of classical music, specifically that focused on piano, seems to be the most common type of music utilised in these studies. This highlights that there is a missing link within audio generation research due to the lack of studies that focus on more complex types of music that use a variety of audio textures (subsection 2.1.5).

Chapter 5: Datasets

Preceding implementation or experimentation of the system, an analysis of available datasets was performed. This chapter presents a brief introduction into the contents of each dataset and their overall suitability for the system.

Each subsection presents the suitability of a dataset considered for use in the training of the system. To judge their suitability the ideal criteria each dataset should meet was provided. In Table 5.1 all datasets are displayed with their details concerning each criterion.

Reliability, representation of genre, size (number of individual unique samples), audio quality and audio texture are the criteria each dataset was judged against to determine its suitability.

The reliability of each dataset was determined by the degree of mislabelled data within it. No dataset can be considered perfect, but an effort was made to ensure erroneously labelled data was kept minimal.

Genre representation was also considered a large quality factor for datasets as a variety of genres was necessary for achieving condition **C4**.

The size is determined by the number of unique audio samples within the dataset, in all cases a large sample size is considered to be more beneficial.

Audio quality relates to condition **C5**, and is discussed in section 2.3. The sample rate of audio is used as the main determining factor for its quality when assessing datasets in this chapter. Higher sample rates lead to more accurately recorded representations of analogue sound. Due to this only the sample rate of audio was considered for each dataset.

Finally, the texture of the audio (subsection 2.1.5) is taken into account to support condition **C4**. Various audio textures were considered to be present within all datasets containing multiple genres, other than in cases where audio texture information was provided by the dataset authors.

5.1. NSynth

The NSynth dataset (Engel et al. 2017) contains waveforms of acoustic instruments and vocals with a diverse spread of pitch and timbres. Motivations for the creation of the dataset are that the authors wished to create:

> “a benchmark and entry point into audio machine learning”
> (Engel et al. 2017, p.10)

The dataset consists of 305,979 notes (with a note being defined as a four second audio snippet) with a unique pitch. While the dataset is large and well documented throughout the research paper it does not label its music into any musical genre - instead separating data by the family of instrument. Additionally all notes within the dataset are monophonic and 16kHz meaning the audio texture and quality was unsuitable, especially in comparison to other datasets described within this section. NSynth was not considered a useful dataset mostly for its lack of genre labels and focus on monophonic instrumentation in all of its samples.

5.2. AudioSet

Google's AudioSet (Gemmeke et al. 2017) was created for use in audio event detection and contains 2,084,320 human-labelled 10-second audio clips from YouTube split into 632 classes with the largest class being music, containing 1,011,949 examples. Even with its high number of music tracks the amount with genre labels is smaller, at 137,153 over 25 genres with an uneven distribution among them. Additionally, Google does not offer audio waveform directly for download, instead providing a csv file containing YouTube links and timestamps to the audio. AudioSet was not be considered to be very reliable because no quality assessment was performed on any of the music genres within the dataset. All labels in the dataset are automatically generated via view count, metadata, tags and other factors. The authors mention that the labelling can be considered imperfect due to this. No mention of audio quality or texture is given by the authors but due to the size of the dataset it is considered to be varying for this analysis. AudioSet met some criteria that allowed it to be suitable for the system but the issues involved in sourcing all of the data from YouTube was unappealing compared to other datasets that were discovered.

5.3. MagnaTagATune

MagnaTagATune (Law et al. 2009) provides the audio of 25,863 songs, sourced from the free music website Magnatune[1]. The dataset is hosted by the University of London Machine Intelligence and Music Informatics Group and was labelled via a digital tagging game using human users. All labels contained within the dataset are: song title, artist and album, but not genre. Unlike AudioSet the dataset is available for download in its entirety as an archive containing mp3 format files. Benzi et al. (2016) list MagnaTagATune as a dataset containing low quality audio due to its 16kHz sample rate and describe its audio selection as limited. It was not considered unreasonable to naïvely label by genre using the artist as the basis for the choice. However, as a whole, the dataset was considered unsuitable due to its low audio quality in comparison to the other datasets available.

5.4. Free Music Archive

While not intended for machine learning tasks, the Free Music Archive (FMA) (Tribe of Noise 2020) is a website that hosts user-submitted music on royalty-free licences available for free download. The majority of the music on the website was released under various Creative Commons licences, with most licences allowing the music to be used freely provided it is not used commercially. Additionally, the maintainers of the FMA categorise

[1] http://magnatune.com/

music into genres meaning that music was already labelled for use in a dataset, giving greater potential to the site's repository of music compared to the previously mentioned datasets.

Further investigation into FMA found existing datasets created for machine learning tasks that use music sourced from the FMA website. Benzi et al. (2016) created the Free Music Archive: A Dataset For Music Analysis (*FMA Dataset*) with the aim of contributing an easily accessible dataset for use in music information retrieval. All audio in the dataset is dumped from the website, and uses the site-provided genre labels and track metadata to create a complete dataset. The creators offer numerous datasets of varying sizes, the smallest containing 8,000 tracks evenly distributed across 8 genres, and the largest having up to 106,574 full length tracks over 161 genres.

Due to the labelled genres of music being taken from the FMA site there is reasonable concern that some data could be mislabelled. All music submitted to the FMA is submitted by the creator, and is verified by a human curator to ensure it meets the standards provided by the site. Benzi et al. (2016) provide an insight into the reliability of the labelling within the dataset:

> "While the artists are the best placed to judge the positioning of their creations, they might be inconsistent and motivated by factors not necessarily objective, such as achieving a higher play count." (Benzi et al. 2016, p.4)

In spite of this the dataset was considered a strong contender compared to the others listed in this section due to being the first found with significant size and genre labels.

5.5. International Society for Music Information Retrieval

The International Society for Music Information Retrieval (ISMIR) released the ISMIR04 Genre dataset (Cano et al. 2018) which consists of 2187 tracks originating from Magnatune. Originally made for use in genre classification, the dataset contains mp3 format files of six different genres of music, unevenly distributed, with classical music making up over 30 percent of the dataset, meaning the genre representation is quite poor. A 44.1kHz sample rate is provided for all audio in the dataset. Unfortunately its size is very small in comparison to all other datasets mentioned within this section. This in addition to the overabundant amount of classical music (a genre that is one considered to not be outwith the standard) the dataset was considered very unsuitable.

5.6. Bandcamp

While it does not host any datasets, Bandcamp is another website that hosts music similar to the FMA and Magnatune. Bandcamp's genre list is large and allows for easy filtering of subgenres. Artists can host their music for free or as a paid service. Some artists active on the site were contacted in the search for creating a new dataset. Due to condition **C2** there was a preference towards datasets that use non-standard genres so an effort was made to source music from Bandcamp that fit into this condition. The vaporwave genre music artist ‘Death’s Dynamic Shroud’ was contacted and gave permission for their discography to be used in this thesis. Their entire discography consists of 289 unique tracks which was small in comparison to the other datasets. All music listed on Bandcamp has a 44.1kHz sample rate and can be downloaded in a variety of formats. The site itself also allows for downloads in bulk if you have created an account and are following an artist. Alone the music from one genre is not suitable but the aim of sourcing this music was to explore the possibility of adding more genres to one of the existing datasets.

5.7. Self Made Dataset

In a similar vein to the music sourced from Bandcamp, the viability of creating a new dataset was also considered using FMA as a source due to the large amount of music released on copyright free licences available for download. While free download was supported on the site there was no efficient interface for downloading large libraries of music in bulk. A webscraper was implemented to download large amounts of files automatically although, music files are large in size and downloading large quantities of music was a time consuming task. Regardless, an effort was made to create a simple dataset using music scraped from the FMA website. However there were some caveats to this approach. The FMA site has undergone a number of potential shutdowns and acquisitions by various companies. At the time of writing the owners have revoked public access to a variety of pages and turned off all search functionality, thus limiting the amount of music that can be accessed by the average user. Overall the existing FMA dataset was preferred over creating a new one using the same source due to the difficulties in creating an effective webscraper.

Dataset Name	No. Tracks	No. Genres	Total audio length (hours)	Sample rate (Hz)	Varying Audio Textures
ISMIR2004	2,187	6	151.0	44,100	Yes
AudioSet	1,011,305	25	2,801.8	44,100	Yes
MagnaTagATune	25,000	N/A	215.5	16,000	Yes
NSynth	305,979	N/A	339.9	16,000	No
FMA: Dataset	8,000	8	~66.5	44,100	Yes
Bandcamp	289	1	16.7	44,100	Yes
Self Made	1,500	3	~35.2	44,100	Yes

Table 5.1.: Dataset comparison

5.8. Chosen Dataset

From the comparisons made and the existing evaluation of suitability the dataset that was chosen to be used throughout the majority of the project was the FMA Dataset because of its even distribution of genres and high audio quality. The size of the dataset is lacking in comparison to the others such as AudioSet and MagnaTagATune but due to concerns with storage space it was decided that the FMA Dataset would be the easiest to begin with, and can be expanded to larger sizes if necessary. To add more variety to the genres available the Bandcamp data was also added to the dataset, which causes one genre - Vaporwave, to be poorly represented. Regardless this was done to increase the amount of represented genres as well as explore how the genre of music is handled within genre transfer. Thus, all genres represented in the dataset were:

- Electronic
- Hip-Hop
- Vaporwave
- Instrumental
- Pop

Chapter 6: Architecture and Tools

This chapter discusses the basic architecture that was designed to implement the system and help meet the project goal (section 1.2) and the tools used throughout the development and experimentation. A pipeline architecture for the system was adopted in a similar vein to Huang et al. (2018), in which each part is an independent modular process. Each section describes a process of the pipeline, its purpose, how it interacts with processes that supersede it and the conditions it contributes to. Implementation details are omitted from this chapter and are instead mentioned in chapter 7, although possible software solutions are discussed here.

A basic data-flow diagram of the pipeline is shown in Figure 6.1. Benefits of the architecture included, the capability to separately implement, test and experiment with different parts of the system without impacting any of the other parts, due to each being intentionally modular. Finally, in section 6.4 an overview of all of the software tools used throughout the development of the system are described, along with reasoning as to why they were chosen.

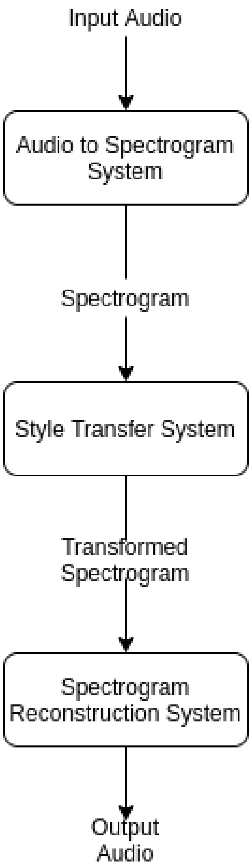


Figure 6.1.: Pipeline Data-flow Diagram

6.1. Audio to Spectrogram System

The first process within the pipeline includes the audio to spectrogram system which is responsible for transforming an arbitrary audio input to a spectrogram representation output as a png format image. This process within the pipeline contributes to condition **C3** as it supplies a spectrogram to the Style Transfer System which performs selective remixing. The decision to use spectrograms as an intermediate representation comes from the previously discussed literature in chapter 4, with the main reasons for their usage being:

1. Most studies attempting some kind of domain transfer between audio opt to use spectrogram data as an intermediate representation.
2. There are a number of programming libraries that allow for the creation of spectral information from audio.
3. Style transfer techniques are more commonly used within the image domain, meaning spectrograms can be applicable to more typical style transfer techniques compared to MIDI, piano roll or raw audio itself.
4. Using spectrograms as an intermediate transformation still ensures that raw audio can be used as both input and output to the system.

Huang et al. (2018) utilise CQT spectrograms as an intermediate representation within their pipeline and note their effectiveness over STFT and mel spectrograms. Despite this both mel and STFT were shown to be practical in creating suitable results in Vande Veire et al. (2019) and Gatys et al. (2015) respectively.

The choice of spectrogram representation influenced all succeeding processes of the pipeline, specifically the spectrogram reconstruction system, so the reconstruction methods available were also taken into account when implementing the first process of the pipeline. Two libraries (subsection 6.4.1 and subsection 6.4.1) that can be used to create spectrograms of various types are experimented with in the next chapter, with the choice of spectrogram chosen being CQT as a result.

6.2. Style Transfer System

The style transfer system also contributes to condition **C3** using the spectrogram image output from the audio to spectrogram system. Image-based style transfer techniques have shown to be an effective method of achieving genre transfer using spectrograms which is why the second process of the pipeline was decided to be this type of system.

A majority of the research from section 4.3 utilises CycleGAN (Zhu et al. 2017) conditioned on spectrograms, while other image style transfer techniques involved using some type of CNN.

The output of the style transfer system process is intended to be a transformed rendition of the spectrogram from the output of the audio to spectrogram system. By using an image-based style transfer technique selective remixing can be achieved by applying elements from one image domain to another. For example, a spectrogram image from the hip-hop genre of music is input into the style transfer system which then applies elements from another genre of music, such as electronic, to produce an altered version of the original input spectrogram.

Two machine learning image-based style transfer models were experimented with in section 7.2, with a modified version of CycleGAN being chosen for the final system.

6.3. Spectrogram Reconstruction System

Solving conditions **C2** and **C5** was the aim of the spectrogram reconstruction system since audio output is necessary and its quality is dependent on the effectiveness of the reconstruction method used. The importance of the reconstruction method chosen for this process is based on section 3.1 which shows that reconstructing audio from most types of spectrogram is a non-trivial task.

From experiments performed in the subsequent chapter, and based on results obtained from Huang et al. (2018), CQT spectrograms were chosen as the output of the audio to spectrogram process of the pipeline. For that reason a reliable reconstruction method was investigated in the experimentation phase, with possible solutions including: the Griffin-Lim algorithm (Griffin & Jae Lim 1984), WaveNet (van den Oord et al. 2016), MelGAN (Kumar et al. 2019) and WaveGlow (Prenger et al. 2018). A modified version of MelGAN is used in the final system based on these experiments.

6.4. Tools

All software used throughout development of the system is given in this section along with their role in development and why they were chosen. The entirety of the system was implemented and trained on the NTNU IDUN high performance computing cluster (Själander et al. 2019) which runs nodes on the CentOS Linux distribution with two Intel Xeon cores per node and up to 128GB of memory. Either Nvidia Tesla P100 or V100 GPUs ranging from 16 to 32GB of VRAM were present on each node and were utilised for training related tasks. All software used is presented in the following list with their version numbers for the sake of reproducibilty:

- Python **3.7.4**
  - PyTorch **1.4.0**
  - nnAudio **0.0.11**
  - Librosa **0.7.2**
  - Chromaprint and Acoustid for Python **1.2.0**
  - OpenCV **4.2.0.34**
- FFmpeg **4.0.1**
- GNU Bash **4.2.46**
- GstPEAQ **0.6.1**

6.4.1. Python

The use of Python for the majority of all programming in this thesis is due to my own familiarity with the language which allowed for smoother development. The availability of machine learning and audio analysis libraries and open source implementations of previously mentioned models (chapter 4) made Python a well-suited choice as the language for implementation. A discussion of all Python related libraries is presented below.

PyTorch

PyTorch (Paszke et al. 2019) was the main library used for the implementation of all deep learning models in this thesis. Similarly to Python, PyTorch was chosen due the familiarity with the library, allowing for easier development and testing. Due to multiple independent processes being the goal of the architectural design of the system (chapter 6) a focus was put on having all code developed under the same libraries. Majority of development was done using existing open source implementations as a base. Therefore, other popular Python machine learning libraries were avoided when considering open source code to keep the integration of systems as simple as possible.

Librosa

For audio processing tasks Librosa (McFee et al. 2020) was the most commonly mentioned library from the literature review in chapter 4 - with its primary use in the studies being to generate spectrograms from raw audio. Librosa is not used as extensively for development in this project but is used within initial experiments to investigate spectrogram reconstruction.

nnAudio

Another audio processing library nnAudio (Cheuk et al. 2019) is used for all spectrogram creation from audio files in place of Librosa. A number of notable benefits over Librosa were found with this library:

1. It can be integrated into PyTorch models easily.
2. It offers a more optimised version of the CQT algorithm for spectrogram generation and can generate more accurate CQT spectrograms as a result.
3. It can generate spectrograms quickly on the fly without the need of saving them to disk first.
4. It runs “at least 100 times faster”(Cheuk et al. 2019, p.11) than Librosa in a number of cases when used via GPU.

Chromaprint and Acoustid for Python

For initial experiments a Python package (Sampson 2020) was used that bundles the Acoustid Chromaprint C library (Lalinský 2020) into a Python wrapper for use in simple audio fingerprinting. With audio fingerprinting unique signatures can be generated that outline the ‘identity’ of a piece of audio. Fingerprinting is has seen use in information retrieval tasks (Cano & Batlle 2005) to compare two audio files using distance measures. Similarly, audio fingerprinting is used in some experimentation to compare the similarity of two pieces of music.

OpenCV

The OpenCV library was utilised to implement a FLANN feature matcher (Muja & Lowe 2009) for use in spectrogram comparisons in chapter 7. This allowed for a basic evaluation to be performed when comparing hyperparameter configurations. Due to only being used to implement an evaluation tool the OpenCV library is not considered to be a vital component for recreating the software pipeline of the system.

6.4.2. FFmpeg

FFmpeg (FFmpeg Team 2020) is an open source video and audio converter available on a number of platforms. The software allows for audio editing at a command-line level, which was the main reason for it being chosen due to the majority of development being done on a HPC server over a command-line interface. All audio preprocessing is done using FFmpeg, such as altering sample rates and splitting audio into chunks in the experimentation stage and concatenating audio during the evaluation stage.

6.4.3. GNU Bash

As all development was performed on a Linux operating system GNU Bash (Free Software Foundation 2020) was used to efficiently automate various command line tasks related to pre-processing data. Multiple Shell scripts written in Bash are used to run the completed pipeline and pre-process audio data in tandem with FFmpeg.

6.4.4. GstPEAQ

A plugin for the multimedia framework C library - GStreamer (Taymans et al. 2016) - is utilised for performing PEAQ (subsection 4.5.3) analysis to measure audio quality. GstPEAQ (Holters & Zölzer 2015) is used for all investigation regarding the quality of generated audio throughout the experiments performing in chapter 7.

Chapter 7: Experiments and Results

Using the basic architecture defined in chapter 6, a majority of the experiments performed were done to investigate the options available for each process in the pipeline and select those that were most effective. Later experiments were then performed on a completed version of the pipeline using the best methods from the previous experiments.

In section 7.1 the audio to spectrogram and spectrogram reconstruction processes within the pipeline (the first and third processes in Figure 6.1) are the focus, with all experiments aiming to find the best type of spectrogram representation and reconstruction method. In section 7.2 experiments aim to find an effective style transfer technique for use in the second process of the pipeline. Using the best methods found in section 7.1 and section 7.2 all ensuing sections relate to experiments using the models decided for use in the completed pipeline with the aim of investigating how effectively the completed pipeline meets project conditions.

7.1. Spectrogram Representation and Reconstruction Experiments

To find the most effective spectrogram representation and reconstruction method to supplement conditions **C3** and **C5** two of the earlier mentioned spectrogram types were experimented with - CQT and mel types. STFT is not used because the literature review indicated that it was an inferior representation for usage in style transfer techniques (subsection 4.3.5) compared to CQT and mel. With that being the case, all methods of audio reconstruction could only be approximations due to both spectrogram types being non-invertible transforms.

Before discussing experiments the method used to generate the CQT and mel spectrograms is discussed in subsection 7.1.1 to clarify how each spectrogram representation was created. Following this, subsection 7.1.2 discusses two separate construction methods performed on both types of spectrogram. The first being reconstruction of audio via the Griffin-Lim algorithm, and the second being reconstruction via two machine learning models based on MelGAN (subsection 4.4.3) and WaveNet (subsection 4.1.1). WaveGlow is also mentioned but is not directly compared to MelGAN and WaveNet, instead difficulties involved in its setup are discussed.

7.1.1. Experiment Setup

For the creation of spectrograms the nnAudio library (Cheuk et al. 2019) was utilised to generate CQT and mel spectrograms from *.wav* format audio files. To fully test spectrogram recreation the spectrograms are saved as greyscale PNG images to mimic as if they were being fed into the next process in the pipeline. Saving all spectrograms as PNG images presented a unique issue because all spectrograms are represented as real-valued matrices images, while greyscale PNG images are saved as normalised matrices between 0-255. Therefore, in addition to transforming the spectrograms to PNG images their minimum and maximum values were also recorded. These values were used to invert the normalisation process before being reconstructed, using the equation: $image \cdot (max -$

$min) + min$.

The code snippet shows the parameters used in the nnAudio library to create the CQT and mel spectrograms.

```
from nnAudio import Spectrogram
from torchvision.utils import save_image

cqt_layer = Spectrogram.CQT1992v2(sr=16000, n_bins=84, hop_length=256,
 ↪  pad_mode='constant', device='cuda:0', verbose=False, trainable=False,
 ↪  output_format='Magnitude')
mel_layer = Spectrogram.MelSpectrogram(sr=16000, n_mels=80, n_fft=1024,
 ↪  hop_length=256, device='cuda:0', window='hann', pad_mode='constant'
 ↪  fmin=0.0, fmax=None)

cqt_spectrogram = cqt_layer(audio_file) # Generates CQT spectrogram
mel_spectrogram = mel_layer(audio_file) # Generates Mel spectrogram

save_image(cqt_spectrogram) # Saves CQT spectrogram as image
save_image(mel_spectrogram) # Save Mel spectrogram as image
```

Ten 4 second samples of audio from the electronic genre of music are turned into CQT and mel spectrograms and fed into the three spectrogram reconstruction methods, which results in a raw audio waveform output. The Griffin-Lim algorithm (subsection 2.5.3), WaveNet Vocoder (subsection 2.5.3) and MelGAN (subsection 4.4.3) are the three reconstruction methods used.

For the Griffin-Lim algorithm implementation the `griffinlim` and `griffinlimcqt` functions from the Librosa library were used. Open source implementations of a WaveNet vocoder and MelGAN were used as the implementations for experimentation. WaveNet and MelGAN are both solely designed to be conditioned on mel spectrogram, for cases involving CQT the input data was simply swapped and the model was modified where necessary to work with CQT spectrograms, in the case of WaveNet this involves no changes to the code, while in MelGAN the `-n_mel_channels` parameter must be changed to 84. MelGAN is trained for 48 hours while WaveNet is trained for 72 hours based on training details from Kumar et al. (2019) and van den Oord et al. (2016). An attempt to use WaveGlow was also made, but the open source implementation was difficult to setup. This keeps the possibility open that WaveGlow could potentially be more viable than any of the methods evaluated.

7.1.2. Experiment Results

Given condition **C5** the audio generated should be both selectively remixed and of a high quality. Therefore, audio fingerprinting (Foote 1997) is utilised to compare each reconstructed audio track to its original counterpart. The audio fingerprint for each track is calculated and the Levenshtein distance measure (Levenshtein 1966) is used to calculate the similarity for each track. The average distance value is then calculated from the ten tracks. The higher value of the similarity the closer it is considered to be a similar and high quality recreation of the original and correspondingly high quality.

Results from the audio fingerprinting comparisons are are shown in Table 7.1. Overall CQT spectrograms lead to better results for all reconstruction methods, although this was only in the case of Griffin-Lim where the results had substantially improved. However, despite the results of the fingerprinting similarity measure implied Griffin-Lim produced the best results (audio being similar to its original form) a personal listening test was performed and it was decided that Griffin-Lim was not sufficient.

From the experiment it was found that Griffin-Lim is more effective than WaveNet and MelGAN at keeping a much higher range of frequencies within the audio but distorts all of the contents and adds a low frequency hissing sound to most reconstructions. This echoes a similar sentiment from Vande Veire et al. (2019) who mention that Griffin-Lim introduced:

> "a significant low-frequency 'buzzing' sound in the audio" (Vande Veire et al. 2019, p.2)

meaning that while Griffin-Lim may be effective at keeping an overall structure of audio despite sacrificing the quality of the sound.

For transparency it should be noted that audio fingerprinting is only an objective comparison between audio, and cannot account for subjective human hearing. Both WaveNet and MelGAN were capable of generating audio without distortions but were often missing higher or lower frequency sounds from their reconstructions. Ultimately it was decided that CQT spectrograms would be utilised as the spectrogram type used throughout the system while MelGAN would serve as the spectrogram reconstruction system in the pipeline. To differentiate between the MelGAN model conditioned on mel spectrogram and the model conditioned on CQT spectrogram used for the pipeline it is referred to as CQTGAN for throughout this thesis.

7.2. Style Transfer Technique Experiments

For the second process of the pipeline two possible style transfer techniques are experimented with. The purpose of the second process of the pipeline is to achieve selective remixing via style transfer, solving condition **C3**. Henceforth the aim of the experiments performed involving style transfer techniques are to find the system that most effectively performs style transfer on CQT spectrograms and produces intelligible results when reconstructed that differ from the original audio source.

CycleGAN (subsection 3.8.2) and StarGAN (subsection 3.8.3) are two machine learning models that are capable of style transfer. For all experimentation both of these models are compared. The first, CycleGAN, was chosen due to its usage within a lot of the

Reconstruction Method	**Spectrogram**	**Average Similarity (%)**
Griffin-Lim	Mel	44
	CQT	61.5
WaveNet Vocoder	Mel	42
	CQT	43
MelGAN	Mel	48
	CQT	51

Table 7.1.: Audio reconstruction comparison

research in the literature review, showing that it is proven to be capable of achieving promising results with spectrogram. The second technique, StarGAN, was chosen due to the authors (Choi et al. 2018) claiming it to be superior to CycleGAN in terms of quality of output while being able to be trained on multiple domains. In both cases open source versions of CycleGAN and StarGAN were used.

7.2.1. Experiment Setup

To setup the experiments, data for both of the models is first created. Using the nnAudio library greyscale images of CQT spectrograms are created from three of the genres from the dataset. 5400 images from the hip-hop and pop genre and 2150 images from the vaporwave genre are created for use with 300 and 240 images used as test data respectively.

Both models are trained on the same dataset for 200,000 iterations. In the case of CycleGAN two different models needed to be trained because it is only capable of bi-directional transfer, while StarGAN was capable of being trained on multiple domains using a single model. Additionally, one change is made to the CycleGAN model architecture when performing experiments due to a quirk that was observed during training that involved the model creating images containing a *checkerboard-like* artefact in its generated spectrograms (shown in Figure 7.1). This artefacting is mentioned by (Huang et al. 2018) as an issue with CycleGAN that leads to an impact in the quality of sound in the reconstructed spectrograms.

To remedy the effect a change is made to the deconvolution layer in the CycleGAN generator based on research by Odena et al. (2016). The following code change is made to the generator model:

```
    model += [nn.Upsample(scale_factor = 2, mode='nearest'),
                  nn.ReflectionPad2d(1),
                  nn.Conv2d(ngf * mult, int(ngf * mult / 2),
                      kernel_size=3, stride=1, padding=0),
                  norm_layer(int(ngf * mult / 2)), nn.ReLU(True)]
```

The same effect was not witnessed in StarGAN so the default open source implementation was used.

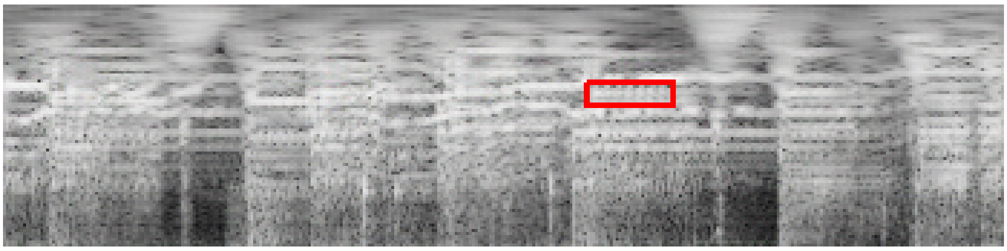


Figure 7.1.: Checkerboard Effect Caused By CycleGAN

7.2.2. Experiment Results

In Table 7.2 the results of CycleGAN and StarGAN are compared on the transfer between two genres from the dataset. Using CQTGAN from subsection 7.1.2 twenty style transferred spectrograms from the model are reconstructed back into audio and a PEAQ analysis (subsection 4.5.3) is performed on each to calculate the objective difference grade (ODG). The average value of the ODG is then calculated.

From all comparisons StarGAN and CycleGAN perform similarly, although CycleGAN produces higher quality reconstructions in the case of transferring pop spectrograms to vaporwave. Overall none of the models were capable of producing audio higher than *slightly annoying*. Additionally this experiment was considered fairly naïve because it involves relying on a spectrogram reconstruction method. It is possible that the results of the style transfer are hindered by CQTGAN. Regardless CycleGAN is chosen as the model to represent the style transfer process of the pipeline.

Style Transfer Model	Genre Transfer	Objective Difference Grade Average
CycleGAN	Vaporwave → Hip-Hop	-2.228
	Pop → Vaporwave	-3.913
StarGAN	Vaporwave → Hip-Hop	-3.612
	Pop → Vaporwave	-3.912

Table 7.2.: Style transfer model comparison

7.3. Completed Pipeline Experiments

From the results in section 7.1 and section 7.2 all processes of the pipeline were considered complete. In the final system the audio to spectrogram system is represented by the nnAudio library creating CQT spectrograms from an audio input. Its output spectrogram image is passed to the style transfer system which is represented by CycleGAN. Finally the output of CycleGAN is passed to the spectrogram reconstruction system which is represented by CQTGAN which restores spectrograms back into audio. The completed pipeline can be seen in Figure 7.2.

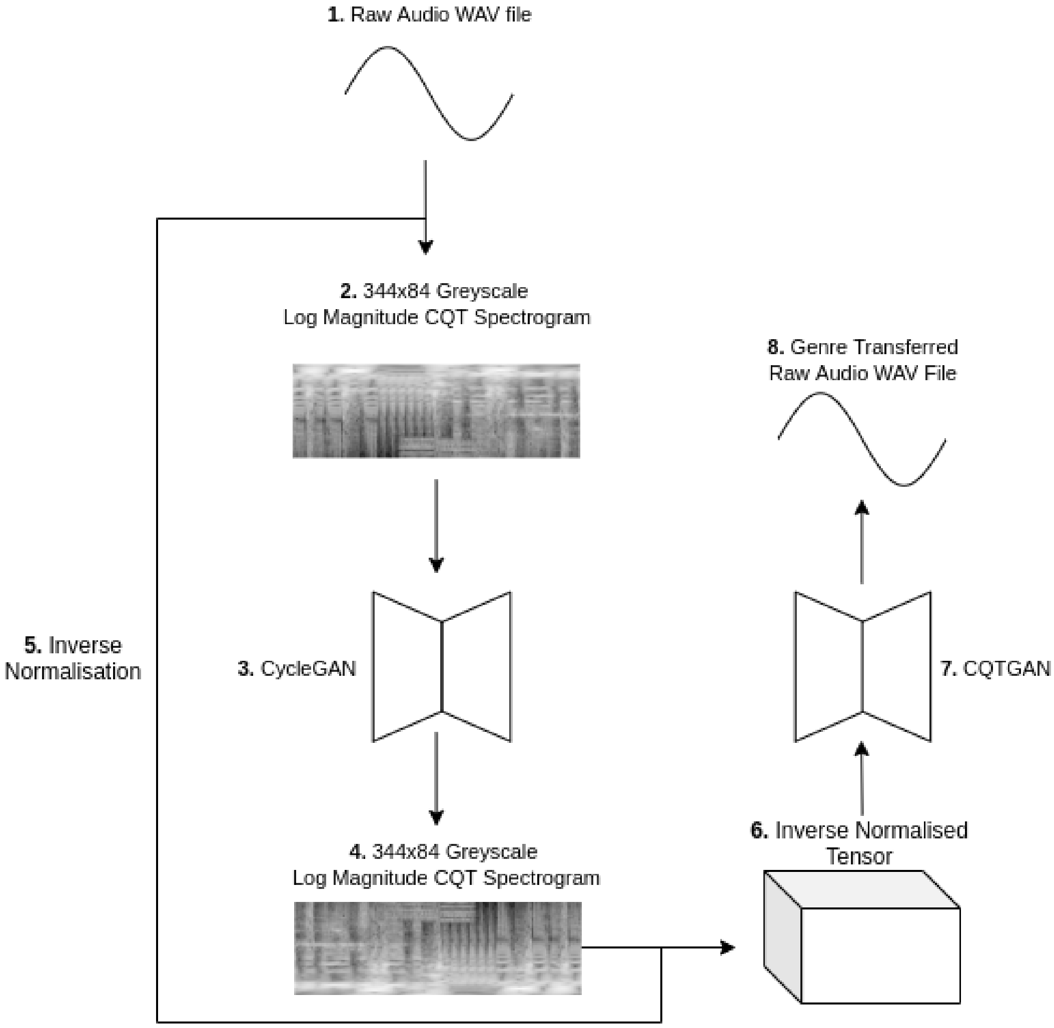

1. Raw Audio WAV file
2. 344x84 Greyscale Log Magnitude CQT Spectrogram
3. CycleGAN
4. 344x84 Greyscale Log Magnitude CQT Spectrogram
5. Inverse Normalisation
6. Inverse Normalised Tensor
7. CQTGAN
8. Genre Transferred Raw Audio WAV File

Figure 7.2.: Implemented System Architecture

Using this completed pipeline additional experiments were performed with the aim of investigating further into the CycleGAN and CQTGAN models used within the pipeline. In subsection 7.3.1 a Fast Approximate Nearest Neighbours Search (Muja & Lowe 2009) analysis was used to describe the degree of change that CycleGAN makes on its output spectrograms via two different hyperparameter configurations, to investigate how effectively it meets condition **C3**. Following this is an exploration into the sample rate of audio used for training and generation in CQTGAN, to show how effectively it meets condition **C5**. Finally a PEAQ audio analysis and audio fingerprinting similarity comparison is done using the entirety of the pipeline on unseen examples is performed to evaluate how well it meets conditions **C3**, **C4** and **C5**.

7.3.1. CycleGAN Hyperparameter Experiment Setup

To further explore how well CycleGAN performs style transfer over two spectrograms a Fast Approximate Neartest Neighbours Search (FLANN) feature matcher was implemented to compare the results of style transfer to the original spectrogram image that was input into CycleGAN. The FLANN feature matcher finds feature descriptors within an input image and tries to match them to the other image, showing which features within an image are identical. Due to selective remixing being the aim of CycleGAN in the pipeline it would be expected that its output spectrogram would contain differences its input spectrogram. To determine "how different" the transformed CycleGAN spectrogram is to its original spectrogram the FLANN matcher is used to determine the number of feature matches found in the same spectrogram. In Figure 7.3 a visual representation of the feature matching run on the same spectrogram is shown. 199 features are matched in this example, which is used to consider the images to be exceedingly similar. The feature matcher is run through all of the spectrograms for one genre and the average value is used as a baseline value for what determines a spectrogram to be similar.

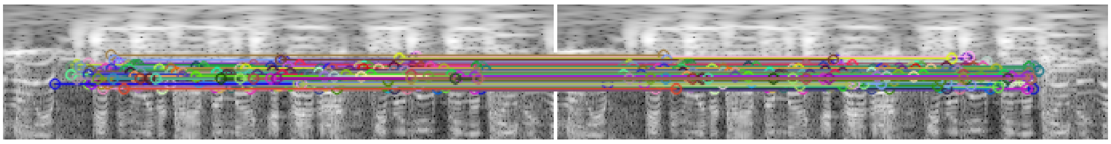


Figure 7.3.: FLANN Feature Matching Example

With this in mind two CycleGAN configurations with differing hyperparameters are investigated, with the aim of the FLANN feature matching describing which is capable of performing the most changes to the spectrogram, and thus the output audio at the end of the pipeline.

The first configuration used the default hyperparameters from the open source repository, while the second was inspired by results from Yang & Chung (2019) in which CycleGAN was used as an accent conversion tool for mel spectrograms. CycleGAN contains over thirty hyperparameters that can have an impact on the training and inference of the model. Due to this only hyperparameters that are changed from the default configuration are shown. A list of differences for both configurations is in Table 7.3.

For Configuration 1 of CycleGAN a batch size of 1 with 3 discriminator layers was used, while in Configuration 2 the batch size is altered to 4 as done by Yang & Chung (2019)

	Configuration 1	**Configuration 2**
Hyperparameters		
Batch size	1	4
Number of discriminator layers	3	5
Data Pre-processing		
Flipping	Yes	No
Crop size	344	344
Load size	286	344

Table 7.3.: CycleGAN Configurations

and the number of discriminator layers is raised from 3 to 5 in Configuration 2 in an effort to evaluate how changing how the discriminator can impact the results.

The CycleGAN source code offers a data pre-processing option to flip images before training for use in data augmentation. In line with Yang & Chung (2019) flipping was turned off for Configuration 2, in addition the cropping size and load size of the spectrograms was set to 344 pixels - which was the full width of each spectrogram in the dataset. This allowed CycleGAN to train on the full sized spectrogram images rather than cropping and resizing sections of the image before training. For a fair comparison both configurations are trained for 200 epochs due to this being the default number used within the CycleGAN code and are trained on the same dataset - with the domains being CQT spectrograms of vaporwave and pop music which consisted of 2391 and 6000 images respectively.

CycleGAN Hyperparameter Experiment Results

Style transfer from vaporwave to pop was performed using both configurations. The results were compared using the FLANN matcher on 10 resultant images output from CycleGAN to the input image put into CycleGAN. As previously stated the FLANN matcher was used to find the average number of matching features to identical spectrograms from each genre to find a baseline value that can be used to compare the transformed spectrograms to. In the case of vaporwave the average number of matching features between identical spectrograms was 145, with a standard deviation of 18. Results of the feature matching experiment are shown in Table 7.4.

Both configurations performed quite similarly. The number of matching features compared to their style transferred spectrogram and the corresponding original spectrogram was approximately 90 on average, meaning there was not a significant difference in the configurations proposed in the experiment setup. Regardless, ninety matching features is

	Configuration 1	**Configuration 2**
Average Feature Matches	92.6	90

Table 7.4.: Feature Matches Compared

below the baseline value of 145 for vaporwave spectrograms meaning that there is approximately a 38% difference in the images generated by CycleGAN in the case of transferring vaporwave spectrograms to the pop genre. Going forward with the completed pipeline, the hyperparameters from Configuration 2 were chosen because it achieved the lowest feature matching average in comparison to Configuration 1, which was interpreted as it being more effective at selective remixing.

7.3.2. CQTGAN Sample Rates Experiment

As defined by research question **R3** and proposed by condition **C5** the quality of sound created from the system must be high enough to be considered music. From section 2.3 a higher sampling rate is interlinked with accurately captured audio - with a majority of music being recorded at 44.1 kHz. However sampling rates for all research described in chapter 4 are often low with Huang et al. (2018), Mor et al. (2018) and Vande Veire et al. (2019) all opting for 16kHz audio. This is likely due to 16kHz of audio containing enough information to retain high quality sound but not enough to be ineffective for model training. While the sampling rate is not the only defining factor for audio quality there is reason to assume that some experimentation in using higher sample rates could have an impact on the overall quality of the audio. As such three experiments are performed using differing sampling rates.

Experiment Setup

To setup the experiment three separate models of CQTGAN were trained with on 16kHz, 22.05kHz and 44.1kHz sample rate audio. 16kHz is chosen to emulate the sampling rates used by previously mentioned studies, while 44.1kHz is chosen due to its popular usage as a sample rate for music (section 2.3) and 22.05kHz is chosen due to another commonly referred to sample rate.

Music from the electronic genre was used to train all of the models (with the default hyperparameters and no architectural changes). Three separate versions of the tracks must also be created with the previously mentioned sample rates. Using FFmpeg, copies of the electronic music from the dataset were made and the 44.1kHz samples were converted into 22.05kHz and 16kHz versions. PEAQ analysis was used to compare the results from each model at reconstructing the same track from the test set. All models were trained for 500 epochs with the default parameters and architecture of the model. ODG values were calculated using the original version of the track from the dataset as ground truth and present the results in Table 7.5.

Experiment Results

From the results presented in Table 7.5 both 16kHz and 22.05kHz placed within *perceptible but annoying* according to their ODG scores, while the 44.1kHz examples place the slowest at tiers lower into *annoying*. The results support 16kHz and 22.05kHz being the most effective sample rates for reconstructing the audio tracks while 44.1kHz is implied to be less suited. To explain why 44.1kHz may be unsuitable for generating high quality reconstructions consider that a higher sample rate increases the complexity and amount of information stored within an audio file. This increase in complexity is likely unsuited to CQTGAN's model, as the generator could be incapable of creating raw audio that meets the complexity expected of 44.1kHz audio and the discriminator architecture may not be able to handle the feature mapping of the wider range of frequencies present in a 44.1kHz waveform. Following this logic it would be expected that the 16kHz samples would be easier for the model to generate samples for, thus leading to higher quality. From training the 16kHz samples caused the model to reach intelligible much results quicker than the 22.05kHz and 44.1kHz models. The discriminator and generator loss for all three models is shown in Figure 7.4 and Figure 7.5. The 44.1kHz discriminator loss jumps between high and low values much more sporadically compared to the other models, while its generator loss is more consistent in comparison. There is a possibility that its discriminator may be failing to notice lower quality audio sufficiently due to being unable to properly extract the depth of information made available in 44.1kHz audio.

Sample Rate (kHz)	Objective Difference Grade Average
16	-2.136
22.05	-2.360
44.1	-2.922

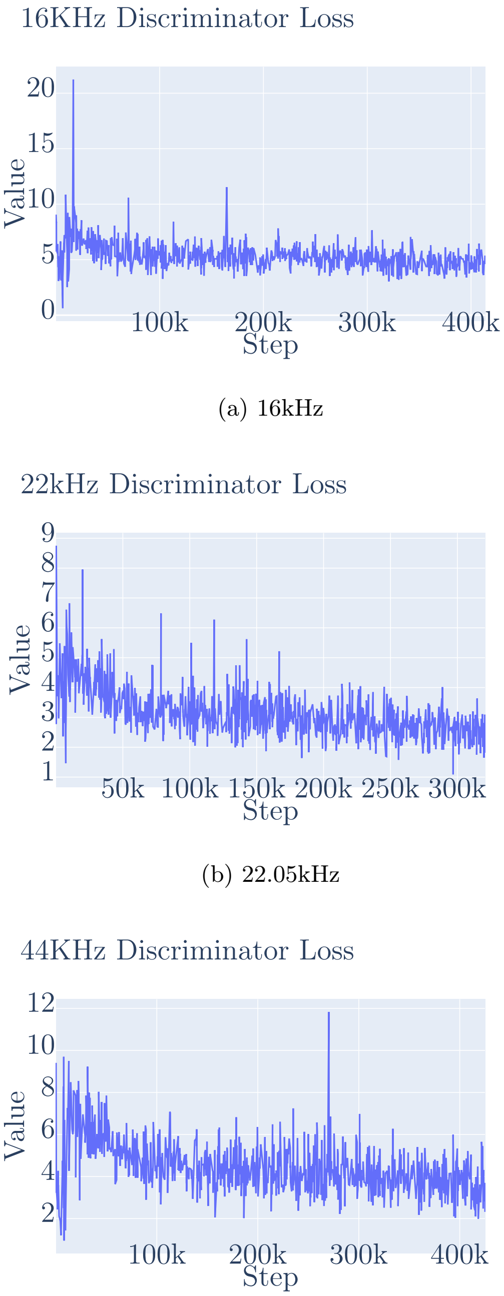
Table 7.5.: ODG Comparison of Sample Rates in CQTGAN

7.3.3. Unseen Audio Examples Experiment

The last experiment performed on the pipeline is the use of unseen audio examples to evaluate how well the system meets conditions **C3**, **C4** and **C5** which are the conditions used to judge the applicability of the model.

Experiment Setup

For all of the genres represented in the dataset unseen examples of each are experimented with and a PEAQ and audio fingerprinting analysis is performed. All unseen examples are sourced from the newest tracks from the Free Music Archive to ensure that they are not present in the dataset. Five tracks from five different artists from each of the genres represented in the dataset are taken and transferred into another genre. All tracks are converted to 22.05kHz wav files from 44.1kHz mp3 files. PEAQ analysis

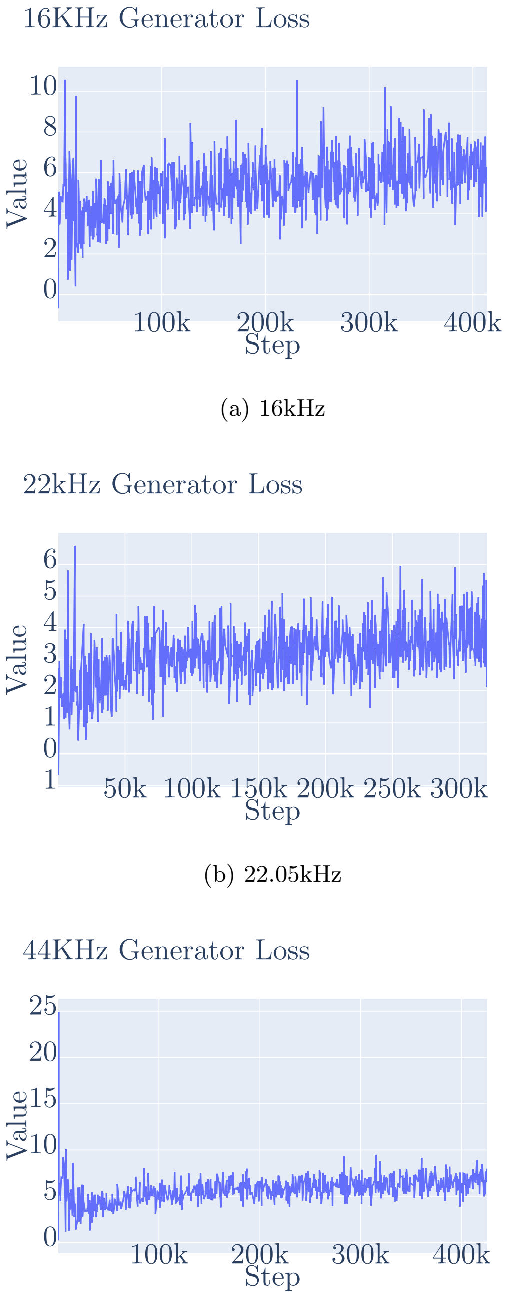


(a) 16kHz

(b) 22.05kHz

(c) 44.1kHz

Figure 7.4.: Sample Rate Discriminator Loss



(a) 16kHz

(b) 22.05kHz

(c) 44.1kHz

Figure 7.5.: Sample Rate Generator Loss

using GstPEAQ is performed on the results produced from the pipeline, with the aim of evaluating the quality of the audio. The average ODG score is taken using the original audio as a reference file. Similarly, audio fingerprinting is performed to determine how effectively selective remixing is performed. The original audio is used as a reference which is compared the output of the pipeline and a similarity score is produced.

Experiment Results

From the results shown in Table 7.6 the genre transfer that produced the highest quality of audio was electronic to instrumental which had an average ODG value of -2.913 which would be classified as *annoying* in the judgement of impairment scale. Transferring pop to hip-hop and vaporwave to hip-hop also seemed to produce *annoying* audio, meanwhile hip-hop to pop and instrumental to vaporwave examples seemed to be closer to *very annoying* on the scale. Interestingly, transferring electronic genre music to instrumental also provided the highest average similarity value of 51.4%. From personally listening to the tracks there is a possibility that the genre transfer performed by CycleGAN is not making a noticeable impact to the spectrograms in comparison to the other genre transfers, with instead most of the audio similarity difference coming from the reduced quality in audio.

The genre transfer that had the smallest similarity value was the instrumental to vaporwave music, which had an extremely low 13.2%. Upon listening to the audio there was not a difference made in composition but rather the timbre and sound, which was entirely different in all of the remixed tracks, which may be why the similarity value is so low. This genre transfer was also responsible for the lowest ODG value so it also possible that the poor quality of the reconstructed tracks could have impacted the similarity, although this was not the case for other genre tranfers, which kept their similarity value at approximately 50%.

From the results it appears that the pipeline performs quite poorly at meeting condition **C5** due to its low audio quality with all of the genre transfers attempted. Some success at meeting condition **C3** was shown because of the similarities noticed in the audio fingerprinting comparison. Within the experiments performed in this chapter, objective measures are used via PEAQ. So conversely, a subjective evaluation of the pipeline was performed in chapter 8 to complement the objective measures and develop more discussion on how well the system meets the project goal.

Genre Transferred	Average ODG	Average Similarity (%)
Electronic → Instrumental	-2.913	51.4
Hip-Hop → Pop	-3.593	49.2
Instrumental → Vaporwave	-3.612	13.2
Pop → Hip-Hop	-2.917	50.6
Vaporwave → Hip-Hop	-3.016	49.8

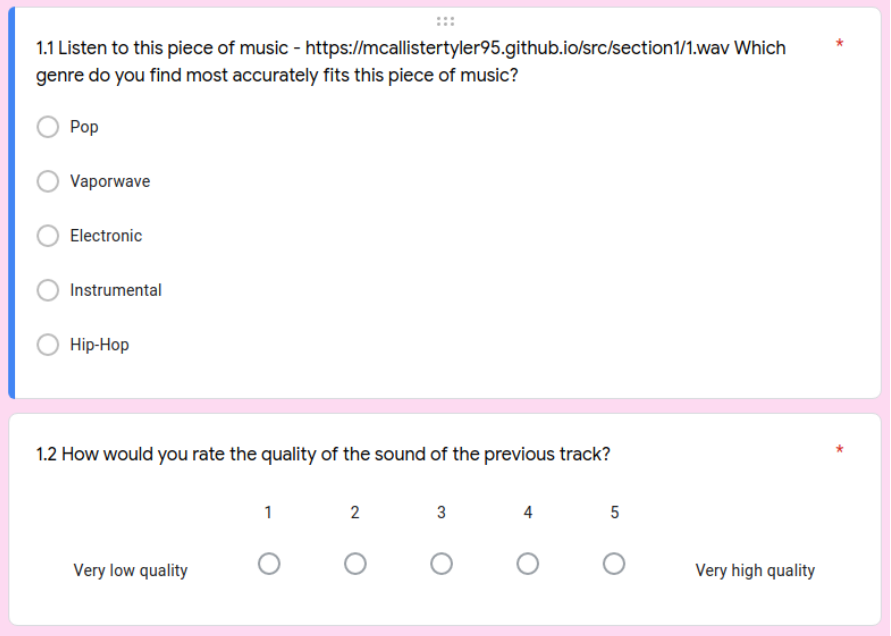
Table 7.6.: Analysis of Unseen Examples

Chapter 8: Survey Evaluation

After the previous experiments on the system, which were all based on using objective measures, a subjective evaluation was performed. Gathering mean opinion scores (subsection 4.5.2) (MOS) using surveys were a common trend from research in the literature review, with almost all related to music using MOS to get human verified evaluations of the audio quality. Similarly, looking back to the motivations (section 1.1) and project goal outlined in the introduction there was a necessity for evaluation by human participants. Were the system to be utilised for selective remixing, it would need to adequately meet the project goals from the perspective of a human user (i.e. a human user should consider the music produced to be high quality and selectively remixed). Additionally the objective measures from the previous experiments cannot be considered valuable information on their own because one's perception of music is largely subjective. Therefore, a survey was created and made available online with the aim of further evaluating conditions **C3** and **C5** by asking participants to rate the quality of the music created from the system and attempt to classify it into one of five genres.

8.1. Survey Setup

With evaluating the previously mentioned conditions in mind a survey was created using Google Forms[1]. The structure of the survey was split into two sections: *Genre Identification and Quality Evaluation* and *Similarity Comparison*. The first section asked five sets of two questions (example shown in Figure 8.1). Each set involved supplying the



1.1 Listen to this piece of music - https://mcallistertyler95.github.io/src/section1/1.wav Which genre do you find most accurately fits this piece of music? *

○ Pop

○ Vaporwave

○ Electronic

○ Instrumental

○ Hip-Hop

1.2 How would you rate the quality of the sound of the previous track? *

	1	2	3	4	5	
Very low quality	○	○	○	○	○	Very high quality

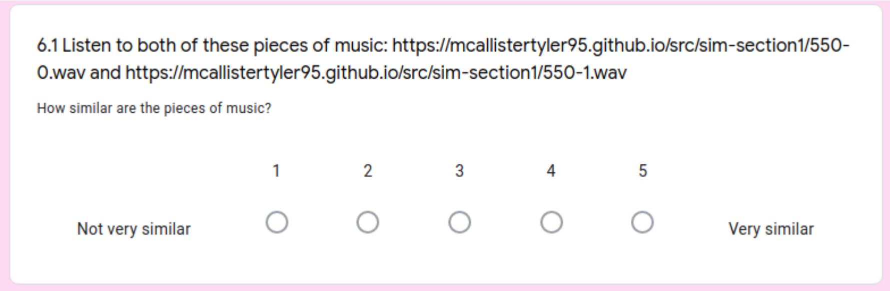
Figure 8.1.: Example Survey Section 1 Questions

[1] https://www.google.com/forms/about/

participant with a full-length piece of audio output from the system and asking them to fit it into one of five genres, and then judge its quality using a MOS scale. It should be noted that before the survey began each participant was given three 30 second length examples of each genre of music taken from the dataset, with the aim of familiarising the participant with genres they were not familiar with.

The aim of the first question of this section was to judge how noticeable the genre transfer performed was. It was expected that a successfully remixed piece of music would contain elements of its original genre and the genre it was transferred to, meaning a well-received result should reflect this by having its origin genre and transferred genre as the most chosen options. The second question asked for a quality rating using the MOS scale, with 1 being very low quality and 5 being very high quality.

The second section of the survey (example in Figure 8.2) asked the participants to listen to the original version of a track and a genre transferred version of it, and then rate their similarity using a five-point Likert scale with 1 representing no similarity at all and 5 representing high similarity. The aim of this question was to determine how effectively the track had been remixed. Recalling the definition of selective remixing (section 1.2) the remixed version of the audio should be a reinterpretation of its original form, so there should be some similarity present between both tracks.



6.1 Listen to both of these pieces of music: https://mcallistertyler95.github.io/src/sim-section1/550-0.wav and https://mcallistertyler95.github.io/src/sim-section1/550-1.wav

How similar are the pieces of music?

	1	2	3	4	5	
Not very similar	○	○	○	○	○	Very similar

Figure 8.2.: Example Survey Section 2 Question

From the survey, the population it was aimed to were people of any age and gender that were capable of listening to music. The survey was spread across friends and family and was posted on the website Reddit[2] on two community sections of the site (called subreddits) designed for posting surveys and questionnaires, named Take My Survey and Sample Size. Being spread in such a way meant the sample subset of the survey consisted predominantly of males aged between 20-50. The bias present from this demographic is discussed in the next section. The survey was left open for two weeks and garnered twenty-four responses.

[2] www.reddit.com

8.2. Survey Bias

As previously mentioned, because of the way the survey was distributed (to family members, friends and online communities) the sampling bias must be taken into account before results are discussed. The demographic of the survey was judged to be predominantly males aged from 20-50 years old (supported by a demographics study performed on Reddit users (Barthel et al. 2016). Due to being spread among participants who know me personally there was a possibility that some results carried a response bias that could potentially impact their validity. For example, respondents who wished for their answers to meet some expected standard, such as being nice or forgiving, rather than being truthful. A bias was also likely present from the Reddit respondents due to the voluntary nature of the survey, meaning some nonresponse bias was also present due to only one type of respondent being present in the sample (those who browse survey answering communities). Finally, the low sample size of respondents carried some significant sample error when compared to the population the survey aimed to target. With the bias contextualising the results a discussion on them is performed in the next section, and how well the results help meet the project conditions.

8.3. Survey Results

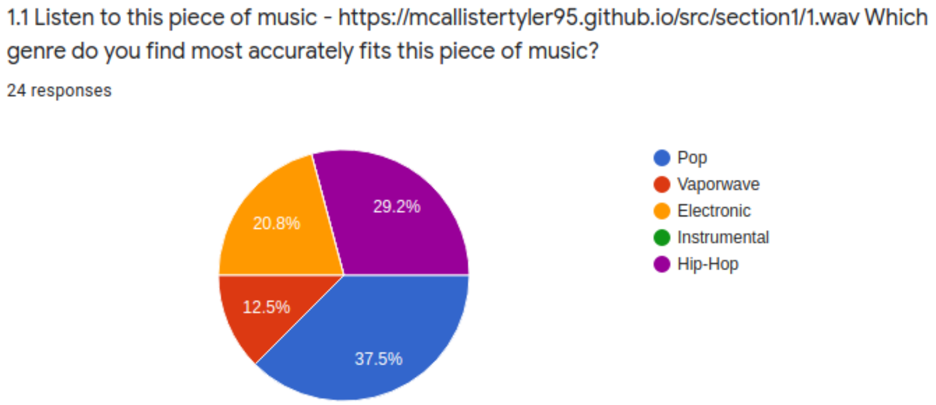
The results from each question in the survey are summarised in the tables below. Each subsection explores each of the questions and makes a conclusion from their results. In Appendix A the results from each question are presented from the Google Form. Noteworthy results are presented throughout this section.

8.3.1. Genre Identification Results

Genre identification was performed using ten tracks from the dataset transferred to a differing genre with the purpose of evaluating how well the system meets condition **C3** with human participants. To fit into the definition of selective remixing it was expected that a suitable system would be capable of altering the genre of the original audio while keeping elements of its origin genre. Therefore, an ideal output from the implemented system was considered to be a piece of audio that is mostly identified by its transferred genre[3] while also having some participants identify it to its origin genre. All results from the genre identification are shown in Table 8.1, where the number of times a genre was chosen for a particular track is listed.

50% of all of the examples had their original genre and transferred genre as the most selected genre by participants. The hip-hop track that was transferred to pop had the most even representation with 37.5% of participants identifying it as pop music while 29.2% thought it was hip-hop. The results from the form are shown in Figure 8.3. Similarly, the electronic to instrumental track had a larger 45.8% of participants identify it as instrumental music, while 29.2% thought it was electronic. From these tracks it can

[3] The term transferred genre is used to mean the genre that is applied by the style transfer system within the pipeline.



1.1 Listen to this piece of music - https://mcallistertyler95.github.io/src/section1/1.wav Which genre do you find most accurately fits this piece of music?

24 responses

Figure 8.3.: Hip-Hop to Pop results

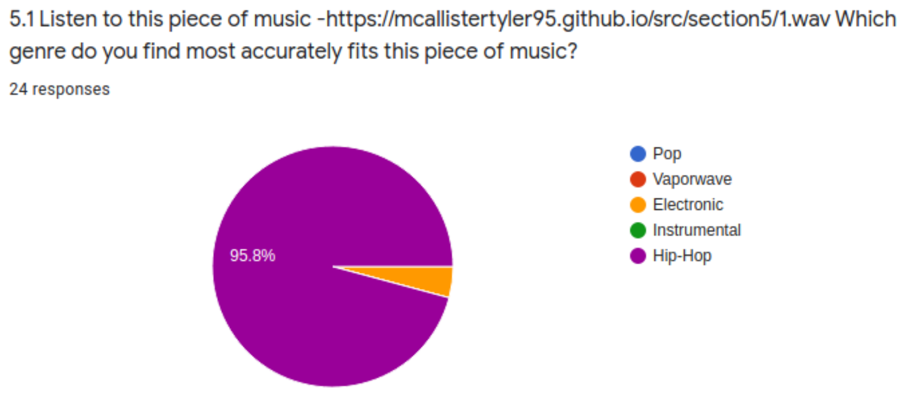
be seen that the genre transfer is showing signs of being quite successful.

Genre Transferred	Genre Selected				
	Vaporwave	Pop	Electronic	Instrumental	Hip-Hop
Hip-Hop → Pop	3	9	5	0	7
Vaporwave → Instrumental	9	2	3	6	4
Electronic → Vaporwave	14	3	6	1	0
Hip-Hop → Vaporwave	3	3	6	1	11
Vaporwave → Pop	12	5	6	1	0
Electronic → Instrumental	3	3	7	11	0
Vaporwave → Hip-Hop	5	0	14	2	3
Vaporwave → Electronic	3	2	12	5	2
Pop → Hip-Hop	0	0	1	0	23
Instrumental → Electronic	13	2	6	2	1

Table 8.1.: Genre Identification Results

However, some tracks were unsuccessful when having their genre transferred. The vaporwave to pop track was most commonly identified as either vaporwave or electronic with only 20.8% identifying it as pop music.

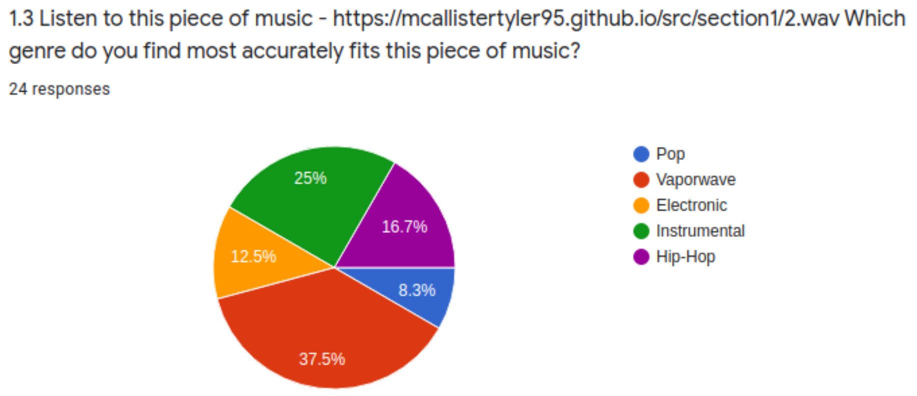
One surprising case was the pop to hip-hop track that was identified as hip-hop by all participants but one, who thought it was pop. Figure 8.4 shows the results from the survey. No other results show such a strong transfer so instead of assuming that the pop track was fully transferred to appear as a hip-hop track it was assumed that the pop track chosen for transfer may have already included some hip-hop elements, which would have only been exacerbated when hip-hop elements were applied by the system. A further investigation into this showed that the pop song chosen contained a lot of hip-hop elements such as long beats and singing, which likely lead to the creation of a very hip-hop like track.


5.1 Listen to this piece of music -https://mcallistertyler95.github.io/src/section5/1.wav Which genre do you find most accurately fits this piece of music?

24 responses

Figure 8.4.: Pop to Hip-Hop results

Some tracks received very mixed results. The electronic to vaporwave example was primarily chosen as vaporwave by 37.5% participants but received results for every other genre (shown in Figure 8.5). To investigate why this track received such varying results the example was listened to. It involved a short drum beat with distant vocals. It could be argued that there wasn't enough defining information in the track that fit it into decisively into one genre, meaning most participants that were not familiar with electronic or vaporwave music could have perceived the track much differently.


1.3 Listen to this piece of music - https://mcallistertyler95.github.io/src/section1/2.wav Which genre do you find most accurately fits this piece of music?

24 responses

Figure 8.5.: Electronic to Vaporwave results

Electronic and instrumental seemed to be the genres that were most effective at being identified when used as the transfer genre as both were in the top two results for every example when they were used. It is possible that these genres produce the best selective remixes out of all of the genres.

8.3.2. Mean Opinion Score Audio Quality Results

The MOS results to determine the audio quality of each track show that overall, the tracks were considered to be low quality, with some exceptions. Results for each track are shown in Table 8.2 with the frequency of the opinion scores chosen from the survey, along with the calculated mean opinion score of each track. The mean opinion score of the entire system was 2.237 which is classified as poor audio quality from the MOS scale in subsection 4.5.2. The vaporwave to instrumental track had the highest MOS out of every other track, reaching 2.958 putting it close to *fair* on the MOS scale. 58.3% of participants gave it a quality rating of 3 or higher which is quite significant compared to most of the other tracks. The tracks that had the lowest quality out of all the results were the vaporwave track transferred to hip-hop and the hip-hop track transferred to pop, which had a MOS of 1.916 and 1.875 respectively. From Table 8.1 the vaporwave to hip-hop track was also the track that was identified incorrectly the most out of all of the other tracks - it is possible that these results may be correlated, in that poor quality tracks do not often have a clearly identifiable genre. However, this did not seem to be the case for the hip-hop to pop track which was one of the best identified tracks in the survey. Instead, the hip-hop and pop genres may be much easier to identify despite the loss of quality while vaporwave being mixed with hip-hop could have resulted in a lower quality sound in addition to an unclear mixture of timbre and composition — causing most participants to misclassify it and consider it low quality. Listening to the vaporwave

Genre Transferred	Opinion Score					Mean
	1	2	3	4	5	
Hip-Hop → Pop	9	12	1	1	1	1.875
Vaporwave → Instrumental	3	7	4	8	2	2.958
Electronic → Vaporwave	7	10	5	1	1	2.125
Hip-Hop → Vaporwave	4	7	10	2	1	2.541
Vaporwave → Pop	8	9	4	2	1	2.125
Electronic → Instrumental	6	6	10	1	1	2.375
Vaporwave → Hip-Hop	11	7	4	1	1	1.916
Vaporwave → Electronic	9	6	7	1	1	2.125
Pop → Hip-Hop	10	5	8	0	1	2.041
Instrumental → Electronic	5	11	5	2	1	2.291

Table 8.2.: MOS Audio Quality Results

to hip-hop track seemed to confirm this as its sound was sparse and contained little instrumentation or singing. Conversely, the hip-hop to pop track contained distinct beats and a more clear composition despite having low quality sound. Ultimately, some genres may not be very compatible with each other since there were cases where a vaporwave track performed well in both audio quality and genre identification. For example the vaporwave to instrumental track had the highest MOS out of all other examples and was one of the best performing from the genre identification results.

In Figure 8.6 a histogram of all of the opinion scores chosen by the participants is shown. It is immediately clear from the distribution that low quality scores were primarily the most selected by far with 63.3% of all selections being low quality audio (2 or lower). When paired with the PEAQ analysis results from subsection 7.3.3, the system seemed to be commonly outputting low quality audio reconstructions.

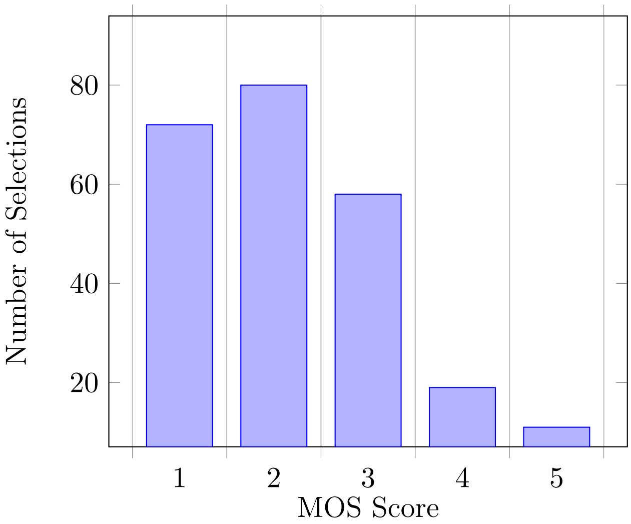


Figure 8.6.: Opinion Score Histogram

8.3.3. Similarity Comparison Results

Finally the results from the similarity comparison section of the survey are presented (Table 8.3 and Figure 8.7). Only four tracks of audio were used for the similarity comparison and all were different tracks from the ones used in the previous section of the survey. The first two tracks (electronic to instrumental and vaporwave to hip-hop) performed quite well with participants finding some degree of similarity. The first track only had two participants that found it to be not very similar to its original track (having a similarity of 2 or less). Interestingly, in this case the vaporwave to hip-hop track achieved one of the highest similarity values, while a different track of the same genre transfer performed poorly in the genre identification question. Because both tracks were different this is likely due to differences in the tracks, with one being better suited to genre transfer. The track that performed poorest was the pop to hip-hop track which was not considered to be similar to its original track at all.

Genre Transferred	Similarity Value				
	1	2	3	4	5
Electronic → Instrumental	0	2	10	11	1
Vaporwave → Hip-Hop	1	5	9	8	1
Pop → Hip-Hop	10	10	3	0	1
Hip-Hop → Vaporwave	0	5	7	7	5

Table 8.3.: Similarity Comparisons

From the histogram in Figure 8.7, very few tracks were ever chosen to be very similar (5 similarity value) to their original but this also seemed to be the case for having no similarity (1 similarity value). Primarily, a similarity value of 3 was the most prominently chosen option by participants meaning that some similarity was perceptible but it was never considered to be significantly similar to the original track. The possibility of audio quality having an impact on these results is discussed in the succeeding chapter.

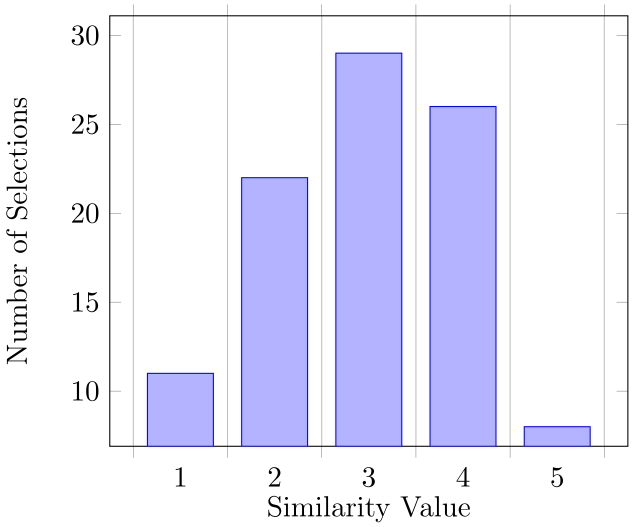


Figure 8.7.: Similarity Comparison Histogram

Chapter 9: Discussion

Given the results from the experiments and survey, an evaluation of the system was performed. Each research question and condition was taken into account when evaluating the system in its entirety. Each subsection in this chapter details an analysis of each research question and condition that was made to conclude whether the system met the project goal or not.

9.1. System Evaluation and Discussion

Three subsections makeup the evaluation of the system. Audio quality (subsection 9.1.1) and genre transfer (subsection 9.1.2) were two metrics used to evaluate the system's performance based on the project goal.

Limitations (subsection 9.1.3) and weaknesses within the implemented system, the experiments and survey were also taken into consideration when evaluating the system. Following this discussion, a focus on suggestions for improvement to remedy these limitations were also taken into account in the conclusion chapter (chapter 10).

9.1.1. Audio Quality

Audio quality was one of the conditions and research questions proposed at the beginning of the project. As such the evaluation of audio quality was one of the most determining factors for evaluating the system. Results from subsection 7.3.3 and subsection 8.3.2 were used to evaluate the quality of the audio generated from the system. The average objective difference grade (ODG) was -3.210 (*annoying*) from the experiment, while an average mean opinion score (MOS) of 2.237 (*poor*) was found as a result of the survey. To understand why the system created low quality audio a number of factors were explored to determine what could have been responsible for it. Previous research from the literature review was re-examined to look for weaknesses that stood out in the implemented system, in comparison to other similar systems.

Kumar et al. (2019) used MelGAN to achieve an average MOS of 3.49 (*fair*) on unseen audio, one rank above the MOS achieved by the implemented system. Due to the similarity in model architecture, a comparison of MelGAN and CQTGAN's training was made to determine a reason for the poor audio quality created by the system. The entirety of MelGAN's audio quality evaluation was based on speech recorded at 16kHz, in addition all of the speech data used had a monophonic texture. In subsection 7.3.2 lower sample rates seemed to converge to intelligible results much quicker and had better ODG scores than higher sample rates. Both the audio texture and sample rate used in the original MelGAN paper may have played a role in creating higher quality audio results. Increased sample rates and audio textures above monophonic likely increased the complexity of the audio, resulting in a decline in generated audio quality.

Huang et al. (2018), implemented a similar system named TimbreTron, and trained it entirely on 16kHz audio of piano, harpsichord, violin and flute music. While no audio quality evaluation was performed on TimbreTron its results were personally considered to be of reasonable quality[1].

Ultimately, the poor audio quality output from the system was decided to be due to the use of 22.05kHz audio paired with the use of homophonic and polyphonic textures which impacted the audio quality greatly. To support this, the results of using the instrumental genre as the transferred genre produced the best ODG scores in subsection 7.3.3 and one of the highest MOS score in Table 4.1. It is possible that some of the instrumental examples contain monophonic textures which would be reasonable to believe, as the instrumental genre is commonly focused on one instrument playing. This further supports that the increase in audio texture paired with higher sample rate was the cause of poor audio quality.

9.1.2. Genre Transfer

Evaluating the system on its potential for performing genre transfer was heavily dependent on the results from the survey. While objective audio quality measures are more common within research, this was not the case for objective methods of genre classification which are still an active research topic (Cano et al. 2006, Pachet et al. 1999). Therefore, a higher reliance was put onto subjective measures for this evaluation.

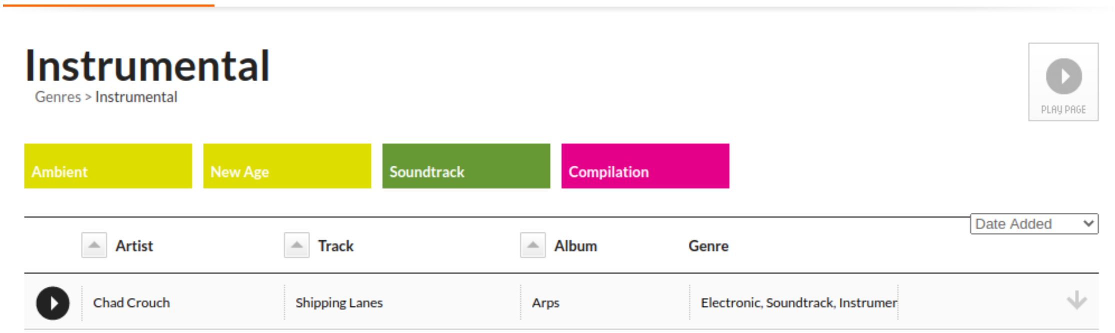
In subsection 7.1.2 the most effective transferred genres were instrumental and electronic, although there is some variability throughout the results. Whenever the vaporwave genre was used, either as a transferred genre or as the original genre, it received a large amount of selections from the participants. In the case of electronic and instrumental music being well suited for use as the transferred genre, it was worth considering that the genres could be very broad in definition. A look at examples[2] on the Free Music Archive (FMA) gave a variety of tracks with multiple subgenres, all of which come under the instrumental genre. In Figure 9.1 an example of this is given, in which one track is classified under three genres - Electronic, Soundtrack and Instrumental but is listed under the Instrumental genre category regardless. All tracks from the small version of the dataset used (Benzi et al. 2016) only contained a singular genre in their labelling, but the use of multiple genres from the FMA site shows that these were likely removed in the dataset used.

Consequently, some transfers were identified incorrectly, such as the vaporwave to hip-hop track, which was identified as electronic fourteen times, and the instrumental to electronic track which was identified as vaporwave thirteen times. Glitsos (2018) considers the vaporwave genre to be very distinct in nature calling it:

> “a genre that emerges from a host of heavily intertextual electronic musics available since the turn of the millennium” (Glitsos 2018, p.100)

As such it made sense that vaporwave and electronic were commonly mistaken for each other. Despite some key differences in the genres, vaporwave is still heavily within the domain of electronic music which could have confused survey participants.

[1] https://www.cs.toronto.edu/ huang/TimbreTron/samples_page.html
[2] https://freemusicarchive.org/genre/Instrumental



	Artist	Track	Album	Genre
	Chad Crouch	Shipping Lanes	Arps	Electronic, Soundtrack, Instrumer

Figure 9.1.: FMA Genre Example

The similarity comparison performed in the survey (Table 8.3) showed that the instrumental genre being applied to electronic created the highest amount of similarity out of all tracks, leading to the possibility that instrumental music was best suited as a transferred genre.

Huang et al. (2018) asked questions involving similarity in their research to determine how effectively the timbre of the music was changed while keeping the structure of the original track. They expected results to be difficult to evaluate due to the perceptual differences involved in using human participants. In a similar vein, for selective remixing, some degree of similarity to the original track was expected, but there was also a concern that a genre transferred track being too similar to its original could mean that no significant change was made by the system. This created some uncertainty in the effectiveness of the survey question, which is discussed further in subsection 9.1.3. Regardless, in a majority of cases the participants detected some similarity in the tracks but rarely considered any two tracks to be very similar or complete dissimilar. Due to this the genre transfer was considered to be successful as a whole.

9.1.3. Limitations

Some limitations were present throughout the system, experiments and survey. The first being that concept of music genre is difficult to define and is naïvely defined for the genre labels present in the dataset used for training. Scaringella et al. (2006) claim that boundaries between genre taxonomies are fuzzy, claiming genre classification to be a non-trivial task. Additionally they claim that fitting a song into one specific genre is questionable. Applying these claims to the dataset showed some clear weaknesses, as all genre labels were lifted directly from the Free Music Archive. Without musical genre being a well-defined taxonomy accurate genre labelling was not possible on the dataset. Looking at results from the survey there is evidence of multiple misclassifications, which are likely based on genre taxonomies being interpreted differently by the participants.

Another limitation included multiple models needing to be trained for use in genre transfer. For example five CQTGAN models had to be trained to reconstruct each genre and ten CycleGAN models were needed for each possible genre transfer. This impacted training and experimentation considerably. Making model modifications incurred all models to be retrained as opposed to one and all experiments involving audio quality or music genre had to be run multiple times. Also, testing all possible combinations of genre transfer was considered to be excessive for a survey, so some meaningful experiments could not be performed. For example instrumental music is only used as the origin genre in a genre transfer task once.

Audio duration was another limitation caused by the CQTGAN model. From the architecture used in the MelGAN model only the same static size of audio could be used. For example, a spectrogram's length will differ depending on the length of audio it was created from. Using audio of varying length to train CQTGAN would commonly break the model. To remedy this, all training data samples were split into to four second snippets, which caused the model to only be capable of generating four seconds of audio. To create full length samples with any piece of audio longer than four seconds FFmpeg is used to split the track into segments, which are then fed into CQTGAN individually and concatenated together at the end.

Lastly, a more objective method for genre classification could have been performed. Brunner et al. (2018) implemented a neural network to classify genres to evaluate their genre transfer system. This could have been done to determine more substantial results for the genre transfer evaluation of the system as well as create a more in-depth discussion on the differences between human perception of genre compared to an objective method.

9.2. Research Questions

After evaluating the system a look at each research question was done with the aim of determining how well they were answered throughout the thesis.

Research Question R1

How are raw audio waveforms generated in other deep learning music generation systems?

To answer Research Question R1 the literature review (section 4.1) mentioned various distinct deep learning systems that have been used to create raw audio waveform. WaveNet (van den Oord et al. 2016) and WaveGAN (Donahue et al. 2018) were two deep learning systems capable of generating and training on raw audio waveform. Some limitations present in both were, their lengthy training times and relatively poor audio quality. Alongside those systems, three deep learning systems were also covered, which focused on spectrogram reconstruction, MelGAN (Kumar et al. 2019), WaveNet Vocoder (Wang et al. 2017) and WaveGlow (Prenger et al. 2018). All three models claimed to be capable of creating high quality audio, using MOS to compare their results to real speaker audio. This made them more favourable as a choice for the implemented system over the two non-spectrogram models.

Spectrograms of audio became an essential part of development due to this. By using them as an intermediate representation they could be represented as an image (supporting research question **R2**) and reconstructed back into audio. In subsection 7.1.2 experiments were performed to find the best spectrogram reconstruction method, along with evaluating the best type of spectrogram that could be used for music. As a result a modified MelGAN model conditioned on constant-Q spectrograms, named CQTGAN was created. CQTGAN was capable of creating higher quality music from CQT spectrograms than mel spectrograms, and was slightly superior to the WaveNet model conditioned on the same spectrograms. To meet conditions **C1**, **C2** and **C5** CQTGAN was used as the spectrogram reconstruction system in the pipeline of the completed system. As such, this research question was considered to be answered and contributed to the aforementioned conditions.

Research Question R2

How can selective remixing be performed using deep learning?

Looking back at the interpretation of selective remixing from section 1.2 investigating research question R2 involved looking at style transfer methods utilised by deep learning systems. Selective remixing was performed in state-of-the-art research, as shown by subsection 4.3.5 and subsection 4.3.7. In both studies CycleGAN (Zhu et al. 2017) was used to achieve genre transfer using spectrogram images which motivated the use of spectrogram reconstruction models like WaveNet.

Exploring these methods of style transfer lead to experimentation with CycleGAN along with another similar model, StarGAN (Choi et al. 2018). Both models were comparable in their performance (section 7.2) but CycleGAN was chosen due to its effectiveness in previous research. To evaluate its capability in selective remixing, feature matching was performed via FLANN (Muja & Lowe 2009) to evaluate the extent of modifications performed on spectrograms by the style transfer system. Results from the experiment showed that there was a noticeable impact on the spectrograms created from the model. Similarly, the results from the survey showed that at least 50% of participants were able to identify the transferred genre of the music output from the system. Due to this, research question **R2** was considered to be adequately answered, as CycleGAN proved to be an effective tool for selective remixing and contributes to condition **C1** and **C3**.

Research Question R3

Can high quality convincing remixed generated music via deep learning methods be reasonably evaluated?

Linking to condition **C5**, the evaluation audio quality was explored in subsection 4.5.3 which consisted of three objective measures. PEAQ (Thiede et al. 2000), POLQA (Beerends et al. 2013) and ViSQOL (Hines et al. 2015) were all discussed along with an explanation of the purpose of objective difference grade (ODG) scores. Subjective audio quality measures were also discussed in subsection 4.5.2 which involved utilising a five-point scale to measure the quality of audio among human participants. Both means of measuring audio signal quality were utilised throughout numerous experiments and in the survey evaluation of the pipeline. PEAQ was used to calculate ODG scores in numerous experiments (section 7.2, subsection 7.3.2 and subsection 7.3.3). Meanwhile, MOS scores were gathered in one of the survey questions (subsection 8.3.2).

In addition to audio quality, the success of the genre transfer achieved by the system is evaluated via audio fingerprinting (Foote 1997) and a genre identification test in the survey. With all of this taken into account the previously mentioned methods were sufficient to answer whether generated music can be reasonably evaluated.

9.3. Project Goal and Conditions

Create a deep learning system capable of remixing and creating high quality samples of modern genres of music.

All three research questions were proposed to aid in the completion of the project goal. In addition, five conditions were created which acted as prerequisites that had to be met before the project goal could be considered complete. Four of the five conditions were confidently met, with the unfulfilled condition being the most difficult to achieve. Overall this means the project goal was not fully met, but the system implemented was considered to be a modest attempt at reaching the goal. The audio texture and sample rate of the data used (subsection 9.1.1) to train the system were considered to be the main contributors that lead to low quality audio being created by the system. Regardless, the system was capable of performing genre transfer and outputting remixed music. Further development into the creation of high quality tracks was considered to be one way in which the system could be improved upon (chapter 10).

The next five subsections provide insight into each condition to determine whether it was met by the implemented system.

Condition C1

Deep learning must be one of the key characteristics of the implemented system.

The completed architecture of the system takes the form of a pipeline (Figure 7.2) consisting of two deep learning models — CQTGAN and a modified CycleGAN model. Taking the definition of deep learning from Deng & Yu (2014):

> “A class of machine learning techniques that exploit many layers of non-linear information processing for supervised or unsupervised feature extraction and transformation and for pattern analysis and classification.”
> (Deng & Yu 2014, p.199)

Both CQTGAN and CycleGAN fit into this definition of deep learning, with both containing many layers and performing non-linear operations using spectrogram and image data respectively. Although both are loosely coupled and could presumably be swapped out for non deep learning related systems they were considered key characteristics to the implemented system.

Condition C2

The system must output audio waveform.

Using CQT spectrograms as an intermediate representation of audio was done so they could have image-based style transfer techniques applied to them, and allow CQTGAN to reconstruct them back into audio. 22.05kHz sample rate audio was output from the system with a maximum of four second samples. The length of audio was constrained by the architecture of the CQTGAN model. For the creation of longer audio streams inputs were split into chunks, put into the model separately and FFmpeg was used to concatenate the results back together. In spite of this limitation the condition is considered met.

Condition C3

Selective remixing via genre-transfer must be performed on audio.

As discussed in section 9.2 it was shown that most human participants of the survey detect genre changes to an original composition. In a majority of cases they also were able to identify the genre that has been applied by the style transfer system. Audio fingerprinting was also utilised to judge how similar the audio output from the system is to its original composition. CycleGAN was determined to be a suitable model for achieving genre transfer as a result. This condition was tricky to truly meet due to it specifying that the transfer must be performed “on audio”. In the system the genre transfer is performed on an image which is subsequently transformed into audio. Taking the perspective of the system to a new user, the use of audio as an input and output means the entirety of the system is usable without knowledge of spectrograms. For this reason the condition is considered to be met as the spectrogram image transfer is purely an intermediate step within the system pipeline.

Condition C4

The genres of music used must be modern and outwith the standard genres used in state-of-the-art systems.

In chapter 5 the dataset to be used for training the CQTGAN and CycleGAN models was chosen based on a number of factors, with the most prominent being the number of genres and the texture of the audio available in the dataset. The dataset chosen contained nine genres in total, although only five were used throughout experiments and the conducted survey. All of the genres selected were considered modern and outwith those present from the state-of-the-art research covered in the literature review, which commonly used classical music. Hence, condition **C4** was regarded as being completed.

Condition C5

The audio generated must be high quality.

As mentioned in section 9.2 PEAQ analysis and MOS were both methods used to estimate how effective the system was at creating high quality audio output. From the results in Table 8.2, the MOS scores overwhelmingly determined that the examples used within the survey were low quality. Supporting this, the results in Table 7.6 show that unseen examples of audio also perform quite poorly in terms of audio quality. Looking back at the style transfer techniques considered in Table 7.1, it seems that more thorough experimentation could have been performed to find a suitable audio reconstruction method. Overall this condition is not considered to be met because the current audio quality produced by the system is too low.

Chapter 10: Conclusions and Future Work

At the beginning of this thesis a proposal was made to create a deep learning system capable of creating high quality remixed music. To achieve the creation of remixed music the aim was to replicate selective remixing via genre transfer of music tracks.

A study was performed on state-of-the-art research focusing on audio waveform generation and style transfer within the audio domain (chapter 4). This brought the focus of development into using style transfer on spectrogram images of audio and approximately reconstructing them into audio waveform. Utilising image-based style transfer methods on spectrogram images of audio was considered to be a feasible method of achieving selective remixing. Constant-Q transform (CQT) spectrograms were chosen as the image representation of audio due to their superiority in representing music compared to other spectrogram types.

For the implementation of the system a three process pipeline architecture was designed, inspired by Huang et al. (2018). The first process used the nnAudio library (Cheuk et al. 2019) for generation of CQT spectrograms, the second used CycleGAN (Zhu et al. 2017) for style transfer to modify spectrograms with features from other genres. The third used a modified MelGAN (Kumar et al. 2019) model named CQTGAN, conditioned to reconstruct CQT spectrograms to audio. Using four genres of music from the FMA: Dataset (Benzi et al. 2016), and an additional genre from a sourced dataset from Bandcamp (Bandcamp, Inc. 2020) multiple models were trained. Five CQTGAN models trained on different genres of spectrogram were created along with ten CycleGAN models capable of performing bi-directional style transfer.

Experimentation concerning the audio similarity and audio quality was performed to find the best hyperparameter configurations and evaluate the model's capability on unseen audio examples. Subsequently, a survey of human participants was created to evaluate the model from a subjective perspective, to determine the audio quality and capability of genre transfer.

In summation, the implemented system was capable of performing genre transfer to some degree but produced very low quality audio. Creating high quality audio was considered to be an extremely difficult task given the nature of the audio being used (homophonic texture) with much greater sample rates and variety of genres compared to those in other studies. Despite the overall goal not being met the experiments and survey performed showed that the capability of fair audio quality is possible meaning some further developments could be made to the system to improve its quality.

Recalling the limitations discussed in subsection 9.1.3, avenues for future improvements that could be made are discussed.

Firstly, a more accurately labelled dataset should be the main concern of any future work involving genre transfer. The use of classical music or singular instruments in previous studies allowed for a much smaller scope with much more clearly defined taxonomies compared to the breadth of musical genres available. A well curated dataset could allow for easier model training and for style transfer to become more accurate and impactful. Likewise, limiting datasets to contain specific artists, or music put into a more well-defined taxonomy (e.g. a dataset based on the timbral features rather than genre) could lead to more successful genre transfer.

Training multiple models was a large obstacle when experimenting with the system which could have been avoided by more effective deep learning systems. Another model capable of multi-domain style transfer by being trained on multiple datasets (StarGAN) was created by Choi et al. (2018) but was not chosen due to previous studies opting for CycleGAN. This decision may have been rather hasty, as the consequences of training multiple models was not realised until the decision to implement the model fully into the pipeline was already made. At the time of writing, an improved StarGAN model, named StarGAN v2 (Choi et al. 2020, in press) was created by the same authors. Using this model to possibly increase the performance of the style transfer system in the pipeline, and reduce the overhead caused by multiple models is a relevant avenue for future work to take.

Bibliography

Abbado, A. (1988), Perceptual correspondences of abstract animation and synthetic sound, Master's thesis, Massachusetts Institute of Technology.

Ableton (2019), 'Ableton Live'. Accessed: 2020-11-04.
URL: *https://www.ableton.com/en/*

Agarap, A. F. (2018), 'Deep Learning using Rectified Linear Units (ReLU)', *CoRR* **abs/1803.08375**.
URL: *http://arxiv.org/abs/1803.08375*

Apple Inc (2019), 'GarageBand'. Accessed: 2020-11-04.
URL: *https://www.apple.com/mac/garageband/*

Bandcamp, Inc. (2020), 'Bandcamp',
URL: *https://bandcamp.com.* Accessed: 2020-11-04.

Barthel, M., Stoking, G., Holcomb, J. & Mitchell, A. (2016), *Reddit news users more likely to be male, young and digital in their news preferences.* Accessed 2020-04-20.
URL: *https://www.journalism.org/2016/02/25/reddit-news-users-more-likely-to-be-male-young-and-digital-in-their-news-preferences/*

Beerends, J., Schmidmer, C., Berger, J., Obermann, M., Ullmann, R., Pomy, J. & Keyhl, M. (2013), 'Perceptual Objective Listening Quality Assessment (POLQA), The Third Generation ITU-T Standard for End-to-End Speech Quality Measurement Part I-Temporal Alignment', *AES: Journal of the Audio Engineering Society* **61**, 366–384.

Benward, B. & Saker, M. (2009), *Music in Theory and Practice Eighth Edition*, number Volume 1 *in* 'Music in Theory and Practice', McGraw-Hill, pp. 145–151.

Benzi, K., Defferrard, M., Vandergheynst, P. & Bresson, X. (2016), 'FMA: A dataset for music analysis', *CoRR* **abs/1612.01840**.
URL: *http://arxiv.org/abs/1612.01840*

Berenzweig, A., Logan, B., Ellis, D., Whitman, B. & A, C. (2003), 'A large-scale evaluation of acoustic and subjective music similarity measures', *Computer Music Journal* **28**, 63–76.

Bosi, M. & Goldberg, R. E. (2002), *Introduction to Digital Audio Coding and Standards*, Kluwer Academic Publishers, USA, pp. 6–9.

Briot, J., Hadjeres, G. & Pachet, F. (2017), 'Deep learning techniques for music generation - A survey', *CoRR* **abs/1709.01620**.
URL: *http://arxiv.org/abs/1709.01620*

Brown, J. C. (1991), 'Calculation of a constant Q spectral transform', *The Journal of the Acoustical Society of America* **89**(1), 425–434.

Brunner, G., Wang, Y., Wattenhofer, R. & Zhao, S. (2018), 'Symbolic music genre transfer with cyclegan', *CoRR* **abs/1809.07575**.
URL: *http://arxiv.org/abs/1809.07575*

Cano, P. & Batlle, E. (2005), 'A review of audio fingerprinting', *Journal of VLSI Signal Processing* **41**, 271–284.

Cano, P., Gómez, E., Gouyon, F., Herrera, P., Koppenberger, M., Ong, B., Serra, X., Streich, S. & Wack, N. (2006), 'ISMIR 2004 Audio Description Contest', *Tech. Rep. MTG-TR-2006-02, Universitat Pompeu Fabra* .

Cano, P., Wack, N. & Herrera, P. (2018), 'ISMIR04 Genre Identification task dataset'. Licensed under a Creative Commons Attribution- NonCommercial-ShareAlike 1.0 Generic license.
URL: *https://doi.org/10.5281/zenodo.1302992*

Cheliotis, G. & Yew, J. (2009), An analysis of the social structure of remix culture, *in* 'Proceedings of the Fourth International Conference on Communities and Technologies', C&T '09, ACM, New York, NY, USA, pp. 165–174.
URL: *http://doi.acm.org/10.1145/1556460.1556485*

Chesney, T. (2004), '"other people benefit. i benefit from their work." sharing guitar tabs online', *Journal of Computer-Mediated Communication* **10**(1), 00–00.
URL: *https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1083-6101.2004.tb00230.x*

Cheuk, K. W., Anderson, H. H., Agres, K. & Herremans, D. (2019), 'nnaudio: An on-the-fly gpu audio to spectrogram conversion toolbox using 1d convolution neural networks', *ArXiv* **abs/1912.12055**.
URL: *https://arxiv.org/abs/1912.12055*

Choi, K., Fazekas, G., Cho, K. & Sandler, M. B. (2017), 'A tutorial on deep learning for music information retrieval', *CoRR* **abs/1709.04396**.
URL: *http://arxiv.org/abs/1709.04396*

Choi, Y., Choi, M., Kim, M., Ha, J.-W., Kim, S. & Choo, J. (2018), StarGAN: Unified Generative Adversarial Networks for Multi-Domain Image-to-Image Translation, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition', pp. 8789–8797.

Choi, Y., Uh, Y., Yoo, J. & Ha, J.-W. (2020, in press), StarGAN v2: Diverse Image Synthesis for Multiple Domains, *in* 'Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition'.

Cockos (2020), 'Reaper'. Version 6.11.
URL: *https://www.reaper.fm*

Dai, S., Zhang, Z. & Xia, G. (2018), 'Music style transfer issues: A position paper', *CoRR* **abs/1803.06841**.
URL: *http://arxiv.org/abs/1803.06841*

Deng, L. & Yu, D. (2014), *Deep Learning: Methods and Applications*, Vol. 7, Foundations and Trends in Signal Processing.

Donahue, C., McAuley, J. J. & Puckette, M. S. (2018), 'Synthesizing audio with generative adversarial networks', *CoRR* **abs/1802.04208**.
URL: *http://arxiv.org/abs/1802.04208*

Dong, H.-W., Hsiao, W.-Y., Yang, L.-C. & Yang, Y.-H. (2018), MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment, *in* 'The Thirty-Second AAAI Conference of Artificial Intelligence', pp. 34–41.
URL: *https://www.aaai.org/ocs/index.php/AAAI/AAAI18/paper/view/17286/15668*

Engel, J. H., Resnick, C., Roberts, A., Dieleman, S., Eck, D., Simonyan, K. & Norouzi, M. (2017), 'Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders', *CoRR* **abs/1704.01279**.
URL: *http://arxiv.org/abs/1704.01279*

Fagerjord, A., Klastrup, L. & Allen, M. (2010), *After Convergence: YouTube and Remix Culture*, Springer Netherlands, Dordrecht, pp. 187–200.
URL: *https://doi.org/10.1007/978-1-4020-9789-8_11*

FFmpeg Team (2020), 'FFmpeg version 4.0.1'.
URL: *https://ffmpeg.org*

Foote, J. T. (1997), Content-based retrieval of music and audio, *in* C.-C. J. Kuo, S.-F. Chang & V. N. Gudivada, eds, 'Multimedia Storage and Archiving Systems II', Vol. 3229, International Society for Optics and Photonics, SPIE, pp. 138 – 147.
URL: *https://doi.org/10.1117/12.290336*

Free Software Foundation (2020), 'GNU Bash version 4.2.46'.
URL: *https://www.gnu.org/software/bash/manual/bash.html*

Gatys, L. A., Ecker, A. S. & Bethge, M. (2015), 'Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks', *CoRR* **abs/1505.07376**.
URL: *http://arxiv.org/abs/1505.07376*

Gatys, L. A., Ecker, A. S. & Bethge, M. (2016), Image style transfer using convolutional neural networks, *in* '2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)', pp. 2414–2423.

Gemmeke, J. F., Ellis, D. P. W., Freedman, D., Jansen, A., Lawrence, W., Moore, R. C., Plakal, M. & Ritter, M. (2017), Audio set: An ontology and human-labeled dataset for audio events, *in* 'IEEE ICASSP 2017', New Orleans, LA.

Gillen, E., Damien, K., Hines, A., Skoglun, J., Kokaram, A. & Harte, N. (2015), 'ViSQOLAudio: An objective audio quality metric for low bitrate codecs', *The Journal of the*

Acoustical Society of America **137 (6)**, EL449–EL455.
URL: *http://asa.scitation.org/doi/full/10.1121/1.4921674*

Glitsos, L. (2018), 'Vaporwave, or music optimised for abandoned malls', *Popular Music* **37**(1), 100–118.

Gold, B., Morgan, N. & Ellis, D. (2011), *Speech and Audio Signal Processing: Processing and Perception of Speech and Music*, 2nd edn, Wiley-Interscience, New York, NY, USA.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S. h., Courville, A. & Bengio, Y. (2014), Generative adversarial nets, *in* Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence & K. Q. Weinberger, eds, 'Advances in Neural Information Processing Systems 27', Curran Associates, Inc., pp. 2672–2680.
URL: *http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf*

Griffin, D. & Jae Lim (1984), 'Signal estimation from modified short-time fourier transform', *IEEE Transactions on Acoustics, Speech, and Signal Processing* **32**(2), 236–243.

Grinstein, E., Duong, N. Q. K., Ozerov, A. & Pérez, P. (2018), Audio style transfer, *in* '2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)', pp. 586–590.

Heideman, M. T., Johnson, D. H. & Burrus, C. S. (1985), 'Gauss and the history of the fast fourier transform', *Archive for History of Exact Sciences* **34**(3), 265–277.
URL: *https://doi.org/10.1007/BF00348431*

Hiller, L. A. & Isaacson, L. M. (1979), *Experimental Music; Composition with an Electronic Computer*, Greenwood Publishing Group Inc., Westport, CT, USA.

Hines, A., Skoglun, J., Kokaram, A. & Harte, N. (2015), 'ViSQOL: an objective speech quality model', *EURASIP Journal on Audio, Speech, and Music Processing* **2015 (13)**, 1–18.

Holters, M. & Zölzer, U. (2015), GstPEAQ – An Open Source Implementation of the PEAQ Algorithm, *in* 'Proceedings of the 18th International Conference on Digital Audio Effects', pp. 181–184.
URL: *https://www.ntnu.edu/documents/1001201110/1266017954/DAFx15_final_ index.pdf*

Hu, Y., Huber, A. E. G., Anumula, J. & Liu, S. (2018), 'Overcoming the vanishing gradient problem in plain recurrent networks', *CoRR* **abs/1801.06105**.
URL: *http://arxiv.org/abs/1801.06105*

Huang, S., Li, Q., Anil, C., Bao, X., Oore, S. & Grosse, R. B. (2018), 'TimbreTron: A WaveNet(CycleGAN(CQT(Audio))) Pipeline for Musical Timbre Transfer', *CoRR* **abs/1811.09620**.
URL: *http://arxiv.org/abs/1811.09620*

Huzaifah, M. (2017), 'Comparison of time-frequency representations for environmental sound classification using convolutional neural networks', *CoRR* **abs/1706.07156**.
URL: *http://arxiv.org/abs/1706.07156*

Huzaifah, M. & Wyse, L. (2019), 'Applying visual domain style transfer and texture synthesis techniques to audio - insights and challenges', *CoRR* **abs/1901.10240**.
URL: *http://arxiv.org/abs/1901.10240*

International Telecommunication Union (2016), Methods for objective and subjective assessment of speech and video quality, Technical report, International Telecommunication Union.
URL: *https://www.itu.int/rec/dologin_pub.asp?lang=eid=T-REC-P.913-201603-I!!PDF-Etype=items*

International Telecommunication Union (2017), Vocabulary for performance, quality of service and quality of experience, Technical report, International Telecommunication Union.
URL: *https://www.itu.int/rec/T-REC-P.10-201711-I/en*

Isola, P., Zhu, J., Zhou, T. & Efros, A. A. (2016), 'Image-to-image translation with conditional adversarial networks', *CoRR* **abs/1611.07004**.
URL: *http://arxiv.org/abs/1611.07004*

Karras, T., Aila, T., Laine, S. & Lehtinen, J. (2017), 'Progressive growing of gans for improved quality, stability, and variation', *CoRR* **abs/1710.10196**.
URL: *http://arxiv.org/abs/1710.10196*

Karras, T., Laine, S. & Aila, T. (2018), 'A style-based generator architecture for generative adversarial networks', *CoRR* **abs/1812.04948**.
URL: *http://arxiv.org/abs/1812.04948*

Kingma, D. P. & Dhariwal, P. (2018), Glow: Generative flow with invertible 1x1 convolutions, *in* 'Advances in Neural Information Processing Systems', pp. 10215–10224.

Koenig, W., Dunn, H. K. & Lacy, L. Y. (1946), 'The sound spectrograph', *The Journal of the Acoustical Society of America* **18**(1), 19–49.
URL: *https://doi.org/10.1121/1.1916342*

Krizhevsky, A., Sutskever, I. & Hinton, G. E. (2012), Imagenet classification with deep convolutional neural networks, *in* 'Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1', NIPS'12, Curran Associates Inc., Red Hook, NY, USA, p. 1097–1105.

Kumar, K., Kumar, R., de Boissiere, T., Gestin, L., Teoh, W. Z., Sotelo, J., de Brebisson, A., Bengio, Y. & Courville, A. (2019), 'MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis'.

Lalinský, L. (2020), 'Chromaprint'. Accessed: 2020-05-25.
URL: *https://github.com/acoustid/chromaprint*

Law, E., West, K., Mandel, M. I., Bay, M. & Downie, J. S. (2009), Evaluation of algorithms using games: The case of music tagging, *in* '10th International Society for Music Information Retrieval Conference', pp. 387–392.

LeCun, Y., Boser, B., Denker, J. S., Henderson, D., Howard, R. E., Hubbard, W. & Jackel, L. D. (1989), 'Backpropagation applied to handwritten zip code recognition', *Neural Computation* **1**(4), 541–551.

Lefaivre, A. & Zhang, J. Z. (2018), Music genre classification: Genre-specific characterization and pairwise evaluation, *in* 'Proceedings of the Audio Mostly 2018 on Sound in Immersion and Emotion', AM'18, Association for Computing Machinery, New York, NY, USA.
URL: *https://doi.org/10.1145/3243274.3243310*

Levenshtein, V. (1966), 'Binary Codes Capable of Correcting Deletions, Insertions and Reversals', *Soviet Physics Doklady* **10**, 707.

Li, M., Zuo, W. & Zhang, D. (2016), 'Deep identity-aware transfer of facial attributes', *CoRR* **abs/1610.05586**.
URL: *http://arxiv.org/abs/1610.05586*

McCulloch, W. S. & Pitts, W. (1943), 'A logical calculus of the ideas immanent in nervous activity', *The bulletin of mathematical biophysics* **5**(4), 115–133.
URL: *https://doi.org/10.1007/BF02478259*

McFee, B., Lostanlen, V., McVicar, M., Metsai, A., Balke, S., Thomé, C., Raffel, C., Malek, A., Lee, D., Zalkow, F., Lee, K., Nieto, O., Mason, J., Ellis, D., Yamamoto, R., Seyfarth, S., Battenberg, E., Морозов, , Bittner, R., Choi, K., Moore, J., Wei, Z., Hidaka, S., nullmightybofo, Friesch, P., Stöter, F.-R., Hereñú, D., Kim, T., Vollrath, M. & Weiss, A. (2020), 'librosa/librosa: 0.7.2'.
URL: *https://doi.org/10.5281/zenodo.3606573*

Mirza, M. & Osindero, S. (2014), 'Conditional generative adversarial nets', *CoRR* **abs/1411.1784**.
URL: *http://arxiv.org/abs/1411.1784*

Mor, N., Wolf, L., Polyak, A. & Taigman, Y. (2018), 'A universal music translation network', *CoRR* **abs/1805.07848**.
URL: *http://arxiv.org/abs/1805.07848*

Muja, M. & Lowe, D. G. (2009), Fast approximate nearest neighbors with automatic algorithm configuration, *in* 'VISAPP International Conference on Computer Vision Theory and Applications', pp. 331–340.

Navas, E. (2010), *Regressive and Reflexive Mashups in Sampling Culture*, Springer Vienna, pp. 157–177.
URL: *https://doi.org/10.1007/978-3-7091-0096-7_10*

Odena, A., Dumoulin, V. & Olah, C. (2016), 'Deconvolution and checkerboard artifacts', *Distill* .
URL: *http://distill.pub/2016/deconv-checkerboard*

Odena, A., Olah, C. & Shlens, J. (2017), Conditional image synthesis with auxiliary classifier GANs, *in* D. Precup & Y. W. Teh, eds, 'Proceedings of the 34th International Conference on Machine Learning', Vol. 70 of *Proceedings of Machine Learning Research*, PMLR, International Convention Centre, Sydney, Australia, pp. 2642–2651.
URL: *http://proceedings.mlr.press/v70/odena17a.html*

O'Shaughnessy, D. (1987), *Speech communication: human and machine*, Addison-Wesley series in electrical engineering, Addison-Wesley Pub. Co.
URL: *https://books.google.no/books?id=mHFQAAAAMAAJ*

Oxenham (ed.), A. & Oxenham, A. (2005), *Pitch: Neural Coding and Perception*, Springer.

Oxford University Press (2020*a*), 'Oed online'.
URL: *https://www.oed.com/view/Entry/40839*

Oxford University Press (2020*b*), 'Oed online'.
URL: *https://www.oed.com/view/Entry/116237*

Pachet, F., Roy, P. & Cazaly, D. (1999), 'A combinatorial approach to content-based music selection', **1**, 457–462 vol.1.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. & Chintala, S. (2019), Pytorch: An imperative style, high-performance deep learning library, *in* H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox & R. Garnett, eds, 'Advances in Neural Information Processing Systems 32', Curran Associates, Inc., pp. 8024–8035.
URL: *http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf*

Perarnau, G., van de Weijer, J., Raducanu, B. & Álvarez, J. M. (2016), 'Invertible Conditional GANs for image editing', *CoRR* **abs/1611.06355**.
URL: *http://arxiv.org/abs/1611.06355*

Ping, W., Peng, K. & Chen, J. (2018), 'ClariNet: Parallel Wave Generation in End-to-End Text-to-Speech', *CoRR* **abs/1807.07281**.
URL: *http://arxiv.org/abs/1807.07281*

Prenger, R., Valle, R. & Catanzaro, B. (2018), 'WaveGlow: A Flow-based Generative Network for Speech Synthesis', *CoRR* **abs/1811.00002**.
URL: *http://arxiv.org/abs/1811.00002*

Purwins, H., Li, B., Virtanen, T., Schluter, J., Chang, S.-Y. & Sainath, T. (2019), 'Deep learning for audio signal processing', *IEEE Journal of Selected Topics in Signal Processing* **13**(2), 206–219.
URL: *http://dx.doi.org/10.1109/JSTSP.2019.2908700*

Radford, A., Metz, L. & Chintala, S. (2015), 'Unsupervised representation learning with deep convolutional generative adversarial networks', *CoRR* **abs/1511.06434**.

Raffel, C. (2016), Learning-Based Methods for Comparing Sequences, with Applications to Audio-to-MIDI Alignment and Matching, PhD thesis, Columbia University.

Reynolds, D. (2009), *Gaussian Mixture Models*, Springer US, Boston, MA, pp. 659–663.

Romney, J., Burg, J. & Schwartz, E. (2016), *Digital Sound & Music: Concepts, Applications, and Science*.

Ronneberger, O., Fischer, P. & Brox, T. (2015), 'U-net: Convolutional networks for biomedical image segmentation', *CoRR* **abs/1505.04597**.
URL: *http://arxiv.org/abs/1505.04597*

Salimans, T. & Kingma, D. P. (2016), 'Weight normalization: A simple reparameterization to accelerate training of deep neural networks', *CoRR* **abs/1602.07868**.
URL: *http://arxiv.org/abs/1602.07868*

Sampson, A. (2020), 'Chromaprint and Acoustid for Python'. Accessed: 2020-05-25.
URL: *https://github.com/beetbox/pyacoustid*

Scaringella, N., Zoia, G. & Mlynek, D. (2006), 'Automatic genre classification of music content: a survey', *Signal Processing Magazine, IEEE* **23**, 133 – 141.

Schörkhuber, C. & Klapuri, A. (2010), Constant-Q transform toolbox for music processing, *in* 'Proceedings of the 7th Sound and Music Computing Conference, Barcelona, Spain', pp. 3–6.

Seyerlehner, K., Widmer, G. & Knees, P. (2010), A comparison of human, automatic and collaborative music genre classification and user centric evaluation of genre classification systems, pp. 118–131.

Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerry-Ryan, R. J., Saurous, R. A., Agiomyrgiannakis, Y. & Wu, Y. (2017), 'Natural TTS Synthesis by Conditioning WaveNet on Mel Spectrogram Predictions', *CoRR* **abs/1712.05884**.
URL: *http://arxiv.org/abs/1712.05884*

Själander, M., Jahre, M., Tufte, G. & Reissmann, N. (2019), 'EPIC: An energy-efficient, high-performance GPGPU computing research infrastructure'.
URL: *https://arxiv.org/pdf/1912.05848.pdf*

SoundCloud (2007). Accessed: 2020-11-04.
URL: *https://soundcloud.com/*

Stevens, S. S. (1937), 'A Scale for the Measurement of the Psychological Magnitude Pitch', *Acoustical Society of America Journal* **8**, 185.

Taymans, W., Baker, S., Wingo, A., Bultje, R. S. & Kost, S. (2016), 'GStreamer Application Development Manual'.

Thiede, T., Treurniet, W., Bitto, R., Schmidmer, C., Sporer, T., Beerends, J., Colomes, C., Keyhl, M., Stoll, G., Brandenburg, K. & Feiten, B. (2000), 'PEAQ—The ITU Standard for Objective Measurement of Perceived Audio Quality', *Journal of the Audio Engineering Society* **48**, 3–29.

Toivonen, H. (2010), *Apriori Algorithm*, Springer US, Boston, MA, pp. 39–40.

Tribe of Noise (2020), 'Free Music Archive',
URL: *https://freemusicarchive.org/*. Accessed: 2020-11-04.

Ulyanov, D. & Lebedev, V. (2016), *Audio texture synthesis and style transfer*.
URL: *https://dmitryulyanov.github.io/audio-texture-synthesis-and-style-transfer/*

van den Oord, A., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A. W. & Kavukcuoglu, K. (2016), 'WaveNet: A Generative Model for Raw Audio', *CoRR* **abs/1609.03499**.
URL: *http://arxiv.org/abs/1609.03499*

van den Oord, A., Li, Y., Babuschkin, I., Simonyan, K., Vinyals, O., Kavukcuoglu, K., van den Driessche, G., Lockhart, E., Cobo, L. C., Stimberg, F., Casagrande, N., Grewe, D., Noury, S., Dieleman, S., Elsen, E., Kalchbrenner, N., Zen, H., Graves, A., King, H., Walters, T., Belov, D. & Hassabis, D. (2017), 'Parallel WaveNet: Fast High-Fidelity Speech Synthesis', *CoRR* **abs/1711.10433**.
URL: *http://arxiv.org/abs/1711.10433*

Vande Veire, L., De Bie, T. & Dambre, J. (2019), A CycleGAN for style transfer between drum and bass subgenres, *in* 'ML4MD at ICML2019, Proceedings', p. 3.
URL: *https://sites.google.com/view/ml4md2019/home*

Vasquez, S. & Lewis, M. (2019), 'MelNet: A Generative Model for Audio in the Frequency Domain', *CoRR* **abs/1906.01083**.
URL: *https://arxiv.org/pdf/1906.01083.pdf*

Verma, P. & III, J. O. S. (2018), 'Neural style transfer for audio spectograms', *CoRR* **abs/1801.01589**.
URL: *http://arxiv.org/abs/1801.01589*

Wang, Y., Skerry-Ryan, R. J., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., Le, Q. V., Agiomyrgiannakis, Y., Clark, R. & Saurous, R. A. (2017), 'Tacotron: A fully end-to-end text-to-speech synthesis model', *CoRR* **abs/1703.10135**.
URL: *http://arxiv.org/abs/1703.10135*

Weiss, P. & Taruskin, R. (2007), *Music in the Western World: A History in Documents*, p. 555.

Xu, M., Duan, L.-Y., Cai, J., Chia, L.-T., Xu, C. & Tian, Q. (2005), HMM-Based Audio Keyword Generation, *in* K. Aizawa, Y. Nakamura & S. Satoh, eds, 'Advances in Multimedia Information Processing - PCM 2004', Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 566–574.

Yang, S. & Chung, M. (2019), 'Self-imitating feedback generation using GAN for computer-assisted pronunciation training', *CoRR* **abs/1904.09407**.
URL: *http://arxiv.org/abs/1904.09407*

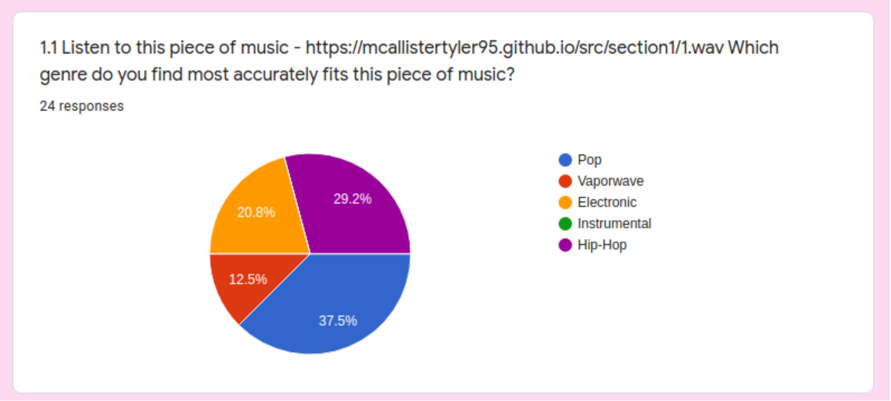
Youtube (2005). Accessed: 2020-11-04.
URL: *https://www.youtube.com/*

Zhang, R., Isola, P. & Efros, A. A. (2016), 'Colorful image colorization', *CoRR* **abs/1603.08511**.
URL: *http://arxiv.org/abs/1603.08511*

Zhou, T., Krähenbühl, P., Aubry, M., Huang, Q. & Efros, A. A. (2016), 'Learning dense correspondence via 3d-guided cycle consistency', *CoRR* **abs/1604.05383**.
URL: *http://arxiv.org/abs/1604.05383*

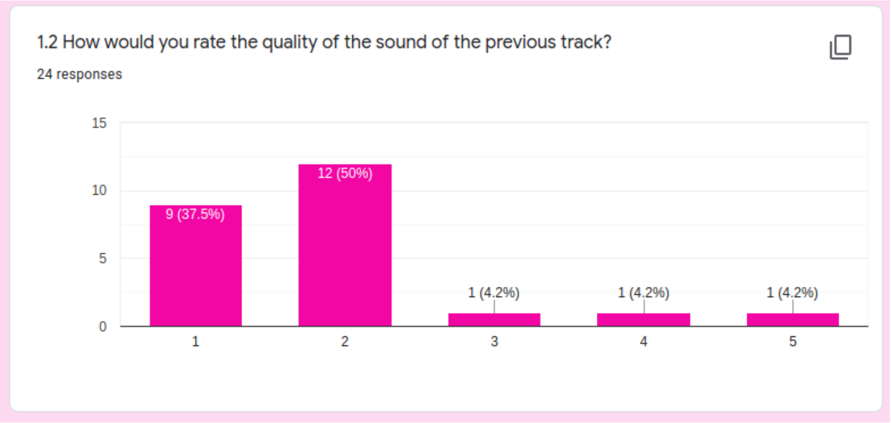
Zhu, J., Park, T., Isola, P. & Efros, A. A. (2017), 'Unpaired image-to-image translation using cycle-consistent adversarial networks', *CoRR* **abs/1703.10593**.
URL: *http://arxiv.org/abs/1703.10593*

Appendix A: Survey Results



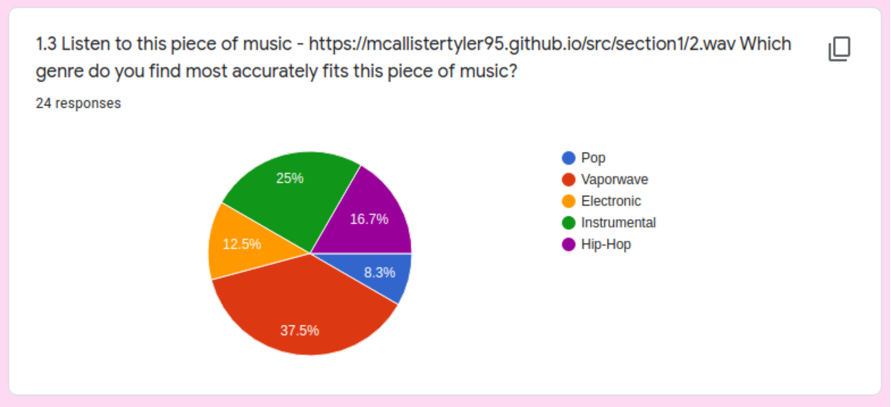
1.1 Listen to this piece of music - https://mcallistertyler95.github.io/src/section1/1.wav Which genre do you find most accurately fits this piece of music?

24 responses



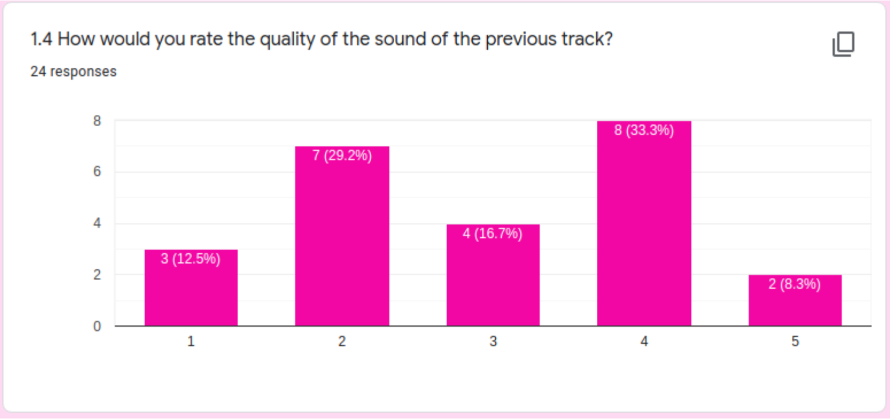
1.2 How would you rate the quality of the sound of the previous track?

24 responses



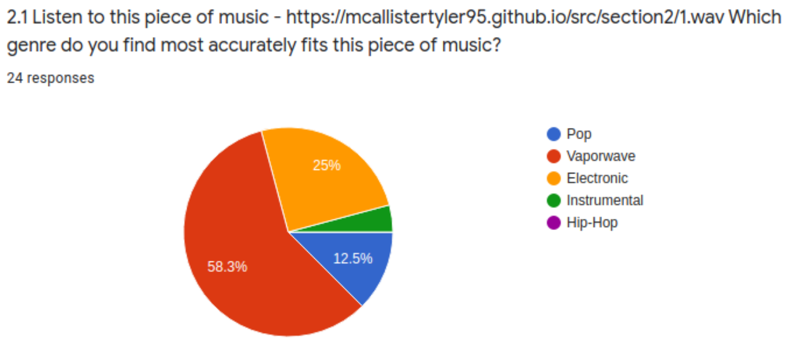
1.3 Listen to this piece of music - https://mcallistertyler95.github.io/src/section1/2.wav Which genre do you find most accurately fits this piece of music?

24 responses



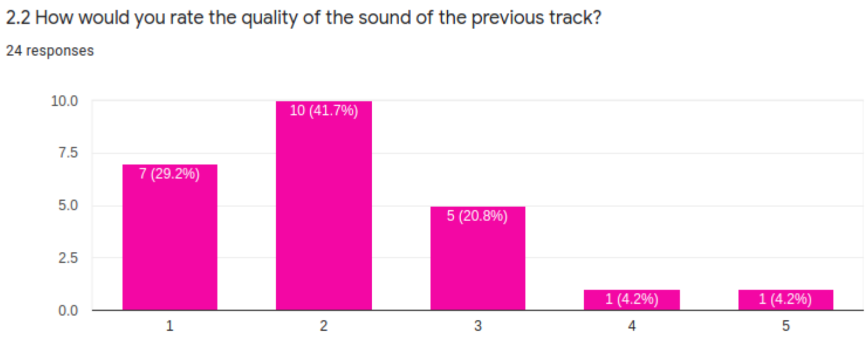
1.4 How would you rate the quality of the sound of the previous track?

24 responses



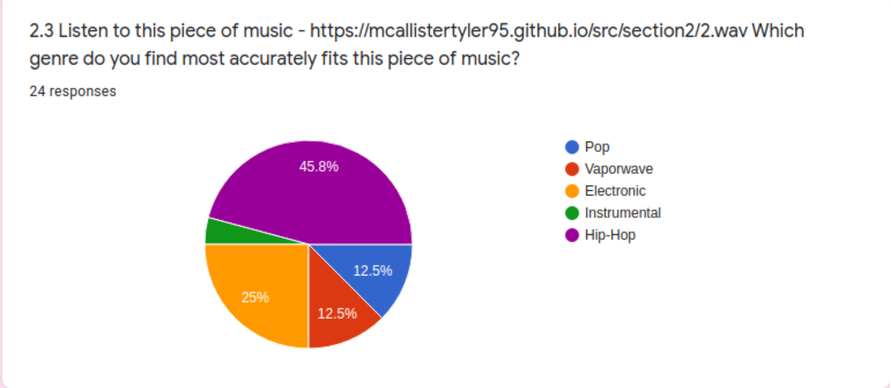
2.1 Listen to this piece of music - https://mcallistertyler95.github.io/src/section2/1.wav Which genre do you find most accurately fits this piece of music?

24 responses



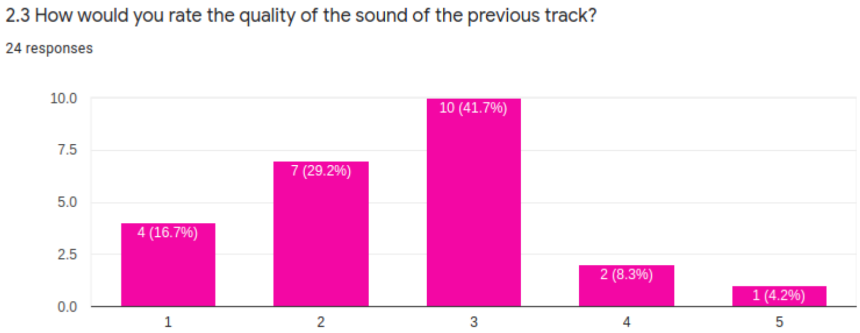
2.2 How would you rate the quality of the sound of the previous track?

24 responses



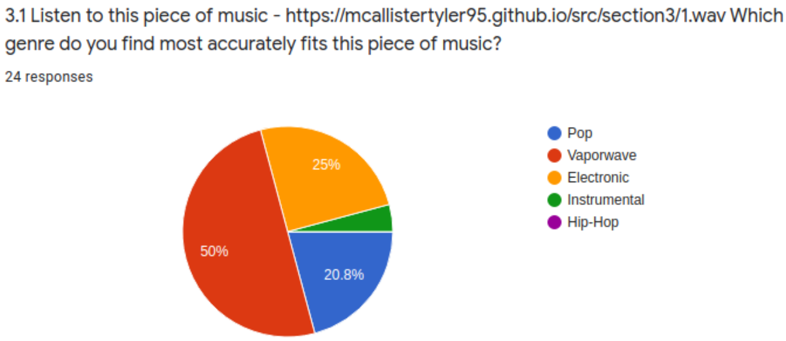
2.3 Listen to this piece of music - https://mcallistertyler95.github.io/src/section2/2.wav Which genre do you find most accurately fits this piece of music?

24 responses



2.3 How would you rate the quality of the sound of the previous track?

24 responses

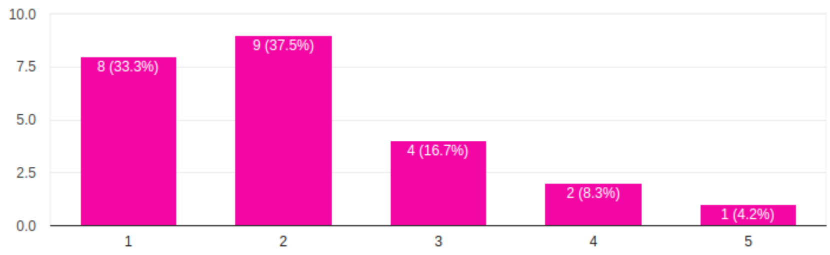


3.1 Listen to this piece of music - https://mcallistertyler95.github.io/src/section3/1.wav Which genre do you find most accurately fits this piece of music?

24 responses

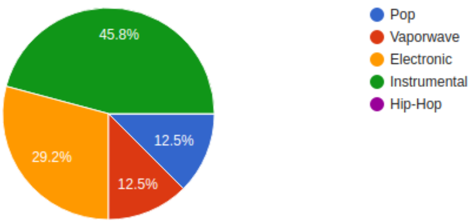
3.2 How would you rate the quality of the sound of the previous track?

24 responses



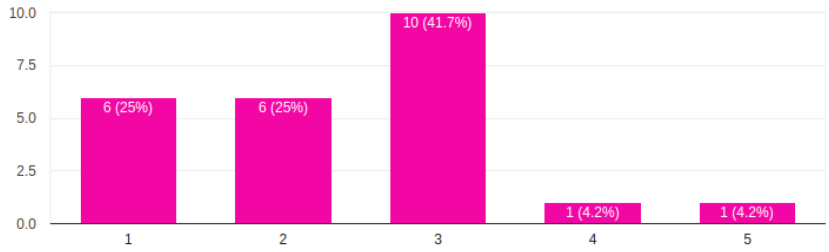
3.3 Listen to this piece of music - https://mcallistertyler95.github.io/src/section3/2.wav - Which genre do you find most accurately fits this piece of music?

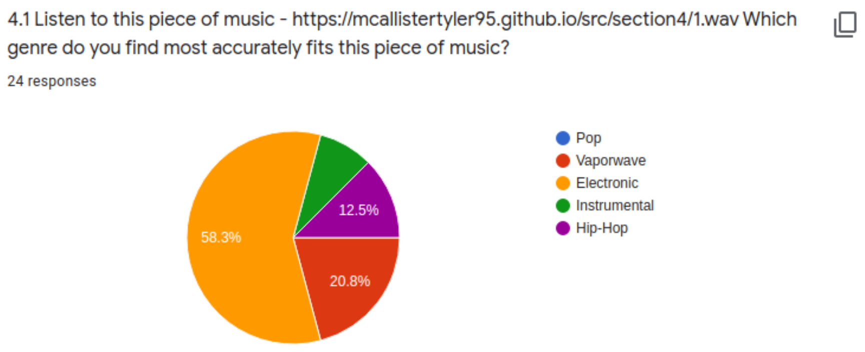
24 responses



3.4 How would you rate the quality of the sound of the previous track?

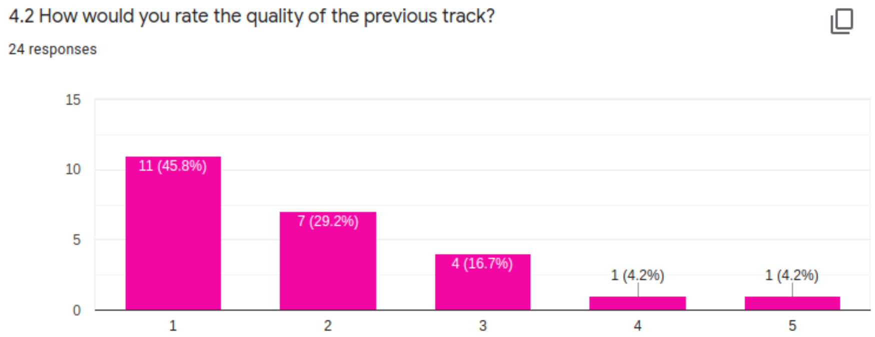
24 responses





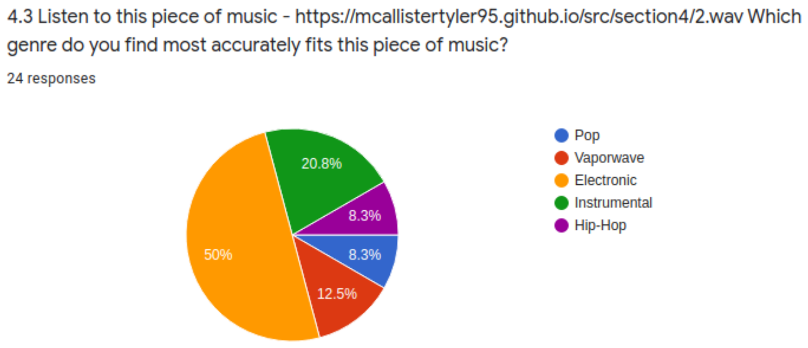
4.1 Listen to this piece of music - https://mcallistertyler95.github.io/src/section4/1.wav Which genre do you find most accurately fits this piece of music?

24 responses



4.2 How would you rate the quality of the previous track?

24 responses

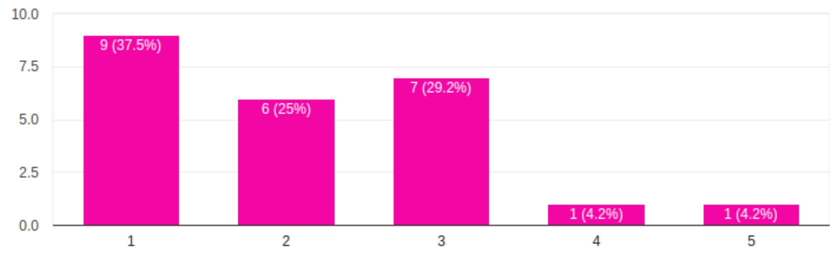


4.3 Listen to this piece of music - https://mcallistertyler95.github.io/src/section4/2.wav Which genre do you find most accurately fits this piece of music?

24 responses

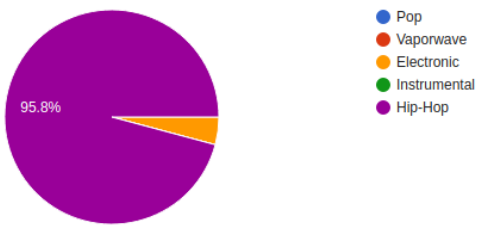
4.4 How would you rate the quality of the sound of the previous track?

24 responses



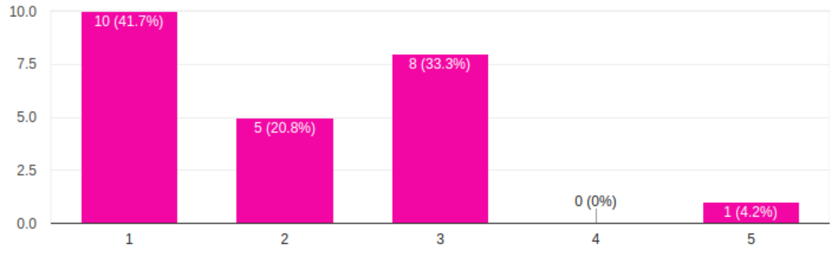
5.1 Listen to this piece of music -https://mcallistertyler95.github.io/src/section5/1.wav Which genre do you find most accurately fits this piece of music?

24 responses



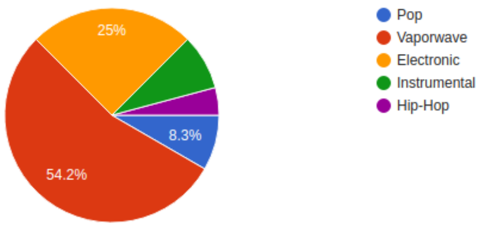
5.2 How would you rate the quality of the sound of the previous track?

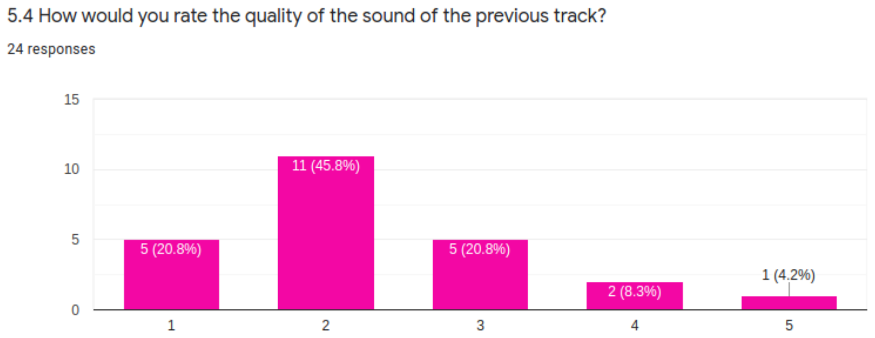
24 responses



5.3 Listen to this piece of music - https://mcallistertyler95.github.io/src/section5/2.wav Which genre do you find most accurately fits this piece of music?

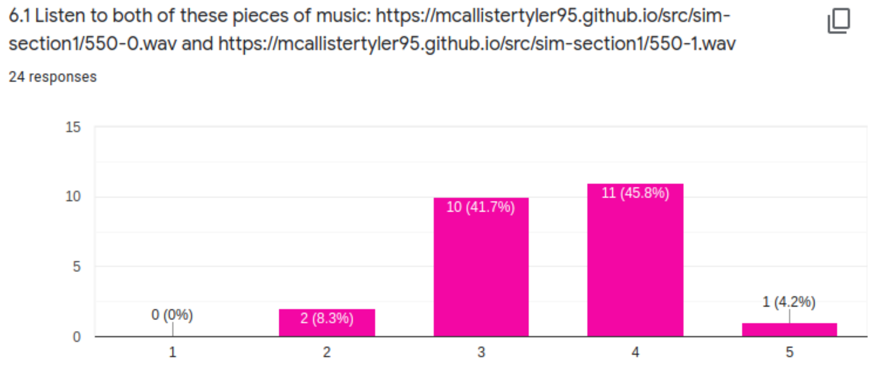
24 responses





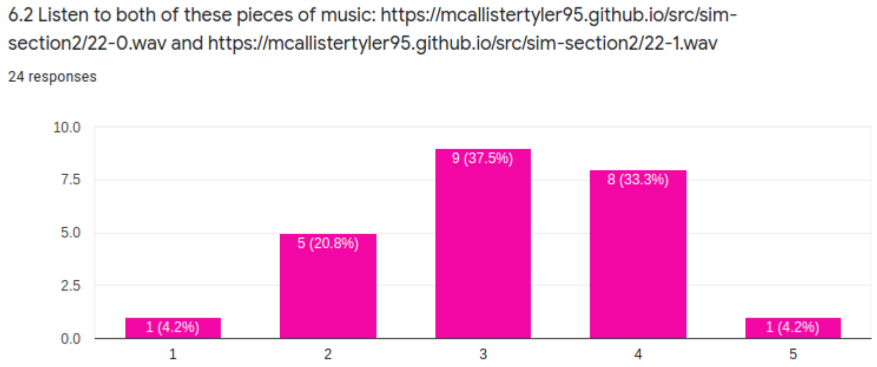
5.4 How would you rate the quality of the sound of the previous track?

24 responses



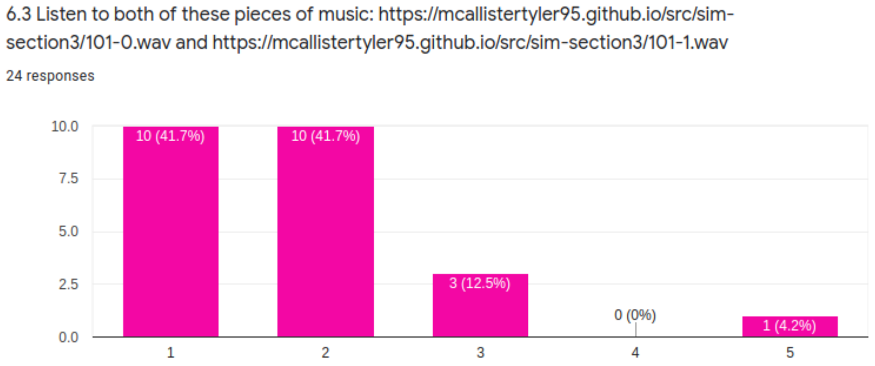
6.1 Listen to both of these pieces of music: https://mcallistertyler95.github.io/src/sim-section1/550-0.wav and https://mcallistertyler95.github.io/src/sim-section1/550-1.wav

24 responses



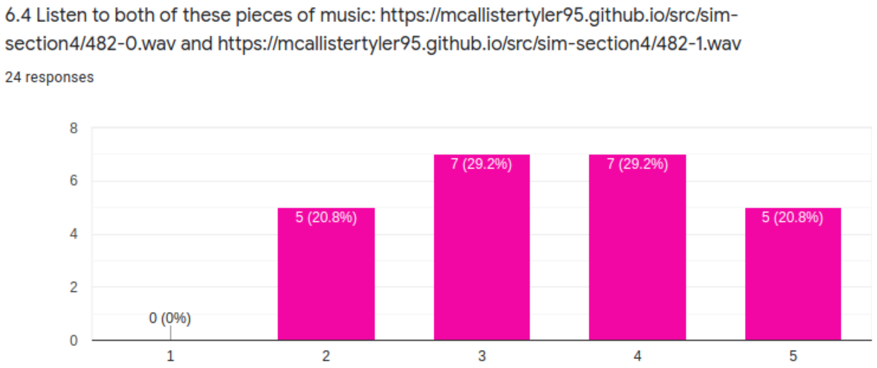
6.2 Listen to both of these pieces of music: https://mcallistertyler95.github.io/src/sim-section2/22-0.wav and https://mcallistertyler95.github.io/src/sim-section2/22-1.wav

24 responses



6.3 Listen to both of these pieces of music: https://mcallistertyler95.github.io/src/sim-section3/101-0.wav and https://mcallistertyler95.github.io/src/sim-section3/101-1.wav

24 responses



6.4 Listen to both of these pieces of music: https://mcallistertyler95.github.io/src/sim-section4/482-0.wav and https://mcallistertyler95.github.io/src/sim-section4/482-1.wav

24 responses