UNIVERSITY OF
CAMBRIDGE

Recognising, generating, and interpolating timbre in electric guitars with semantic descriptors

Joseph Manfredi Cameron

Homerton College

Submitted in partial fulfillment of the requirements for the Master of Philosophy in Advanced Computer Science

Declaration

I, Joseph Manfredi Cameron of Homerton College, being a candidate for the Master of Philosophy in Advanced Computer Science, hereby declare that this project report and the work described in it are my own work, unaided except as may be specified below, and that the project report does not contain material that has already been used to any substantial extent for a comparable purpose. In preparation of this project report I did not use text from AI-assisted platforms generating natural language answers to user queries, including but not limited to ChatGPT. I am content for my project report to be made available to the students and staff of the University.

Signed: Joseph Manfredi Cameron

Date: June 03, 2024

Word count

Total page count: 146

Main chapters (excluding front-matter, references and appendix): 76 pages (pp 15–90)

Main chapters word count: 14949

Methodology used to generate that word count:

Open the terminal and type the following prompt:

```
$ texcount -merge -sum -q -1 thesis.tex
14949
```

Abstract

Recognising, generating, and interpolating timbre in electric guitars with semantic descriptors

Joseph Manfredi Cameron

This dissertation explores the application of deep learning techniques to recognise, generate, and interpolate the timbre of monophonic electric guitar sounds using semantic descriptors. The research focuses on developing a comprehensive understanding of timbral nuances within electric guitar sounds, which are described through a set of twenty distinct timbre descriptors, each represented at varying magnitudes of presence within the sounds. A key innovation of this work is the creation of the SemanticTimbreDataset, a novel dataset of monophonic electric guitar sounds annotated with these timbre descriptors, facilitating the training and evaluation of specialised neural network models. Convolutional neural networks (CNNs) were developed to recognise specific timbral characteristics from the dataset, proving effective in identifying subtle differences in timbre based on semantic descriptors. This enhances understanding of the complex nature of timbre and has practical implications. Furthermore, a variational autoencoder (VAE) was implemented to generate new sounds exhibiting specified timbral qualities described by the semantic descriptors and to interpolate between these characteristics, blending descriptors like 'bright' and 'dark' across different magnitudes. This capability introduces new potential for creative sound synthesis and practical sound design in music production, offering a new dimension to the field. This dissertation also provides a comprehensive evaluation of these models, demonstrating their effectiveness in a controlled, monophonic context and confirming their potential for broader applications in digital audio processing. The project's focused approach on a single instrument, the electric guitar and its effects units, addresses a gap in current research, which often spans multiple instruments without a deep dive into the intricate timbral variations of each. This work underscores the potential of deep learning to enhance the expressive capabilities of musical instruments. The findings have significant implications for music technology, audio engineering, and digital sound design, providing a foundation for more nuanced and expressive tools in digital music production through natural language.

Acknowledgements

Firstly, I want to thank my supervisors Professor Alan F. Blackwell and Dr Peter Harrison for their resolute support and guidance throughout this project. Your continual advice and feedback throughout was invaluable, and I am grateful and proud to have taken this journey of discovery with both of you.

Thank you to my parents Denis Lyons Cameron and Maria Manfredi Cameron for instilling a sense of curiosity, optimism, adventure, and everlasting wonder into my world. Both of you nurtured me into the person I am today, and I will forever be most grateful for your unconditional love and joy.

I also want to thank all of my dear friends and colleagues that I have met here in Cambridge and elsewhere around the globe. Thank you for all the wonderful moments and memories throughout a spectacular and magnificent year. You know who you are, and I want to thank each and every one of you.

Contents

Chapter 1

Introduction

Considering two non-identical sounds of the same pitch and loudness, timbre describes all the other differing tonal qualities between them [170]. Many mechanisms of human sound perception, culture, and individual preferences influence our holistic perception of timbre, making it challenging to measure and explain [126, 124, 106]. Yet, timbre is one of music's most important properties that musicians and producers can use to express emotion and creativity [116, 71]. Understanding timbre and its many facets is crucial for mixing different instruments and sounds or designing individual sounds in production or live performance [123, 37, 175, 47, 6].

Fritz et al. demonstrate that a violin's timbre can be described using descriptors such as 'bright' or 'warm'; furthermore, they successfully correlated these descriptors with specific adjustments to the frequency spectrum of sound that violins produce [50]. With inspiration from Fritz et al., this project researches how humans perceive timbre within the context of communicating descriptors for sounds produced by the electric guitar and its effects units to incorporate an enhanced cultural understanding of timbre into artificial intelligence technology for recognising and generating timbre.

When exploring timbre within musical contexts, the electric guitar stands out as an iconic instrument with a significant impact on modern music. Renowned for its versatility and expressive power, the electric guitar has shaped various music genres and fostered a rich culture of sound exploration [24]. This dissertation focuses on the electric guitar as a primary source for analysing and synthesising timbral characteristics, driven by its unique sound and the extensive palette of timbral modifications made possible through electronic effects units such as guitar pedals and amplifiers [13, 114].

However, timbre manipulation remains primarily a manual process. Current digital music production tools such as equalisers, compressors, and specialised plugins provide control over and understanding of timbre, but they require significant expertise and are often limited by the presets or the algorithms they employ [174]. This project is motivated by the potential to fundamentally enhance how musicians and producers interact with

sound. By leveraging semantic descriptors such as 'bright' or 'dark', intuitive and more nuanced timbre control enabled by artificial intelligence could revolutionise audio tools. This dissertation focuses on deep learning-based approaches to recognising and generating timbral characteristics in electric guitar sounds via semantic descriptors, aiming to bridge the gap between recent technological advancements, musical expression, and human perception.

1.1 Research contributions

The first novel contribution provided by this research is the creation of a new dataset containing monophonic electric guitar sounds that exhibit timbral characteristics described by a guitar-relevant set of twenty timbre descriptors at various timbre magnitudes (i.e. how 'bright' is a given sound?). Chapter 3 details the creation of this new dataset called the SemanticTimbreDataset.

The second novel contribution of this project is the implementation of convolutional neural networks that can recognise timbral characteristics in monophonic electric guitar notes described by each of the SemanticTimbreDataset's twenty timbre descriptors. Chapter 4 details this timbre recognition system, and Chapter 6 comprehensively evaluates it.

This project's third novel contribution is the implementation of a variational autoencoder that can generate monophonic guitar notes at various pitches with timbral characteristics that can be described by the SemanticTimbreDataset's twenty timbre descriptors at varying timbre magnitudes. Chapter 5 details this timbre generation system, and Chapter 6 comprehensively evaluates it.

A profound aspect of this dissertation is the exploration of timbre interpolation through the timbre generation system. This approach represents the fourth novel contribution of the project, focusing on the generation of monophonic electric guitar notes that seamlessly blend and merge timbral characteristics derived from multiple descriptors. Manipulating the latent space of a variational autoencoder where these timbral characteristics reside enables smooth transitions between different timbre descriptors, such as 'bright' and 'dark', along with their respective degrees of presence (timbre magnitudes). This timbre interpolation method highlights the potential of deep learning in musical applications and sets the stage for further innovations in timbre/audio generation. Chapter 5 details timbre interpolation, and Chapter 6 provides its evaluation.

1.2 Project scope

The scope of this research was deliberately focused to ensure depth, precision, and relevance in study outcomes leading to meaningful, applicable results. Given the complexity of timbre as a subject and the vast possibilities within audio processing, setting clear boundaries was crucial for the successful execution and management of this project. The following points portray this project's scope:

- **Use of English Timbre Descriptors**: This project exclusively used English-language descriptors to annotate and describe timbral characteristics in the dataset, ensuring consistency in semantic labelling and avoiding the complexities associated with multilingual data processing. Focusing on English, a widely used language in music production and academic research, makes the findings accessible to a broad audience and relevant to global music technology contexts.

- **Use of Monophonic Electric Guitar Sounds**: Audio data for training deep learning models consisted solely of monophonic sounds, i.e., single-note recordings from electric guitars. Monophonic sounds provide a clearer basis for studying the fundamental aspects of timbre manipulation and recognition without the confounding effects of harmonic interactions.

- **Use of Electric Guitar Effects Units**: Timbre modifications in the dataset were achieved using electric guitar effects units, specifically guitar pedals and amplifiers. These devices are pivotal in shaping the electric guitar's sound and are extensively used in practice [13, 114]. By limiting timbre modification to these tools, the project aligns closely with real-world applications and standard practices in music production.

Chapter 2

Literature review

This chapter introduces essential concepts for understanding various aspects of this dissertation. First, the concept of timbre is formally defined and discussed within this project's scope involving electric guitar sounds, along with a brief exploration of music production tools that can be used to create and modify timbre. Then, the necessary background and relevant literature discussing deep learning techniques with models such as variational autoencoders and convolutional neural networks are introduced. Furthermore, an overview of how these machine learning techniques can be applied to the audio domain is provided. Finally, a review of previous literature that explores the classification, generation, and modification of timbre is presented.

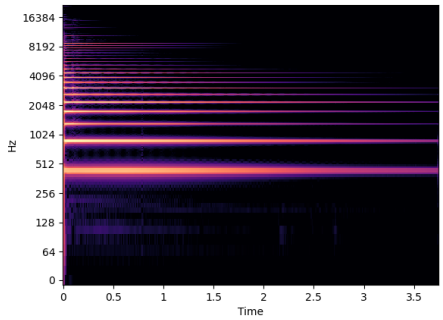
2.1 Timbre

Timbre is an important, distinctive, and defining property of sound in both musical and non–musical terms [171, 38]. Consequently, there is an extensive, rich history of research investigating timbre and its many components [170, 126, 124, 158].

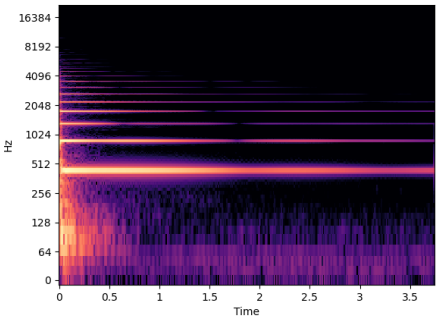
2.1.1 The definition of timbre

Considering two non-identical sounds with the same pitch and intensity, timbre describes all the other differing tonal qualities between the two sounds [170]. Humans can distinguish sounds of the same intensity and pitch emanating from different musical instruments, such as guitars and pianos, simply due to their timbre [66]. Figure 2.1 shows two spectrograms of two musical notes played at the same pitch of A4 [147] and perceived loudness of -22.5 LUFs, but one note is played on a Fender Stratocaster electric guitar and the other on a Steinway grand piano.

The qualities that differentiate these two notes constitute timbre, which is why timbre is sometimes referred to as the 'quality' of a sound [124, 157]. On a similar note, timbre



(a) Spectrogram of an A4 440Hz note played from a Fender Stratocaster



(b) Spectrogram of an A4 440Hz note played from a Steinway grand piano

Figure 2.1

can also be referred to as the 'colour' of sound because many people perceive different types of timbre as different colours [124, 161, 52, 53].

Even though timbre is strictly defined as the quality of a sound excluding pitch and intensity, aspects of pitch and intensity can be changed over short periods of time, or 'oscillated', to change the overall timbre of sounds [45, 124, 53, 157, 161]. Quick oscillations of pitch and intensity within sounds can modify timbre because these quick oscillations do not affect the fundamental perceived frequency or the fundamental perceived intensity of a sound [45, 124]. When a sound's fundamental frequency (pitch) or its fundamental perceived intensity (loudness) is altered, these changes are no longer related to timbre. Understanding this distinction between fundamental and non-fundamental pitch and intensity is crucial when judging what types of sound modification involve timbre.

2.1.2 The perception of timbre

The perception of timbre is influenced by both the physical properties of the sound and the listener's auditory processing mechanisms [124]. Factors of sound's physical properties that influence timbre perception can be separated into two broad categories: a sound's *spectral content* and *temporal characteristics* [65, 125].

- **The Spectral Content of Sound:** The presence of harmonics, their amplitudes, and their distribution play a crucial role in timbre perception. Sounds with similar spectral envelopes are often perceived as having similar timbre. Looking at Figures 2.1a and 2.1b, it is clear the electric guitar contains more harmonics than the piano in the 2kHz to 8kHz frequency range, whereas the piano contains more spectral information in the lower 50Hz to 400Hz frequency range. These spectral differences play a significant role in people perceiving them as two different instruments.
- **The Temporal Characteristics of Sound:** How a sound's amplitude evolves over

time, including its attack, decay, sustain, and release phases (see the ADSR envelope in Figure 2.2), affects timbre perception [17]. For instance, a short or 'sharp' attack time might be associated with percussive sounds, while a longer, more gradual attack time might suggest bowed strings [17, 53].

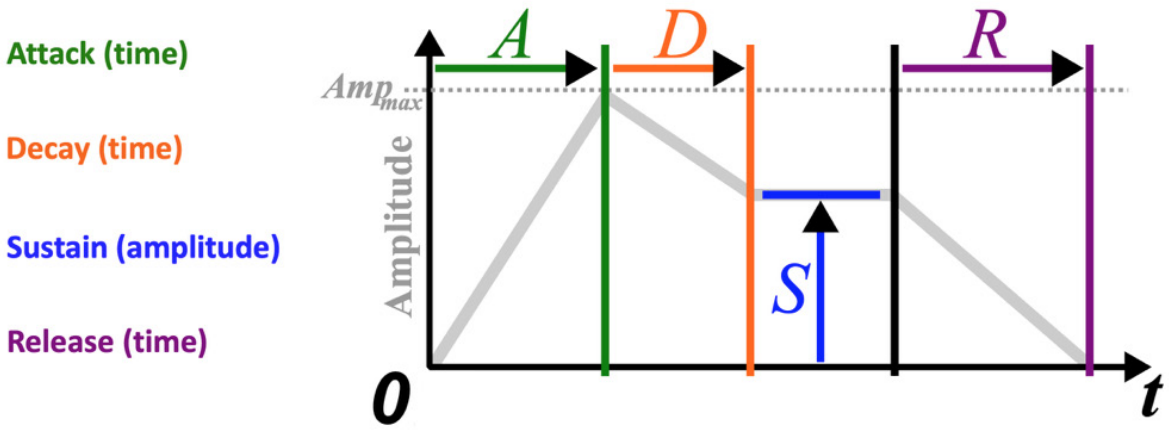


Figure 2.2: The ADSR envelope [65, 17, 51, 125, 82]. Figure from [65].

2.1.3 The semantics of timbre

Timbre's perceptual qualities are typically conveyed using a vocabulary describing physical characteristics and subjective impressions of sounds [158]. Common terms include 'bright', 'dark', 'soft', 'rich', and 'thin' [53]. These descriptors often derive from a listener's subjective experience and cultural associations with sounds [106]. Efforts have been made to categorise and standardise timbre descriptors to facilitate clearer communication and viable research. For example, researchers have used multidimensional scaling and cluster analysis to group similar timbral qualities, aiming to create a more systematic vocabulary for timbre [50, 215, 214, 213, 216].

Multiple methodologies for studying timbre and its semantics have been devised, including lexical studies that analyse the language and descriptors people use to describe timbre. This type of lexical analysis is undertaken to understand the semantic space of timbre and how different sounds are perceptually related according to human listeners. Fritz et al. employed this methodology to demonstrate that violin timbre can be described using descriptors, and they correlated these descriptors with specific adjustments to violin frequency spectra [50]. Another popular method is the semantic differential scale method, which asks participants to rate sounds using a set of continuous or discrete bipolar scales (e.g., bright vs. dark) [201, 91, 35]. The results of this can help quantify how sounds are perceived on different semantic dimensions, providing a structured way to map the semantic attributes of timbre. Reymore and Huron used semantic differential scales with

ranges from 1-7 to gauge participants' opinions on how well multiple semantic terms described the timbre of given sounds [154]. Reymore et al. also used semantic scales that investigated semantic timbre perception vs pitch register across musical notes played by various orchestral instruments, where each scale asked participants to rate how well one semantic term described a particular sound from 1-5 [155]. This project extends these previous studies by additionally using machine learning models to predict values on semantic differential scales for specific timbre descriptors from acoustic features and then generate acoustic features that incorporate timbral characteristics derived from those semantic descriptors and their semantic differential scales.

2.1.4 Exploring the timbre of electric guitars

The timbre of an electric guitar is distinct and versatile, shaped significantly by its physical characteristics and the electronic effects applied to it [162, 195, 13]. Electric guitar sounds are often described using various terms that reflect their rich harmonic content and dynamic expression [195, 47, 84]. Common descriptors include: '*bright*', which describes a sound that emphasises higher frequencies [163, 84]; '*dark*', which describes a sound that often emphasises lower to mid frequencies [195, 84]; and '*tinny*', which describes a thin, metallic sound lacking in bass [195]. These terms help musicians, engineers, and enthusiasts communicate specific tonal qualities and preferences essential for both live performances and studio recordings.

2.1.4.1 Guitar effects units

Guitar effects units, which commonly come in the form of pedals [114, 28, 78, 8, 5], play a pivotal role in defining and modifying the timbre of an electric guitar [13]. They allow guitarists to alter their sound dramatically, leading to new forms of musical expression and innovation. Two major categories of audio effects [209] that are particularly influential in shaping an electric guitar's timbre are **distortion effects** [67] and **modulation effects** [13, 89].

Distortion effects are crucial for many genres, particularly rock and metal [67]. They add harmonic complexity, sustain, and a gritty texture that can make the guitar stand out in a mix and sound more aggressive and powerful. There are multiple types of distortion, including 'crunchy', 'crushed', and 'fuzzy' distortion [163, 175]. 'Fuzz' is a type of extreme distortion that creates a thick, compressed, and 'fuzzy' sound and is notable for its use in psychedelic rock, garage, and modern indie music [67, 42]. Renowned guitarist Keith Richards famously used a Maestro Fuzz-Tone pedal to create fuzz distortion in 'Satisfaction' by the Rolling Stones [114]. Referring to the two main perceptual categories of timbre identified in Section 2.1.2, distortion effects modify sounds concerning the spectral timbre

category.

Modulation effects, which will hereby be called **oscillation effects** due to their effect on temporal timbre, modulate the frequencies and amplitudes of sounds at various temporal rates, which can alter the timbre of sounds without altering their perceived pitch or intensity [13, 89, 157, 161, 52, 53] (see Section 2.1.1). A famous example of an oscillation effect is 'WahWah', which modulates the frequencies of guitars [161] using a foot-controlled expression pedal [48, 163]. It creates a vocal-like sound, where the tone can shift from a muted, low-frequency sound to a sharp, bright sound, mimicking the human voice saying 'wah' [3, 103, 86, 157]. Influential guitarist Jimi Hendrix highlighted the 'WahWah' effect in his playing style regularly, elevating its popularity [107, 48, 3]. 'Shimmer' is another popular temporal-based oscillation effect often used in ambient and experimental music [204, 75]. 'Shimmer' combines reverb with pitch shifting and feedback to create a lush, ethereal sound that adds a spacious, celestial quality to the guitar [207, 217, 206].

2.1.5 Virtual Studio Technology (VST) for timbre modification

Virtual Studio Technology (VST) is a software interface standard that integrates audio synthesiser and effect plugins with audio editors and recording systems such as digital audio workstations (DAWs) like Logic Pro [185, 181, 81]. Developed by Steinberg in 1996 [185, 180], VST technology supports plugins that mimic various audio effects and virtual instruments [119], enabling users to add audio effects to sounds in their DAWs. These plugins include software emulations of the guitar effects units mentioned in Section 2.1.4.1. A well-known example of a guitar-FX VST plugin is Guitar Rig developed by Native Instruments [83], and an example of its interface is displayed in Figure 2.3.

To perform semantic timbre recognition and generation of guitar sounds with deep learning models as desired in this project, a large dataset of guitar sounds with diverse timbre variations must be created. VST provides an essential toolkit for achieving this. Using VST plugins like Guitar Rig [83], specific timbral characteristics of guitar sounds can be altered systematically via specified control parameters to create the required audio samples.

2.2 Relevant background in deep learning

Deep learning is a field within machine learning where neural networks with many layers of neurons (hence the term 'deep' learning) are employed to model complex patterns in data [109, 56]. By utilising deep learning, this project aims to advance the technological aspects of analysing, recognising, and generating timbre and contribute to the artistic processes of sound design and music creation, offering new tools that allow for more expressive and innovative sound production through natural language.

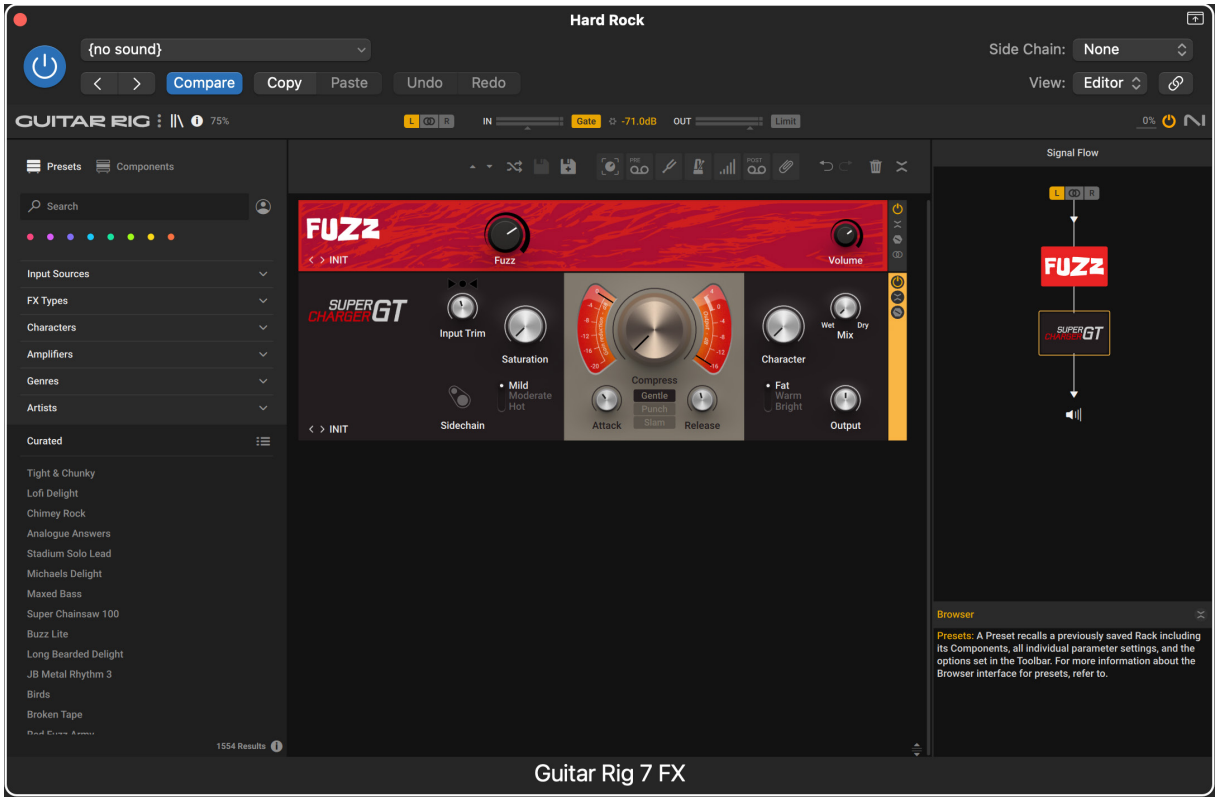


Figure 2.3: Screenshot of the Guitar Rig Pro 7 VST plugin.

2.2.1 Convolutional neural networks

Convolutional neural networks (CNNs) [108] have revolutionised the field of image processing and analysis by efficiently learning hierarchical representations [104]. Each layer of a CNN transforms one volume of activations to another through a differentiable function, using a set of learnable filters that each produce a feature map. This process allows CNNs to automatically learn and detect crucial patterns for classification or regression tasks [55].

2.2.2 Variational autoencoders

Variational autoencoders (VAEs) are a class of deep generative models that are particularly effective in learning latent representations of input data, enabling the generation of new data instances that resemble the original data [100, 101]. To fully grasp the theoretical underpinnings and motivation for VAEs, it is first essential to understand the standard autoencoder.

2.2.2.1 Autoencoders

Autoencoders are neural networks used to learn efficient encodings of unlabeled data. The typical autoencoder architecture is designed to compress (**encode**) the input data into a lower-dimensional, **latent** representation, and then from this latent representation

reconstruct (**decode**) an output to closely match the original input [54]. The encoding and decoding process is facilitated through two main components: the **encoder** and the **decoder**.

The encoder transforms input data into smaller, dense latent representations, which comprise the autoencoder's **latent space** or bottleneck. This step effectively reduces the data's dimensionality, capturing only the most important features. The decoder takes the encoded data from the latent space and reconstructs the input data as accurately as possible. This step is crucial for learning the distribution of the input data.

Autoencoders are trained to minimise the reconstruction loss, typically measured as the mean squared error (MSE) between the input and the reconstructed output. They are widely used for dimensionality reduction, feature extraction, and unsupervised learning in computer vision.

2.2.2.2 From standard autoencoders to variational autoencoders

VAEs extend the idea of standard autoencoders by producing a probabilistic latent space where they learn to encode inputs as distributions over the latent space rather than fixed points [100]. This key characteristic of VAEs provides two significant advantages over standard autoencoders. Firstly, by treating the latent space as a distribution, VAEs facilitate the generation of new data points that are variations of the learned examples by manipulating latent representations. This feature enables interpolation between latent samples [177] and organised exploration of the latent space [156, 9, 136, 31]. Secondly, the probabilistic nature of a VAE's latent space ensures that similar points are mapped closely, creating a smooth gradient of change in data characteristics [100], further bolstering interpolation tasks [31].

One of the notable differences between a standard autoencoder and a VAE lies in the encoder's output. Instead of outputting a fixed vector like an autoencoder, a VAE's encoder outputs parameter vectors of a probability distribution, typically the mean ($\boldsymbol{\mu}$) and variance ($\boldsymbol{\sigma^2}$) of a Gaussian distribution. During training, a sample from this distribution is drawn and passed to the decoder, which then attempts to reconstruct the input data. However, the reparameterization trick must be employed during the training phase of a VAE to sample a point from the Gaussian distribution and reconstruct the input because sampling from a distribution is inherently stochastic [100, 102]. The reparameterization trick can be defined as: $\boldsymbol{z = \mu + \epsilon \odot \sigma}$ where $\boldsymbol{\epsilon \sim \mathcal{N}(0, I)}$.

Another difference between autoencoders and VAEs lies in their respective loss functions. The loss function in VAEs comprises two terms:

- **Reconstruction Loss**: This is similar to the loss function in standard autoencoders, encouraging the decoded samples to match the initial inputs. It can be represented as: $\boldsymbol{L_{Reconstruction} = \mathbb{E}_{q(z|x)}[\log p(x|z)]}$

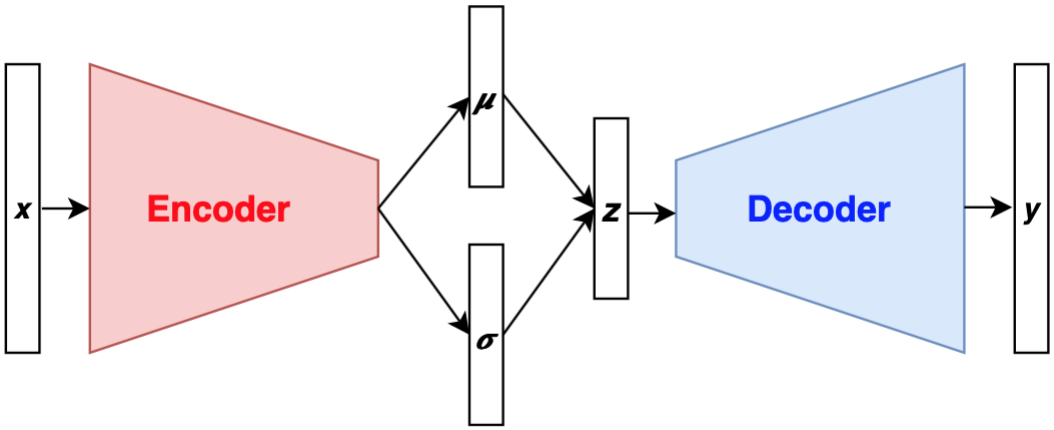


Figure 2.4: General VAE architecture.

- **Kullback-Leibler (KL) Divergence**: This term acts as a regularizer and is the Kullback-Leibler divergence between the encoded distribution and a prior distribution, typically a standard normal distribution. This divergence measures how much information is lost when using the encoded distribution to represent the prior distribution, promoting the encoded data to follow a known distribution, which aids in generating new data. It can be represented as: $\boldsymbol{L_{KL}} = \boldsymbol{KL}[\boldsymbol{q}(\boldsymbol{z}|\boldsymbol{x})||\boldsymbol{p}(\boldsymbol{z})] = \frac{1}{2}\sum_k(\boldsymbol{\sigma_k^2} + \boldsymbol{\mu_k^2} - 1 - \log(\boldsymbol{\sigma_k^2}))$

These two terms combine into the following total loss function which VAEs minimise during training: $\boldsymbol{L_{VAE}} = \boldsymbol{L_{Reconstruction}} + \boldsymbol{\beta} \cdot \boldsymbol{L_{KL}}$ where $\boldsymbol{\beta}$ is a hyperparameter that balances the two loss terms and provides a trade-off between their respective purposes.

2.3 Deep learning in the audio domain

The deep learning techniques mentioned in Section 2.2 have been primarily motivated to tackle problems in computer vision [151, 69]. When applying these techniques to tackle problems in the audio processing domain, a widely adopted process that has proven successful is to convert audio files into spectrogram images and then use this data with the previously mentioned deep learning models in a similar way to typical computer vision tasks [151, 69].

Spectrograms, like those in Figure 2.1, are fundamental tools in audio signal processing that provide visual representations of the frequency spectrum in a sound signal as it varies over time. They are created via the Short-Time Fourier Transform [62]. Spectrograms provide a rich source of features for training deep learning algorithms, and these features can be used to identify and classify different sounds or analyse timbre for timbre recognition.

Also, spectrograms offer methods to manipulate sound in a detailed and precise manner. Changes made in the frequency domain can be transformed back into the time domain, allowing for practical implementations of sound modifications, such as timbre modifications for timbre generation and interpolation.

2.3.1 Deep learning for timbre classification & regression

In the context of deep learning, timbre classification involves categorising sounds into predefined groups based on their timbral qualities, while timbre regression pertains to predicting continuous parameters that describe these qualities, such as 'brightness' or 'richness'.

Hernandez-Olivan and Beltran presented a pitch-invariant neural network that can classify musical sounds into 20 instrument classes by instrument-specific timbre [68]. Pons et al. used multiple CNN architectures to model timbre and classify sounds in the IRMAS dataset [12] by instrument type using magnitude spectrograms as inputs [148]. Pons et al. also recommend that a model designed to learn timbre should be pitch and loudness invariant [148]. Blaszke and Kostek used a set of CNNs to identify numerous instruments present in an audio file from timbre analysis on Mel-frequency cepstral coefficients (MFCCs), where each CNN was trained to recognise timbral characteristics for a single instrument [10].

A particularly relevant publication to this project is 'Guitar Effects Recognition and Parameter Estimation With Convolutional Neural Networks' authored by Comunità, Stowell, and Reiss, where the authors used CNNs to perform classification and regression on electric guitar sounds containing timbre manipulations from guitar effects units [26]. They showed that a CNN could predict what guitar pedals had been used on guitar sounds with a classification accuracy above 80% and that CNNs could also perform regression by predicting the continuous guitar effects unit parameters/settings used to create a given guitar sound with a mean absolute error below 0.05 in most cases [26]. Comunità, Stowell, and Reiss's work on guitar pedal classification was built on previous work by Stein and Stein et al. [179, 178], and their work on guitar pedal parameter extraction was built on previous work by Jürgens et al. [89]. Hinrichs et al. extended the guitar pedal classification and pedal parameter regression work of Comunità, Stowell, and Reiss to achieve a pedal classification accuracy of 97.4% and mean absolute parameter extraction errors below 0.016 using CNNs [70].

The ability of CNNs and their convolutional layers to capture spatial hierarchies in images makes them particularly suited for analysing spectrograms, as seen in this section's highlighted examples of previous work. In the context of semantic timbre recognition, CNNs can perform image regression on spectrograms [26]. This involves using CNNs not to classify images but to predict continuous output values representing timbre characteristics

such as 'brightness' or 'darkness'. This regression task requires the CNN to learn from labelled spectrogram image data where each label quantifies timbre attributes, enabling the prediction of timbre descriptors and their degree of presence (timbre magnitude) from unseen spectrograms. Using CNNs for this purpose builds upon their proven capabilities in image analysis tasks, extending their application to the nuanced domain of audio and timbre analysis.

2.3.2 Deep learning for timbre generation

Timbre generation involves synthesising new sounds with desired timbral characteristics. This task is crucial for music production, sound design, and digital audio processing applications. Utilising models like GANs [39, 74, 57, 30] and VAEs [32, 19, 46, 198, 153, 34, 135, 100], researchers can synthesise, interpolate, and modify timbres in ways that were not possible with traditional digital signal processing techniques. Audio generation models like NSynth [46] and Jukebox [34] can generate sounds with different timbral characteristics.

Bonnici, Benning, and Saitis adopt a VAE-GAN hybrid approach [11] to tackle timbre transfer by generating mel-spectrograms of sound that can be reconstructed into audio with the fast Griffin-Lim algorithm [144, 61] and WaveNet [197]. Importantly, Bonnici, Benning, and Saitis note that VAEs typically have more training stability than GANs as GANs can be more susceptible to mode collapse [191, 120, 4], so they decide to fundamentally use a VAE architecture to replace a typical GAN generator to obtain structured latent representations and generate various forms of timbre [11].

In the context of semantic timbre generation and interpolation for this project, this section's highlighted previous work shows that a VAE can be trained on spectrograms of sounds that contain certain timbral characteristics, resulting in a learned latent space that captures those underlying timbral characteristics and enables the generation of new spectrograms. A VAE is preferable to a GAN because of its increased training stability, as Bonnici, Benning, and Saitis highlighted [11]. Furthermore, a VAE can encode different timbres into a continuous, structured latent space [100], allowing for smooth interpolation between, and manipulation of, timbres [46, 34]. This interpolation capability of VAEs is particularly valuable in creating smooth transitions between different timbres, offering significant utility for sound design and music production, as required by this project's objectives.

2.3.2.1 The Griffin-Lim algorithm

A crucial technique often used as the last step in audio generation when working with spectrograms is the Griffin-Lim algorithm [61, 144]. The Griffin-Lim algorithm is an iterative

method for estimating the phase of a Fourier transform given only its magnitude. Many deep learning methods mentioned in this section are trained on magnitude spectrogram representations (in one and two-dimensional forms) and then generate output magnitude spectrograms to represent generated sound. These magnitude spectrograms contain no phase information, so the Griffin-Lim algorithm is used to reconstruct a time-domain audio signal from these magnitude spectrograms. The 'fast' version of the Griffin-Lim algorithm is the implementation most Python packages use [144, 113].

Algorithm 1 The fast Griffin-Lim algorithm [144]

Fix the initial phase $\angle \boldsymbol{c_0}$
Initialise $\boldsymbol{c_0} = \boldsymbol{s} \cdot \boldsymbol{e}^{\cdot i \angle c_0}, \boldsymbol{t_0} = \boldsymbol{P c_2}(\boldsymbol{P c_1}(\boldsymbol{c_0}))$
Iterate for $\boldsymbol{n} = \boldsymbol{1, 2, ...}$
    $\boldsymbol{t_n} = \boldsymbol{P c_1}(\boldsymbol{P c_2}(\boldsymbol{c_{n-1}}))$
    $\boldsymbol{c_n} = \boldsymbol{t_n} + \boldsymbol{\alpha_n}(\boldsymbol{t_n} - \boldsymbol{t_{n-1}})$
    Update $\boldsymbol{\alpha_n}$
Until convergence
$\boldsymbol{x^*} = \boldsymbol{G^\dagger c_n}$

The algorithm begins with an initial random phase and computes the inverse Fourier transform to obtain a time-domain signal. It then alternates the signal between the time and frequency domains, maintaining the original magnitudes but updating phases, using inverse and forward Fourier transforms. This process is repeated multiple times; with each iteration, the phase estimate improves, gradually leading to a more accurate construction of the desired audio signal.

Chapter 3

Deriving timbre descriptors from guitar pedals to create a comprehensive dataset of electric guitar sounds

Before developing models to recognise and generate timbre in electric guitars based on descriptors, it is first crucial to determine which descriptors are relevant to this task and then create an appropriate dataset of training and test examples that incorporate the relevant timbral characteristics. This chapter details the process used to derive descriptors present on popular guitar pedals and their VST software emulations that describe the timbre of electric guitar sounds. The selection criteria for descriptors relating to an electric guitar's timbre are outlined and justified. Following this, the approach for synthesising a dataset of electric guitar sounds incorporating continuously varying timbre based on the derived timbre descriptor scales and carefully selected guitar pedal parameters is explained and justified.

3.1 Defining a timbre descriptor

Recall that timbre is defined as the quality of sound excluding pitch and loudness (see Section 2.1.1). A descriptor is a word or expression that describes or identifies something. Thus, a **timbre descriptor** is a word or expression that describes and conveys the timbre of a sound [142, 18, 50, 175]. Specifically, a timbre descriptor should always be an adjective and any accompanying adverb that describes the timbre of sounds [35, 213, 58, 87]. Notably, onomatopoeias also concur with the definition of descriptors, as onomatopoeias convey vivid resemblances to the sounds they are describing [158]. Hence, an onomatopoeia such as "Pop!" would count as a timbre descriptor because it describes an acoustic resemblance

to the sound it is describing. These are important nuances to note for the following reasons:

- The word “Loudness” is not a timbre descriptor because it refers wholly to sound intensity; thus, it does not concern timbre.
- The word “Vibrato” is not a timbre descriptor because it is a noun referring to the vibrato musical effect [43, 168] and not an adjective. Even though an adjustable parameter named ”Vibrato” would affect the timbre of a sound [2, 131], it is not a descriptor and thus not a timbre descriptor.
- The word “Bright” is a timbre descriptor because it is an adjective that does not solely concern the pitch or loudness of a sound.
- The expression “WahWah” is a timbre descriptor because it is an onomatopoeia that describes a perceptual acoustic resemblance to a particular sound and its timbral characteristics.

3.2 Obtaining relevant semantic timbre descriptors for the electric guitar

3.2.1 Qualitative content analysis of physical guitar pedals

To maximise external validity when deriving which timbre descriptors to incorporate into this project, a content analysis of the language that appears on real physical guitar pedals and amplifiers was performed. To access existing guitar effects units for a comprehensive content analysis study of this nature, I visited the Play Music Today (PMT) store in Cambridge, UK [193] on December 7th, 2023. PMT is a prominent music technology retailer with 13 stores across the UK and over 30 years of experience supplying guitarists with a widespread catalogue of guitar effects units [194], making their stores some of the best locations for observing widely used guitar effects units. There, I took 59 photographs of 72 guitar pedals and amplifiers available for sale. Figure 3.1 shows four examples of the photos taken in the PMT Cambridge music shop.

From those 59 photographs, every distinct keyword appearing on each effect unit’s adjustable parameters was noted and counted. Tables 3.1 and 3.2 show this content analysis on the guitar pedals and amplifiers observed within PMT Cambridge.

3.2.2 Qualitative content analysis of VST guitar pedals

Another increasingly prevalent medium for guitarists to modify timbre is through VST plugins [185, 181, 180] that emulate popular hardware guitar effects units in software

(a) Big Top Creepy Clown Mini Fuzz Pedal [192]

(b) Boss Blues Driver BD–2 Pedal [28]

(c) Way Huge Atreides Analog Weirding Pedal [196]

(d) Line 6 Spider V20 MkII Amplifier [115]

Figure 3.1

Keyword	Count	Keyword	Count	Keyword	Count
Tone	13	Reduction	2	Tweak	1
Level	10	Sub	2	Tweez	1
Volume	10	Repeats	2	Hard	1
Drive	8	Mod	2	Soft	1
Distortion	6	Mix	2	Lo	1
Gain	5	Range	2	Hi	1
Output	5	Sensitivity	2	Depth	1
Delay	4	Damp	1	Shadow	1
Fuzz	4	Dark	1	Sun	1
Rate	4	Medium	1	Balance	1
Roast	3	Light	1	Octave	1
Boost	3	Filter	1	Echo	1
Feedback	2	Ensemble	1	Tape	1
Threshold	2	Compression	1	Blend	1
Equalizer	2	Release	1	Bright	1
Color	2	Sustain	1		
Decay	2	Dry	1		

Table 3.1: Parameter keyword frequency on PMT's [193] guitar pedals.

Keyword	Count	Keyword	Count	Keyword	Count
Clean	12	Bass	2	Boutique	1
Crunch	10	Tone	2	Chunk	1
Drive (Overdrive)	9	Chime	2	Insane	1
Hi	3	Low	1	Vibe	1
Acoustic	3	Flat	1	Ambient	1
Boost	3	Special	1	Intensity	1
Bright	3	Echo	1	Speed	1
Warm	3	Voice	1	Sensitivity	1
Presence	3	Brown	1	Harsh	1
Lead	2	Dynamic	1	Raw	1

Table 3.2: Parameter keyword frequency (excluding volume-related keywords) on PMT's [193] guitar amplifiers.

[137, 202]. One of the most popular and highly regarded guitar effects units VST plugins is Guitar Rig [15, 14, 83]. Another content analysis study identical to the one undertaken in the PMT store was performed on Guitar Rig 7 Pro and its pedals/amplifiers.

Uniquely, Guitar Rig 7 Pro also provides a detailed user manual describing all effects units and their influence on sounds [84]. The context on which aspect of timbre each effect unit parameter affects is given in these descriptions. This is called 'timbre-context', and it correlates to the spectral and temporal aspects of timbre identified in Sections 2.1.2 and 2.1.4.1. Hence, Guitar Rig's content analysis extends from solely counting the occurrence of parameter keywords on effects units to additionally counting the occurrence of keywords within the effect units' descriptions. Table 3.3 shows the results of this content analysis, but only timbre-related keywords are displayed for conciseness.

Some timbre-related keywords were assigned an 'ambiguous' timbre-context because the keyword was given multiple timbral contexts throughout the pedal parameters and their descriptions. For example, 'warm' was related to various spectral aspects of timbre, such as distortion and filter effects, alongside temporal aspects of timbre, such as dynamics effects like compression.

Another popular guitar effects unit VST plugin is Pedalboard [80, 16], packaged with Logic [81]. An identical content analysis study to the previous PMT studies was carried out on Pedalboard, and the results can be seen in Table 3.4 (again, only timbre keywords are shown for conciseness).

Keyword	Timbre Context	Count
Compressed	Ambiguous	163
Bright	Spectral (Filter)	54
Wide	Ambiguous	41
Clean	Spectral (Distortion)	39
Smooth	Temporal (Dynamics)	35
Resonant	Spectral (Filter)	28
Dry	Ambiguous	27
Warm	Ambiguous	27
Dark	Spectral (Filter)	21
Rich	Ambiguous	19
Presence/Present	Spectral (Filter)	18
Soft	Temporal (Dynamics)	17
Loose	Ambiguous	16
Fuzz/Fuzzy	Spectral (Distortion)	15
Brilliant	Spectral (Filter)	14
Crunch/Crunchy	Spectral (Distortion)	12
WahWah	Temporal (Oscillation)	12
Tight	Temporal (Dynamics)	11
Wet	Ambiguous	11
Dirt/Dirty	Spectral (Distortion)	10
Flutter/Fluttery	Temporal (Oscillation)	9
Wow	Temporal (Oscillation)	9
Shimmer/Shimmering	Temporal (Oscillation)	8
Damp	Ambiguous	7
Sharp	Temporal (Dynamics)	6
Stutter/Stuttering	Temporal (Oscillation)	6
Crush/Crushed	Spectral (Distortion)	5
Fat	Spectral (Filter)	5
Punch/Punchy	Temporal (Dynamics)	5
Thin	Spectral (Filter)	5
Jitter/Jittery	Temporal (Oscillation)	4
Scoop/Scooped	Spectral (Filter)	4
Narrow	Spectral (Filter)	1

Table 3.3: Timbre descriptor keyword frequency in Guitar Rig Pro 7 [83] & its component manual [84].

Keyword	Count	Keyword	Count	Keyword	Count
Fuzz/Fuzzy	8	Dark	1	Growl	1
Bright	4	Dirt/Dirty	1	Scoop/Scooped	1
Fat	3	Flutter/Fluttery	1	Smooth	1
Squash	2	Grain	1	Resonant	1
WahWah	2	Grind	1	Roar	1
Compressed	1	Grit	1		

Table 3.4: Timbre descriptor keyword frequency in Pedalboard [80].

3.2.3 Final derived timbre descriptors

3.2.3.1 Parameter keyword inter-rater reliability

The first step for selecting relevant timbre descriptors was eliminating keywords observed on effects units that were not timbre descriptors. Keywords that did not match the definition for a timbre descriptor (see Section 3.1) were eliminated. This process involves a 'rating' procedure where keywords were rated as timbre descriptors or not. To strengthen the reliability of my judgement on the observed keywords, an inter-rater reliability experiment was performed between myself and an independent second rater experienced in musical timbre research. From the collection of all keywords observed across the previous content analysis studies, 20 randomly selected keywords I rated as non-timbre descriptors and 20 randomly selected keywords I rated as timbre descriptors were presented to the other rater. Table 3.5 shows those 40 keywords.

Vibrato	Brilliant	Output	Pitch
Rich	Reduction	Input	Shimmering
Hard	Inverse	Present	Ratio
Scooped	Bright	Smooth	Boost
AM/RM	Crunchy	Level	Mute
Drive	Reverse	Master	Gain
Volume	Sync	Fuzzy	Crushed
Punchy	Freeze	Warm	Clean
Fat	Speed	Stuttering	Size
Harsh	Thin	Dark	Raw

Table 3.5: Inter-rater keywords.

This experiment achieved 100% inter-rater agreement for rating non-timbre descriptors. This result provides credibility for my judgement on which keywords could be eliminated as non-timbre descriptors.

3.2.3.2 Gaining comprehensive & representative timbre descriptors

When selecting timbre descriptors for further consideration, it was crucial to ensure the final list of descriptors comprehensively represents the spectral and temporal categories of timbre identified in Section 2.1.2 and the guitar-specific timbre categories of distortion and oscillation effects identified in Section 2.1.4.1. These considerations resulted in Table 3.6's four categories of timbre needing representation in the final derived timbre descriptors.

Spectral Timbre		Temporal Timbre	
DistortionFX	FilterFX	DynamicsFX	OscillationFX

Table 3.6: Timbre categories for the descriptors.

Timbre descriptors collected from the previous content analysis studies could be categorised into these four timbral categories by referring to the context of effect units those descriptors originally appeared on. For descriptors collected from Guitar Rig, the accompanying user manual [84] assisted this process.

Moreover, it was decided that the final list of timbre descriptors should represent these timbre categories in equal measures. Hinrichs et al. investigated 8 guitar pedals and their parameters [70], while Communità, Stowell, and Reiss used 13 [26]. This project extends that previous work by incorporating 20 timbre descriptors in total. Consequently, the five most popularly occurring timbre descriptors from the previous content analysis studies aligning with each of the four timbre categories were chosen to represent their corresponding timbre category.

Crucially, if an observed timbre descriptor had an ambiguous timbre context, it was eliminated from the final list of 20 timbre descriptors. This ensures that all final timbre descriptors could be utilised to objectively synthesise new audio data for the proposed dataset with no ambiguous timbral characteristics that may confuse participants in future perceptual studies. Lastly, if two or more timbre descriptors were found to be used interchangeably because they describe the same timbral characteristics, then the most popular timbre descriptor was kept, and the others were discarded. Two notable examples of this occurred with 'bright'/'brilliant' [84] and 'wahwah'/'wow' [84, 103, 86], where 'bright' and 'wahwah' were retained due to their more frequent occurrence.

Table 3.7 shows the final twenty timbre descriptors used in this project, which were derived from the processes detailed in this chapter. For brief explanations of the acoustic characteristics of sounds described by these derived descriptors, refer to Appendix A.

Spectral Timbre		Temporal Timbre	
DistortionFX	FilterFX	DynamicsFX	OscillationFX
Clean	Bright	Punchy	Fluttery
Crunchy	Dark	Sharp	Jittery
Crushed	Fat	Soft	Shimmering
Dirty	Resonant	Smooth	Stuttering
Fuzzy	Thin	Tight	WahWah

Table 3.7: Final timbre descriptors.

3.3 Creating a comprehensive dataset for semantic timbre recognition & generation

A diverse dataset containing a wide range of timbral characteristics is fundamental for training neural networks to accurately recognise and generate distinct timbral qualities. Training on such a dataset helps future models generalise better across unseen data, avoiding overfitting to a narrow set of sounds. No existing datasets contained suitable electric guitar audio for this project's timbre recognition and generation goals with the derived timbre descriptors, so a new dataset had to be created.

3.3.1 Steps for the synthesis of the proposed dataset

3.3.1.1 Estimating size for the proposed dataset

Before constructing the dataset, an initial informed estimation of how big the dataset should be to achieve meaningful results for timbre recognition and generation was made. Hinrichs et al. used 268,800 audio samples of guitar sounds for training their CNNs to classify applied guitar pedal effects with 97.4% accuracy and predict guitar pedal parameter settings with mean absolute parameter extraction errors below 0.016 [70]. With guidance and inspiration from this previous work, the proposed dataset for this project was designed to include a similar number of samples.

3.3.1.2 Obtaining clean guitar sounds

Next, a collection of natural, unprocessed monophonic guitar note recordings had to be obtained. This collection of audio samples would make up the 'clean' electric guitar sounds, providing a controlled platform from which pedal effects can alter timbre to specification. An important requirement for each guitar note recording is that they had to be played with various pitches. This helps future ML models learn general timbral characteristics associated with timbre descriptors spanning real-occurring variations in pitch and allows for further insight into timbre recognition/generation across different pitches.

For evaluation and experimental purposes, sounds originating from only one electric guitar model line were used. This reduces bias between different guitar models in future experiments with human participants. The specific guitar chosen for gaining clean samples was the Fender Stratocaster, one of the most popular and well-regarded electric guitars of all time [76, 88] and the most produced electric guitar worldwide [139, 40]. For these reasons, it is the most appropriate electric guitar model to investigate to maximise the external validity of the timbre recognition/generation models and their evaluation.

The EGFxSet dataset [141] contains clean, unprocessed recordings of real monophonic guitar notes from a Fender Stratocaster that range from pitches E2-D6 [147]. The notes are presented per fret position and string on the 22-fret, six-string Fender Stratocaster. Furthermore, the EGFxSet contains recordings of these notes across five pickup configurations [141, 210, 143, 182, 163]. These EGFxSet notes were chosen, resulting in 690 audio recordings of clean electric guitar notes from which the rest of the dataset was constructed.

3.3.1.3 Modifying timbre according to the timbre descriptors

As mentioned in Section 3.3.1.1, at least 200,000 audio recordings are required to effectively train neural networks to perform the timbre recognition and generation tasks specified in this project. A feasible and valid option to achieve this was to use software emulations of popular guitar pedals and choose different pedal settings for creating new recordings of the original clean EGFxSet sounds, where each pedal/setting introduces unique timbral modifications, allowing the proposed dataset to capture a large spectrum of possible timbre.

Guitar Rig 7 Pro [83] was chosen for emulating pedals because musicians and music producers highly rate it for its authentic and realistic capture of real guitar pedals [185, 27]. It is also one of the most popular software options for emulating guitar pedals, further boosting the external validity of this project's final models and findings. As observed in Section 3.2.2, specific parameters on Guitar Rig's pedals are described or named with timbre descriptors. Hence, for each of the 20 final derived timbre descriptors, the descriptor's most relevant corresponding Guitar Rig pedal and parameter was identified, and Table 3.8 shows these Guitar Rig parameters. Notice there is no pedal/parameter mapped to the 'Clean' descriptor because the original recordings already represent 'clean' sounds.

3.3.1.4 Incorporating timbre magnitude

The collection of 690 clean guitar sounds was then processed through each pedal parameter identified in Table 3.8, where the parameter was gradually increased from its minimum to maximum value. For each intermediate step between a parameter's minimum and maximum value, the pedal's outputs were saved as new audio files. This process resulted

Timbre Descriptor	Guitar Rig Pro 7 Pedal	Pedal Parameter
Crunchy	Bite	Crunch
Crushed	Traktor's Digital LoFi	Crush
Dirty	Dirt	Drive (Mode II)
Fuzzy	Fuzz	Fuzz
Bright	Pro-Filter	Freq & Slope (Mode Set to HP)
Dark	Pro-Filter	Freq & Slope (Mode Set to LP)
Fat	Supercharger GT	Character (Fat Activated)
Resonant	filterbank	Res
Thin	Pro-Filter	Freq & Slope (Mode Set to BP)
Punchy	Supercharger GT	Compress (Punch Activated)
Sharp	Transient Master	Attack (0% to 100%)
Soft	Transient Master	Attack (0% to -100%)
Smooth	Transient Master	Attack (0% to -100%) & Smooth Activated
Tight	Stomp Compressor	Sustain (Attack Min & Release Max)
Fluttery	Tape Wobble	Flutter
Jittery	Bite	Jitter
Shimmering	Replika Shimmer	Shimmer
Stuttering	Tremolo	Intensity (Up & Left Set to Minimum)
WahWah	Wah Wah	Wah

Table 3.8: Guitar Rig pedals & their parameters [84] mapped to timbre descriptors.

in guitar sounds containing varying degrees of presence for the timbre characteristics described by each parameter's corresponding timbre descriptor. In other words, when the 'dark' parameter was gradually increased, the saved sounds got 'darker'. Designing sounds to be synthesised this way purposefully mimics the semantic timbre scales observed and investigated by von Bismarck [201], Reymore and Huron [154], Reymore et al. [155], and Zacharakis et al. [216]. Hinrichs et al. adjusted pedal parameters in 20-point steps on a 0-100 point scale [70]. All identified parameters from Guitar Rig also have a 0-100 point scale, and this dataset looks to extend that of Hinrichs et al. by saving output audio files at every 5-point step for each parameter. This results in more subtle variations in pedal settings that alter the timbre for each descriptor, which aids in exposing neural networks to more nuanced fine-grained timbral characteristics for each timbre descriptor.

The parameter value between 0-100 that results in each set of saved audio files within each timbre descriptor group is called '**timbre magnitude**'. A 'dark' sound produced from a dark parameter setting of 50 has less 'dark' timbre magnitude than a 'dark' sound produced from a dark parameter setting of 100.

3.3.1.5 Final dataset metrics

The 690 clean samples were processed by each timbre descriptor's Guitar Rig pedal parameter in Table 3.8 with timbre magnitudes in 5-point steps from 0 to 100 (0, 5, 10,

..., 95, 100) using the Logic DAW [81]. This results in 14,490 audio samples being saved for each of the 19 timbre descriptors and 275,310 audio samples saved in total across the entire dataset. This number of audio samples is comparable to the number of samples Hinrichs et al. used [70].

This new dataset for semantic timbre analysis in electric guitar sounds is the first novel contribution of this project, and it is hereby called the '**SemanticTimbreDataset**' [20]. Appendix B contains selected sounds from the dataset visualised as spectrograms.

Creating this comprehensive dataset of electric guitar sounds processed through various guitar pedals is foundational for this project's goal of advancing research in semantic timbre recognition and generation. The SemanticTimbreDataset encourages neural networks to associate specific auditory timbral characteristics with semantic timbre descriptor labels such as 'bright', 'dark', or 'fuzzy'. This association is vital for deep learning models to understand and predict the derived semantic timbre descriptors from new guitar sounds and to generate sounds incorporating the descriptors' timbral characteristics.

Chapter 4

Semantic timbre recognition

This chapter explains the design methodology and implementation process for the proposed timbre recognition system powered by CNNs. First, the timbre recognition problem is formally clarified according to the scope of this project and its data. Second, a CNN-based timbre recognition system is proposed where its design and implementation decisions are justified and illustrated. This justification includes reasoning for treating the timbre recognition problem as a regression problem. Lastly, the training procedure for the timbre recognition CNNs is rationalised.

4.1 Clarification of the timbre recognition problem

This project's scope considers analysing timbral characteristics within monophonic electric guitar notes that timbre descriptors can describe. Hence, for a given fixed-length audio file of a single guitar note, the goal is to estimate the sound's timbre magnitude for each desired timbre descriptor.

It is essential to note the emphasis on estimating the magnitude of each timbre descriptor. This distinction reflects the long-standing empirical findings that show humans perceive timbre in terms of linear continuous scales of **timbre magnitude** for timbre descriptors [201, 154, 155, 216], where a sound can be perceived as being more or less of a certain descriptor (dark, bright, etc.) than other sounds.

Furthermore, it is crucial to understand that when recognising timbre via timbre descriptors within a single instrument, any sound produced from that instrument may contain multiple characteristics of timbre that relate to multiple timbre descriptors at varying magnitudes. For example, a monophonic guitar note played softly in the guitar's lowest register could be described as 'dark' **and** 'soft' rather than solely 'dark' or solely 'soft'. As a result, that note is said to have high magnitudes of darkness and softness but also low magnitudes of brightness and harshness.

Previous work on timbre recognition has focused on estimating which instrument a

sound has originated from [68, 148, 10], and thus, they have treated timbre recognition as a classification problem. For the specific context of using descriptors to recognise timbre, splitting each descriptor into separate classes does not make sense because multiple descriptors may describe a sound's timbre to varying degrees, as previously discussed. Consequently, the problem of timbre recognition in this project is treated as a regression problem where timbre magnitudes of timbre descriptors are estimated.

4.2 Proposed timbre recognition system

4.2.1 Timbre recognition system overview

The SemanticTimbreDataset created earlier in the project (see Chapter 3) contains audio examples of monophonic electric guitar notes with pitches from E2 to D6 [147] at 21 equally spaced varying timbre magnitudes (0%, 5%, 10%, ..., 100%) for 19 derived timbre descriptors (all descriptors excluding 'Clean'). These audio files were separated into each of their timbre descriptor categories and then processed into RGB log-spectrogram images. For each timbre descriptor category, the spectrogram images were further separated by their timbre descriptor magnitude, and these magnitudes were then used as example dependent variable values for performing regression. With these spectrogram images separated by timbre descriptor into 19 sets, a distinct CNN was trained on each distinct set of spectrograms to perform image regression and predict the corresponding timbre descriptor's magnitude on new input RGB log-spectrograms.

A notable advantage of using this approach and training multiple CNN image regression models where each model estimates the timbre magnitude for a single descriptor is that the magnitudes of multiple timbre descriptors can be recognised for any given audio file. This can be achieved by simply calling the desired timbre descriptor recognition models to predict the magnitude values of an audio file in unison, allowing for a more holistic and nuanced perspective on semantic timbre recognition that reflects previous literature on timbre perception and semantics [124, 158].

4.2.2 Data preparation

For each of the SemanticTimbreDataset's 19 timbre descriptors, there are 14,490 audio files that can be used to train, validate, and test a single CNN image regression network for estimating timbre magnitude for a timbre descriptor. Those 14,490 audio files are grouped by timbre magnitude and organised in directories according to their descriptor and magnitude (eg. "TimbreDescriptor/Magnitude/AudioFile.wav", "Bright/50/1-1.wav").

For each audio file, an RGB log-spectrogram of the audio in the file was synthesised and saved in a new directory of spectrograms with the same folder structure as the original

audio file (eg. "TimbreDescriptor/Magnitude/spectrogram.png", "Bright/50/1-1.png"). The Python programming language and its Librosa [127], NumPy [64], and Matplotlib [77] libraries were used to generate the spectrograms and save them accordingly. The process for obtaining the RGB log-spectrogram of an audio file was:

- **Step 1:** The audio file was loaded using Librosa's `load` function [111] with the original sampling rate of 48kHZ.
- **Step 2:** The Short-Time Fourier Transform of the audio was taken using Librosa's `stft` function [111]. The `stft` function's default `n_fft` value of 2048 was used along with a `hop_length` of 512.
- **Step 3:** Absolute values of the `stft` were taken to obtain a magnitude spectrogram using NumPy's `abs` function [132].
- **Step 4:** Librosa's `amplitude_to_db` function [111] was applied to the magnitude spectrogram to obtain a log-spectrogram of the audio file. This step is important as it adjusts a spectrogram's linear scale of magnitude per frequency bin into a logarithmic decibel scale, and this decibel scale better portrays how humans perceive sounds [212, 53]. By default, the `amplitude_to_db` function assigns the spectrogram's maximum magnitude to be 0dB, so it also serves as a loudness normalisation step.
- **Step 5:** The resulting log-spectrogram was saved as a figure without axes or other meta-information using Librosa's `specshow` function [112] and Matplotlib's `savefig` function [122].

4.2.3 Convolutional neural network architecture

Each estimation of timbre magnitude for one of the 19 timbre descriptors is achieved by a single image regression CNN that was trained on data related to the timbre descriptor of interest. These CNN models were implemented in TensorFlow v2.15.0 [1, 188] using Keras v2.15.0 [25, 94] as an API. Figures 4.1 & 4.2 illustrate each model's architecture.

4.3 Training the timbre recognition system

Since a single network should perform timbre magnitude estimation for one timbre descriptor, the file paths for the RGB log-spectrograms for one timbre descriptor subgroup were loaded into a pandas dataframe [138, 205] along with the timbre descriptor magnitudes associated with each log-spectrogram. The 14,490 log-spectrograms were divided into an 80%:20% split using scikit-learn's `train_test_split` function [140, 165], where the 80%

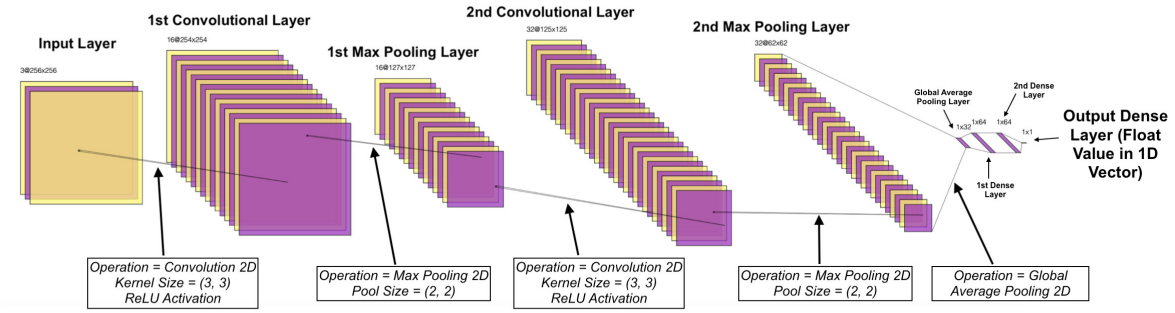


Figure 4.1: Timbre recognition CNN model architecture.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 conv2d (Conv2D)             (None, 254, 254, 16)      448

 max_pooling2d (MaxPooling2  (None, 127, 127, 16)      0
 D)

 conv2d_1 (Conv2D)           (None, 125, 125, 32)      4640

 max_pooling2d_1 (MaxPoolin  (None, 62, 62, 32)        0
 g2D)

 global_average_pooling2d (  (None, 32)                0
 GlobalAveragePooling2D)

 dense (Dense)               (None, 64)                2112

 dense_1 (Dense)             (None, 64)                4160

 dense_2 (Dense)             (None, 1)                 65

=================================================================
Total params: 11425 (44.63 KB)
Trainable params: 11425 (44.63 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 4.2: Timbre recognition CNN model architecture provided by Keras' `model.summary()` [95].

portion was reserved for training, and the 20% portion was reserved for testing the fully trained image regression model after all training epochs.

Next, these training and testing portions were supplied into Keras `ImageDataGenerator` objects [189], where all images were rescaled from 0-255 values into 0-1 values, allowing the CNN to train correctly. Moreover, the training images were split into 80% and 20% portions, where the 80% portion was used to train the CNN and the 20% portion was used to validate the CNN's performance after each training epoch. This results in a final training set of 9274 images and magnitudes, a final validation set of 2318 images and magnitudes, and a final test set of 2898 images and magnitudes for training each timbre descriptor recognition model.

All images were then prepared for training per epoch via Keras' `flow_from_dataframe` function [189], where each image was resized from 496x369 pixels to 256x256 pixels for more efficient and consistent computation [184]. The training and validation sets of images were shuffled, while the test set was not shuffled as it was employed only once after all training epochs.

Models were then configured with Keras' `model.compile()` method [96] to use an Adam optimizer [99] with default settings determined by Keras [93]. The loss that models minimised during training was configured to be the mean squared error (MSE) [97] between a model's output prediction of a timbre descriptor's magnitude and the originally provided timbre magnitude.

At this stage, each model was ready to be fit to the training data with Keras' `model.fit()` method [96], which starts each model's training loop. The `fit` method specified for each model to train using the shuffled training image subset and to be validated on the shuffled validation image subset for each epoch in 100 epochs. Additionally, an early stopping callback was defined in the `fit` method's parameters, specifying that if the validation loss did not improve across five consecutive epochs, the training loop would stop, and the model's weights from the epoch with the smallest validation loss would be restored. This provides a mechanism for alleviating overfitting while allowing training to continue for as long as meaningful learning occurs.

For every training epoch, a model's losses on the training and validation images were reported and saved. These metrics are known as the training metrics. After completing all training epochs, the test images set aside at the start of the process provided the final regression test metrics of a fully-trained model for initial model evaluation. These test metrics include:

- The model's root mean squared error (RMSE) across the test images, obtained via Keras' `model.evaluate()` method [96].
- The model's $\boldsymbol{R^2}$ score on the test images, provided by scikit-learn's `r2_score` function [140, 164].

- The Pearson correlation coefficient between the test images' timbre magnitude labels and the model's estimations of timbre magnitude from the test images, calculated by SciPy's `pearsonr` function [200, 167].

This whole training process results in one trained CNN model for timbre recognition for one timbre descriptor, and it was repeated another 18 times for all the remaining timbre descriptors and their relevant log-spectrogram data. See Section 4.3.1 for the training and test metrics of every timbre descriptor recognition CNN.

4.3.1 Training and test metrics

Plots of the training history metrics for all 19 trained timbre recognition models can be seen in Figures 4.3-4.6. All timbre recognition models trained with minimal overfitting, as evidenced by the close training and validation loss curves in Figures 4.3-4.6.

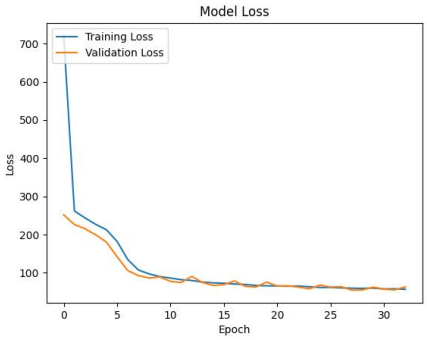
Table 4.1 shows the final test metrics for every trained timbre recognition model.

Table 4.1: Test metrics for every trained timbre recognition model.

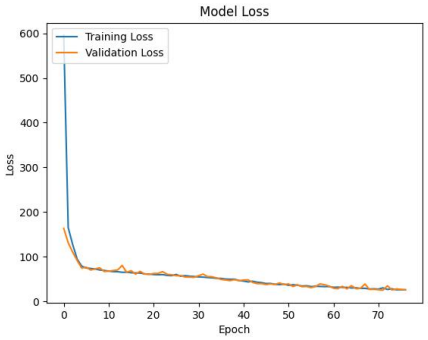
Timbre Descriptor	Test RMSE	Test $\boldsymbol{R^2}$ Score	Pearson Correlation Coefficient (p-value)
Bright	9.873	0.893	0.945 (p: < 0.001)
Crunch	7.422	0.940	0.970 (p: < 0.001)
Crush	5.018	0.972	0.986 (p: < 0.001)
Dark	17.224	0.676	0.827 (p: < 0.001)
Dirt	8.105	0.928	0.965 (p: < 0.001)
Fat	11.273	0.861	0.929 (p: < 0.001)
Flutter	4.517	0.978	0.989 (p: < 0.001)
Fuzz	5.782	0.963	0.982 (p: < 0.001)
Jitter	8.440	0.922	0.961 (p: < 0.001)
Punch	14.691	0.764	0.875 (p: < 0.001)
Resonant	3.630	0.986	0.993 (p: < 0.001)
Sharp	25.850	0.270	0.529 (p: < 0.001)
Shimmer	9.232	0.907	0.953 (p: < 0.001)
Smooth	18.137	0.641	0.842 (p: < 0.001)
Soft	16.824	0.691	0.835 (p: < 0.001)
Stutter	4.961	0.973	0.988 (p: < 0.001)
Thin	7.556	0.938	0.968 (p: < 0.001)
Tight	7.766	0.934	0.967 (p: < 0.001)
WahWah	5.052	0.972	0.986 (p: < 0.001)

The final test metrics confirm that most timbre recognition models successfully learnt to recognise the timbral characteristics described by their corresponding timbre descriptor. However, the 'sharp' recognition model struggled to learn the short attack phase in a 'sharp' sound's ADSR envelope [65, 125, 17, 51, 82]. In hindsight, this may have been caused by the 5-second time window employed to create training spectrograms, which

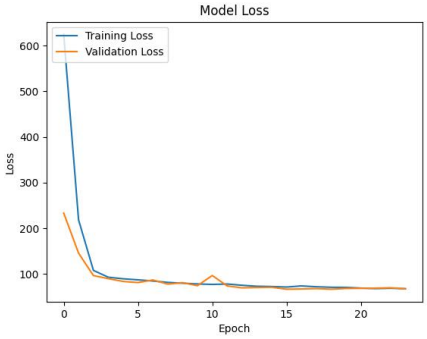
may have been too long for the CNN to capture the short rise in magnitude at the far-left side of the 'sharp' training spectrograms as displayed by SemanticTimbreDataset's 'sharp' spectrograms in Appendix B.



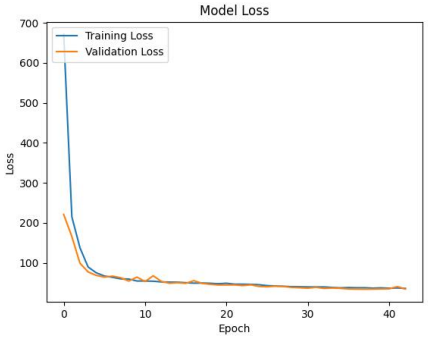
(a) Crunch



(b) Crush

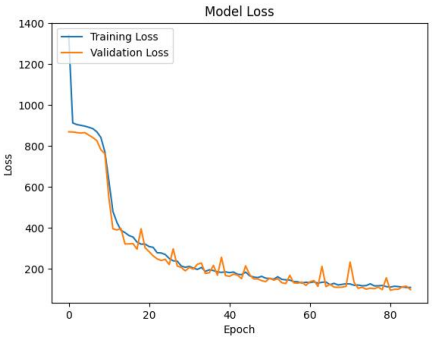


(c) Dirt

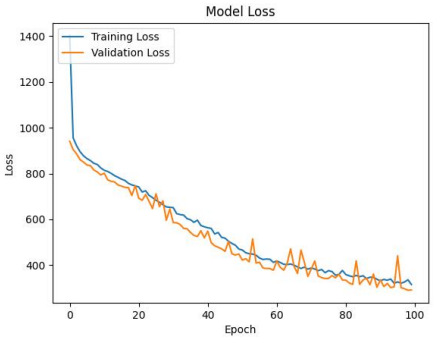


(d) Fuzz

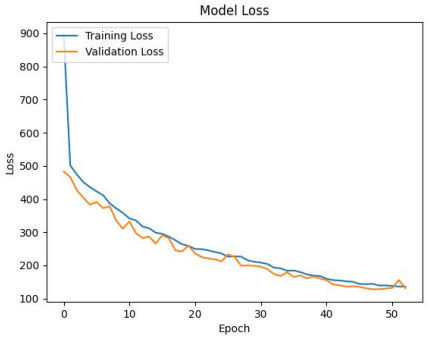
Figure 4.3: DistortionFX timbre recognition models' training metrics.



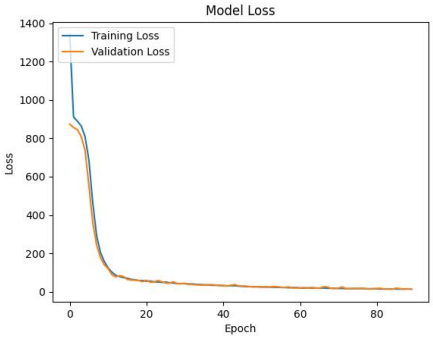
(a) Bright



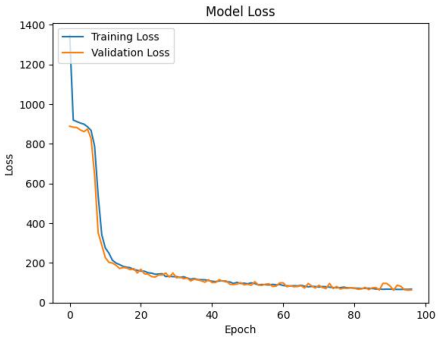
(b) Dark



(c) Fat

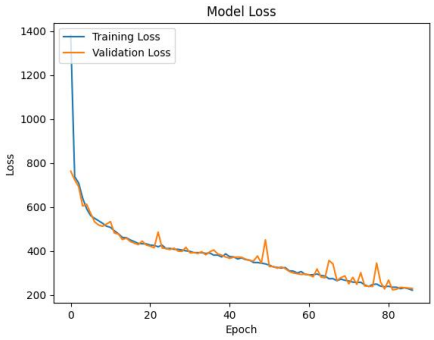


(d) Resonant

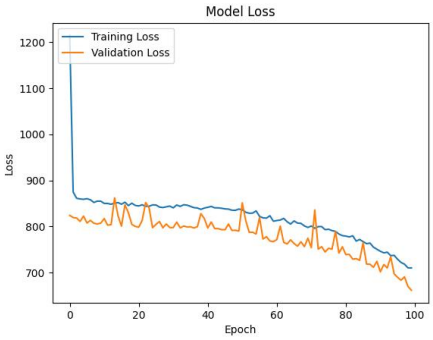


(e) Thin

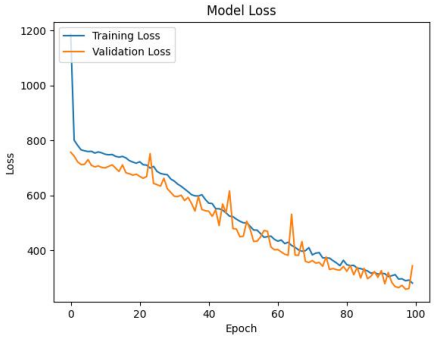
Figure 4.4: FilterFX timbre recognition models' training metrics.



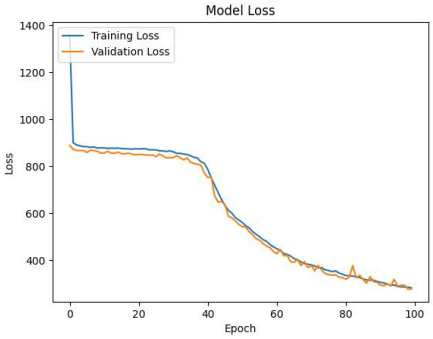
(a) Punch



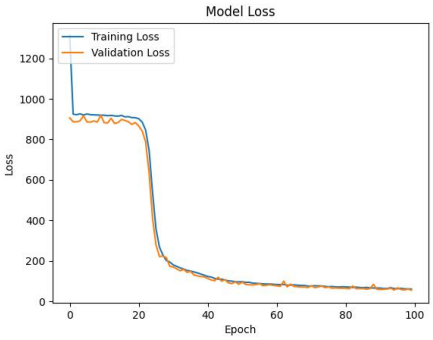
(b) Sharp



(c) Smooth

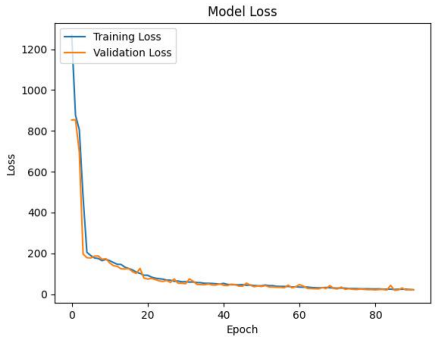


(d) Soft

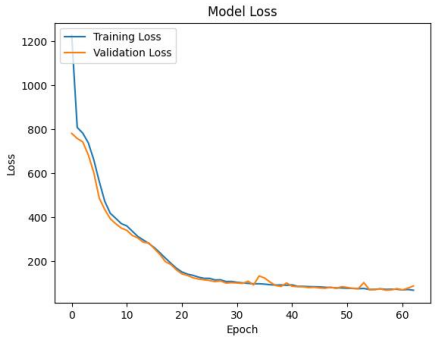


(e) Tight

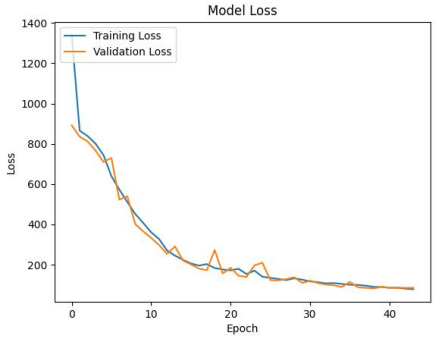
Figure 4.5: DynamicsFX timbre recognition models' training metrics.



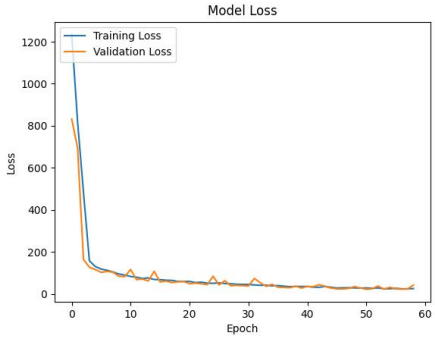
(a) Flutter



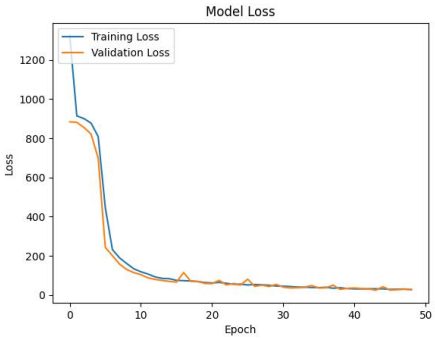
(b) Jitter



(c) Shimmer



(d) Stutter



(e) WahWah

Figure 4.6: OscillationFX timbre recognition models' training metrics.

Chapter 5

Semantic timbre generation and timbre interpolation

This chapter explains the timbre generation system that generates electric guitar sounds containing timbral characteristics described by the timbre descriptors present within the SemanticTimbreDataset (see Chapter 3). Moreover, this chapter also presents the timbre generation system's capacity to perform timbre interpolation between two distinct sounds described by two different timbre descriptors to generate new unseen timbre mixtures.

5.1 Clarification of the timbre generation and timbre interpolation problems

Alongside recognising timbral characteristics described by semantic timbre descriptors, this project aims to synthesise monophonic guitar notes that replicate specific timbral qualities using semantic timbre descriptors from the SemanticTimbreDataset. The generated sounds should perceptually maintain the described timbre and timbre magnitude, sounding natural and closely matching the quality of the original SemanticTimbreDataset samples. For example, prompting the timbre generation system to generate a 'fuzzy' E4 guitar note with a 'fuzz' magnitude of 50/100 should result in an audio file that audibly resembles such a note from the SemanticTimbreDataset. It is also critical for users to control the timbre, timbre magnitude, and pitch during synthesis, especially for applications in music production, to ensure the sound aligns with specific timbre and pitch requirements.

Additionally, the project seeks to create new electric guitar sounds with novel timbral characteristics by interpolating between different samples in the dataset. This method, termed **timbre interpolation**, blends multiple timbral traits to produce unique sounds that are not achievable with conventional instruments or production techniques.

5.2 Proposed timbre generation system

5.2.1 Timbre generation system overview

The SemanticTimbreDataset's audio files were pre-processed into min-max normalised single-channel grayscale log-magnitude spectrograms for training an unsupervised VAE. The VAE's encoder learnt to encode these unlabelled spectrograms into an organised latent space, while its decoder aimed to reconstruct the spectrograms from their latent representations as accurately as possible. These generated spectrograms were then transformed into audio using min-max denormalisation and the Griffin-Lim algorithm [61, 144].

The training data included various examples of electric guitar timbres across different timbre descriptors, magnitudes, and pitches (E4-D6), aiming to facilitate timbre generation for any of the 23 specified pitches within this range. The training note range was intentionally smaller than that used for timbre recognition to align with available computational resources and fit within the project's timeline.

5.2.1.1 Why use an unsupervised VAE instead of a conditional VAE?

Given that the training dataset is labelled with timbre descriptors, their magnitudes, and pitch, one might question the use of a standard unsupervised VAE over a conditional VAE [173] (CVAE). A standard VAE generates new data points by encoding them into a latent space and then sampling from this space, while a CVAE additionally uses condition variables to specifically guide the generation of outputs with desired characteristics [173].

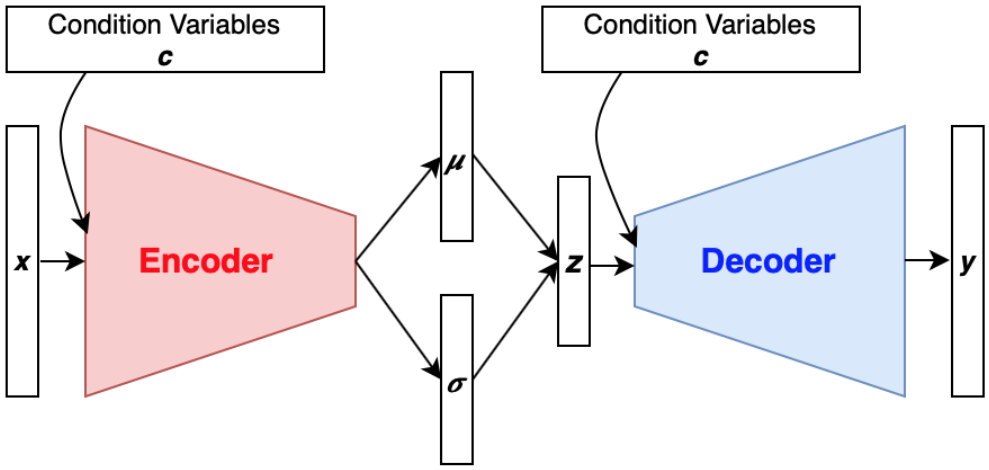


Figure 5.1: CVAE architecture.

In this project, the timbre descriptor, timbre magnitude, and pitch labels from the SemanticTimbreDataset would likely serve as condition variables for a CVAE. A condition vector for a 'fuzzy' guitar note might tell the CVAE it is purely 'fuzzy' with a fuzz

magnitude of 100, setting other descriptors to zero. However, this can misleadingly simplify the note's complexity, implying it lacks other qualities like 'bright' or 'tight', etc. In contrast, an unsupervised (standard) VAE receives inputs with no set conditions, allowing it the necessary freedom to learn and organise the full spectrum of timbre in its latent space. A CVAE could restrict this flexibility by enforcing rigid latent representations, which may hinder the proper grouping of similar sounds.

Grekow and Dimitrova-Grekow used a CVAE to generate monophonic music sequences reflecting specific emotions like happiness and sadness, with the CVAE conditioned by the emotion labels from the training data [59]. Similarly, Liu et al. utilised a CVAE to create spectrograms of specific sounds from the UrbanSound8K dataset [117, 160]. Both cases show CVAEs effectively handling datasets with distinct categories. However, this project focuses on the continuous, intertwining nature of timbre features [124, 155, 215]. Hence, an unsupervised VAE is preferred over a CVAE to achieve the necessary nuanced timbre generation and interpolation.

Grekow and Dimitrova-Grekow noted that using four distinct emotion categories to condition a VAE simplified the emotional range in music and suggested that future work might use continuous scales of emotions [59], which would require modifications to their CVAE architecture and data handling methods. By recognising the continuous and complex nature of timbre features from the start, this project aligns with the advanced direction proposed by Grekow and Dimitrova-Grekow for their music generation research.

The preference for an unsupervised VAE over a CVAE is based on the current limitations of label availability in the SemanticTimbreDataset. In the future, a CVAE could be used with a more holistically labelled dataset that includes all relevant timbre descriptors and magnitudes for each sound, unlike the limited labels currently available. Developing such a dataset would involve extensive human studies on timbre perception but could enable future timbre generation work using CVAEs.

5.2.2 Training data preparation

A subset of the audio files available in the SemanticTimbreDataset was used to train the timbre generation system [22]. For each of the 19 timbre descriptors present in the dataset, the audio files with pitches E4-D6 and 25%, 50%, 75%, and 100% timbre magnitudes were selected. A set of clean guitar samples across the same pitches was also selected to represent the 'Clean' timbre descriptor. This meant that 1771 files in total were used for training.

For each audio file, a single-channel grayscale log-magnitude spectrogram was produced. The Python programming language and the Librosa and NumPy libraries [127, 64] were used to generate these spectrograms using the following process:

- **Step 1:** 0.74 seconds of audio was loaded using Librosa's `load` function [111] with a sampling rate of 22.050kHZ.
- **Step 2:** The Short-Time Fourier Transform of the audio was taken using Librosa's `stft` function [111] with a `n_fft` value of 1024 and a `hop_length` of 512.
- **Step 3:** Absolute values of the `stft` were taken to obtain a magnitude spectrogram using NumPy's `abs` function [132].
- **Step 4:** Librosa's `amplitude_to_db` function [111] was applied to the magnitude spectrogram to obtain a log-magnitude spectrogram of the audio file. See Section 4.2.2 for the significance of this step.
- **Step 5:** Min-max normalisation [145] was applied to the log-magnitude spectrogram. The spectrogram's original minimum and maximum values across the whole array were stored in a separate dictionary for future retrieval during the sound generation process when VAE-generated log-magnitude spectrograms are denormalised.
- **Step 6:** The resulting minmax-normalised log-magnitude spectrogram was saved as a NumPy array using NumPy's `save` function [133].

5.2.3 Variational autoencoder network architecture

All synthesised sounds with the desired timbral characteristics are generated from a single VAE. The VAE model for performing timbre generation was implemented in TensorFlow v2.15.0 [1, 188] using Keras v2.15.0 [25, 94] as an API. Figures 5.2 & 5.5 illustrate the complete VAE architecture, while Figures 5.3 & 5.4 isolate the VAE's encoder and decoder components.

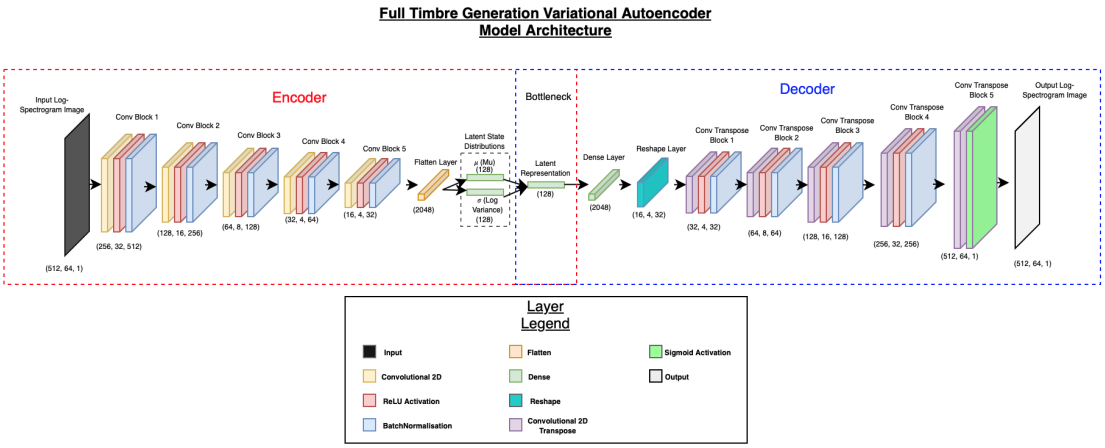


Figure 5.2: VAE architecture.

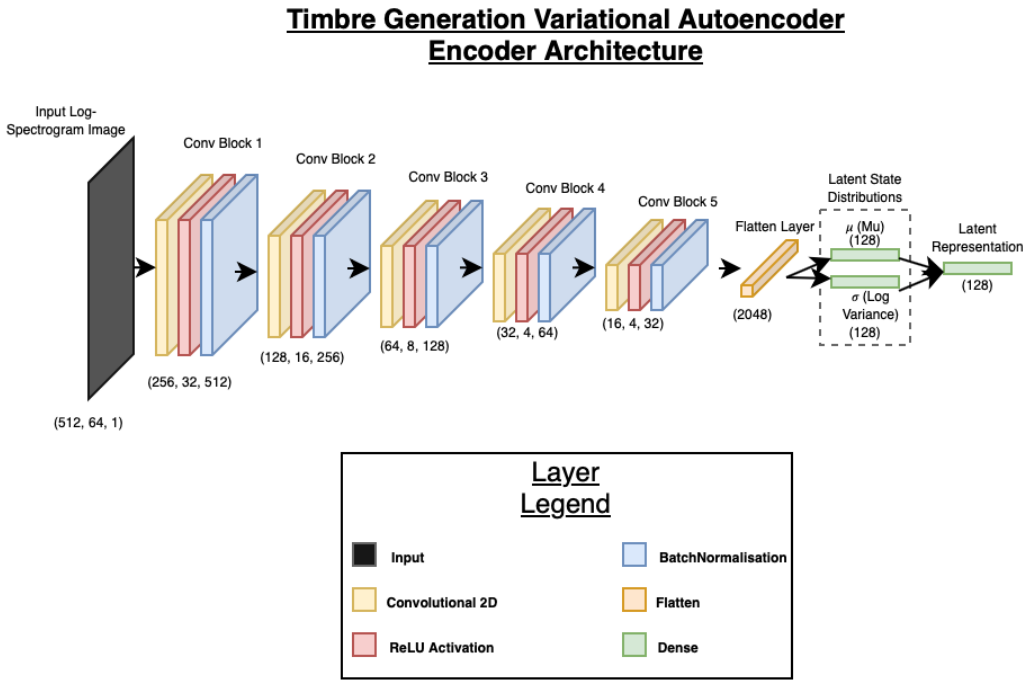


Figure 5.3: Encoder architecture.

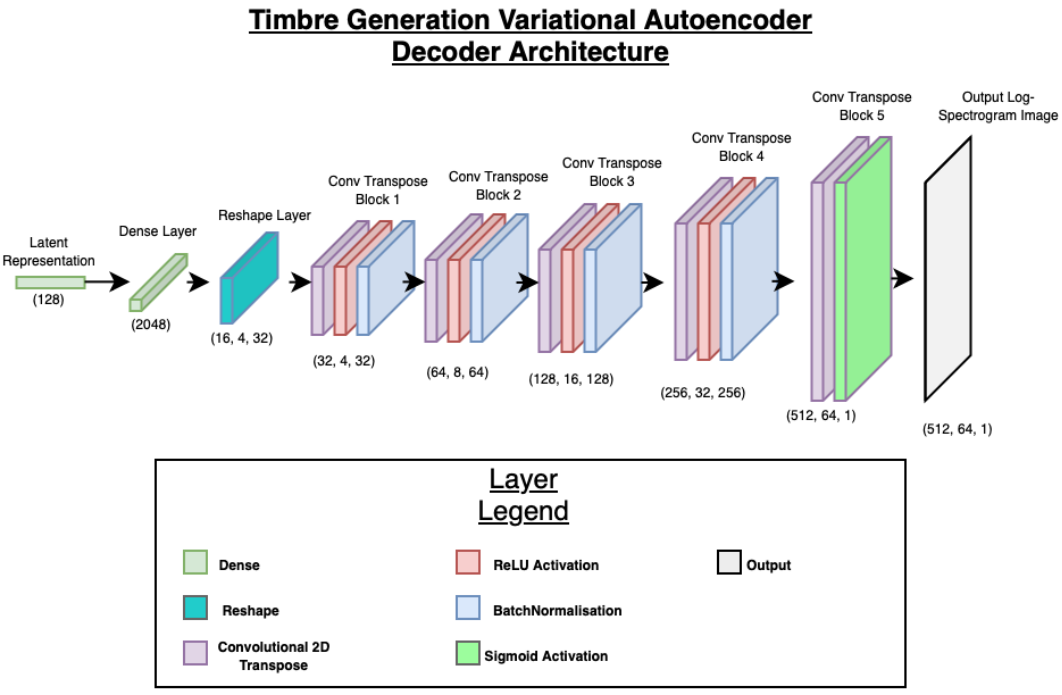


Figure 5.4: Decoder architecture.

```
Model: "encoder"
_________________________________________________________________________________________________
 Layer (type)                Output Shape              Param #   Connected to
=================================================================================================
 encoder_input (InputLayer)  [(None, 512, 64, 1)]      0         []

 conv2d (Conv2D)             (None, 256, 32, 512)      5120      ['encoder_input[0][0]']

 re_lu (ReLU)                (None, 256, 32, 512)      0         ['conv2d[0][0]']

 batch_normalization (Batch  (None, 256, 32, 512)      2048      ['re_lu[0][0]']
 Normalization)

 conv2d_1 (Conv2D)           (None, 128, 16, 256)      1179904   ['batch_normalization[0][0]']

 re_lu_1 (ReLU)              (None, 128, 16, 256)      0         ['conv2d_1[0][0]']

 batch_normalization_1 (Bat  (None, 128, 16, 256)      1024      ['re_lu_1[0][0]']
 chNormalization)

 conv2d_2 (Conv2D)           (None, 64, 8, 128)        295040    ['batch_normalization_1[0][0]'
                                                                 ]

 re_lu_2 (ReLU)              (None, 64, 8, 128)        0         ['conv2d_2[0][0]']

 batch_normalization_2 (Bat  (None, 64, 8, 128)        512       ['re_lu_2[0][0]']
 chNormalization)

 conv2d_3 (Conv2D)           (None, 32, 4, 64)         73792     ['batch_normalization_2[0][0]'
                                                                 ]

 re_lu_3 (ReLU)              (None, 32, 4, 64)         0         ['conv2d_3[0][0]']

 batch_normalization_3 (Bat  (None, 32, 4, 64)         256       ['re_lu_3[0][0]']
 chNormalization)

 conv2d_4 (Conv2D)           (None, 16, 4, 32)         18464     ['batch_normalization_3[0][0]'
                                                                 ]

 re_lu_4 (ReLU)              (None, 16, 4, 32)         0         ['conv2d_4[0][0]']

 batch_normalization_4 (Bat  (None, 16, 4, 32)         128       ['re_lu_4[0][0]']
 chNormalization)

 flatten (Flatten)           (None, 2048)              0         ['batch_normalization_4[0][0]'
                                                                 ]

 mu (Dense)                  (None, 128)               262272    ['flatten[0][0]']

 log_variance (Dense)        (None, 128)               262272    ['flatten[0][0]']

 encoder_output (Lambda)     (None, 128)               0         ['mu[0][0]',
                                                                  'log_variance[0][0]']

=================================================================================================
Total params: 2100832 (8.01 MB)
Trainable params: 2098848 (8.01 MB)
Non-trainable params: 1984 (7.75 KB)
_________________________________________________________________________________________________
```

```
Model: "decoder"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 decoder_input (InputLayer)  [(None, 128)]             0

 dense (Dense)               (None, 2048)              264192

 reshape (Reshape)           (None, 16, 4, 32)         0

 conv2d_transpose (Conv2DTr  (None, 32, 4, 32)         9248
 anspose)

 re_lu_5 (ReLU)              (None, 32, 4, 32)         0

 batch_normalization_5 (Bat  (None, 32, 4, 32)         128
 chNormalization)

 conv2d_transpose_1 (Conv2D  (None, 64, 8, 64)         18496
 Transpose)

 re_lu_6 (ReLU)              (None, 64, 8, 64)         0

 batch_normalization_6 (Bat  (None, 64, 8, 64)         256
 chNormalization)

 conv2d_transpose_2 (Conv2D  (None, 128, 16, 128)      73856
 Transpose)

 re_lu_7 (ReLU)              (None, 128, 16, 128)      0

 batch_normalization_7 (Bat  (None, 128, 16, 128)      512
 chNormalization)

 conv2d_transpose_3 (Conv2D  (None, 256, 32, 256)      295168
 Transpose)

 re_lu_8 (ReLU)              (None, 256, 32, 256)      0

 batch_normalization_8 (Bat  (None, 256, 32, 256)      1024
 chNormalization)

 conv2d_transpose_4 (Conv2D  (None, 512, 64, 1)        2305
 Transpose)

 decoder_output (Activation  (None, 512, 64, 1)        0
 )

=================================================================
Total params: 665185 (2.54 MB)
Trainable params: 664225 (2.53 MB)
Non-trainable params: 960 (3.75 KB)
_________________________________________________________________
```

```
Model: "timbre_generation_vae"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 encoder_input (InputLayer)  [(None, 512, 64, 1)]      0

 encoder (Functional)        (None, 128)               2100832

 decoder (Functional)        (None, 512, 64, 1)        665185

=================================================================
Total params: 2766017 (10.55 MB)
Trainable params: 2763073 (10.54 MB)
Non-trainable params: 2944 (11.50 KB)
_________________________________________________________________
```

Figure 5.5: VAE architecture provided by Keras' `model.summary()` [95].

A latent space dimension of 128 was chosen to balance audio generation quality with available compute resources. Other smaller latent space dimensions were considered, but 128 resulted in the highest audio quality appropriate for further experimentation. An inter-rater reliability experiment involving three anonymous raters with musical experience verified this, with all raters rating 128D VAE-generated guitar note audio samples as higher quality and more similar to real guitar note recordings than those with smaller dimensions.

5.2.4 Reconstructing sounds from the SemanticTimbreDataset

The timbre generation system allows users to generate monophonic guitar notes according to a specified timbre descriptor, timbre magnitude, and pitch from E4-D6. Valid timbre descriptors are equal to the 19 descriptors present in SemanticTimbreDataset, and valid timbre magnitudes are 0, 25, 50, 75, and 100.

Figure 5.6 illustrates an overview for generating a monophonic guitar sound with specified timbre conditions and pitch.

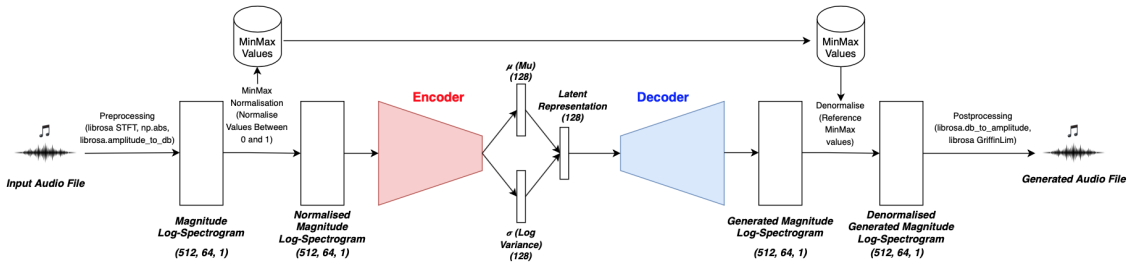


Figure 5.6: Sound generation/reconstruction with the timbre generation system.

When a user provides these specifications, the first step involves finding an example pre-processed log-magnitude spectrogram image from the SemanticTimbreDataset that matches the supplied specifications. This approach is feasible due to the comprehensive nature of the SemanticTimbreDataset for this task. Next, that corresponding spectrogram image is supplied to the VAE's encoder (see Figure 5.3), which encodes the image into a 128-dimensional latent representation.

The relevant latent representation is then supplied to the VAE's decoder (see Figure 5.4), which generates a new output log-magnitude spectrogram representing the specified sound. However, the decoder's output spectrograms are still min-max normalised with values between 0 and 1. Now, the stored min-max values for the initial spectrogram image are found within the external min-max values dictionary, and min-max denormalisation [145] is performed on the generated spectrogram. This denormalisation step ensures the correct dynamic range in the final generated audio file.

After denormalisation, Librosa's `db_to_amplitude` function [111] is applied to the log-magnitude spectrogram to obtain a magnitude spectrogram of the audio file. This prepares the spectrogram for the final step where the Griffin-Lim algorithm is applied to the magnitude spectrogram for phase reconstruction [61, 144], resulting in a final output 0.74-second audio file of the generated sound to specification.

5.3 Training the timbre generation system

Recall that a total of 1771 spectrograms were available for training, where each training spectrogram had the same dimensional shape of (512, 64, 1). All training spectrograms were loaded into NumPy arrays.

The VAE model was then configured with Keras' `model.compile()` method [96] to use an Adam optimizer [99] with a learning rate set to 0.0005. The loss that the model minimised during training was configured to be the combination of the reconstruction loss and the Kullback-Leibler divergence (see Section 2.2.2).

At this stage, the model was ready to be fit to the training data with Keras' `model.fit()` method [96], which starts the model's training loop. The `fit` method specified the model to train on a shuffled set of all training spectrograms during each epoch with a batch size of 64 for 300 epochs. For every training epoch, the VAE model's reconstruction loss, Kullback-Leibler divergence, and the combination of these losses on the training spectrograms were reported and saved. These metrics are known as the training metrics.

This whole training process results in one fully trained VAE model for timbre generation for all timbre descriptors. Figure 5.7 plots the training metrics obtained when training the final timbre generation VAE model. For the final epoch, the model's total combined loss was 3174, with a reconstruction loss of 2870 and a Kullback-Leibler divergence of 304.

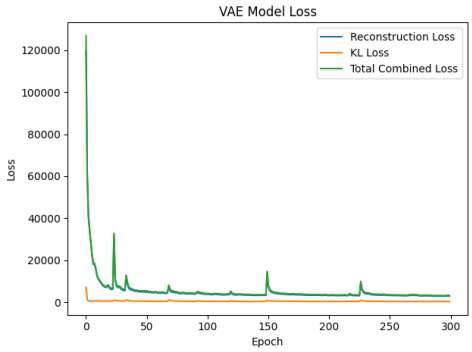
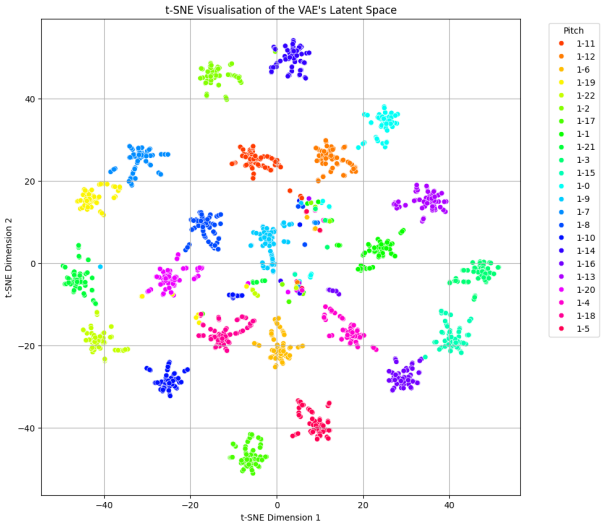
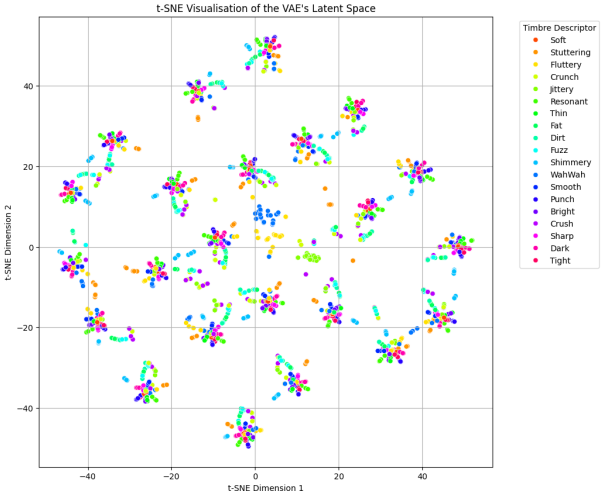


Figure 5.7: Timbre generation VAE's training metrics.

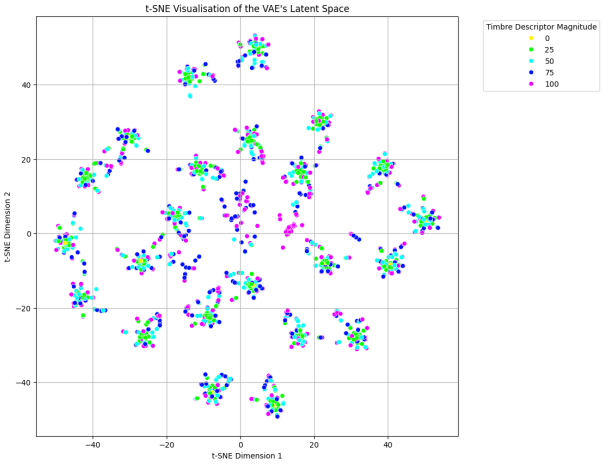
Figure 5.8 shows the timbre generation VAE's learned latent space when its encoder is supplied with all training data, where all latent representations are visualised in two dimensions via t-SNE dimensionality reduction [199].



(a)



(b)



(c)

Figure 5.8: VAE's latent space.

Looking at Figure 5.8a, it is clear that the encoder has firstly clustered the training data by pitch, and Figures 5.8b & 5.8c further reveal that sounds are organised in lines around the origin of each pitch cluster by timbre descriptor, where those sounds are positioned further from the origin as timbre magnitude increases. This structure makes sense because sounds differ more from each other as timbre magnitude increases, so the VAE naturally places these high-timbre magnitude sounds further from each cluster's origin and the cluster's other timbre descriptor lines due to their increased 'perceptual' distance. This also hints that interpolations between timbre descriptors and magnitudes within the same pitch cluster may result in sounds with higher resemblances to training data than interpolations between different pitch clusters.

5.4 Timbre interpolation

After training the timbre generation VAE, its interpolation capabilities were harnessed to perform timbre interpolation. When looking at the latent space of the trained timbre generation VAE, it is clear that the latent representations were organised in terms of similarity distance across pitch, timbre descriptor, and timbre magnitude. This means the VAE's latent space allows for controlled specifications of desired timbral qualities by supplying the VAE's encoder with new 'interpolated' latent representations that enable the VAE's decoder to generate novel spectrograms.

In the vision domain, Carter and Nielsen demonstrate linear interpolation in the latent space of a generative model trained to perform face image synthesis/manipulation to generate novel facial expressions between a neutral and smiling facial expression [23, 208]. Figure 5.9 shows the generative model's outputs from this linear interpolation.

Figure 5.9: Interpolation from neutral-smiling facial expression. Figure from [208].

Carter and Nielsen achieve this by sampling from equally spaced points between the start latent representation (the neutral facial expression) and the target latent representation (the smiling facial expression). This approach has been applied to VAE latent spaces [156, 9, 136, 31].

This idea is applied to the timbre generation task to achieve 'timbre interpolation', where timbral characteristics described by two of the SemanticTimbreDataset's timbre

descriptors can be merged together. After specifying the timbre descriptors, timbre magnitudes, and pitches for a start sound and a target sound, the first step to achieving an interpolation between them is to mimic the process for reconstructing sounds discussed in Section 5.2.4 by selecting closely matching examples for both start and target sounds from the SemanticTimbreDataset. This supplies the encoder with corresponding spectrograms from which the two relevant latent representations for the start and end points of the interpolation are obtained.

Now, users can specify how many interpolation points should be generated for an interpolation (including the start and target data points), from a minimum of 3 to a maximum of 10. Figure 5.10 shows an example of what occurs within the VAE's latent space when a user requests a 5-point interpolation between a 100%-Fuzzy E4 electric guitar note and a 50%-Soft E4 electric guitar note. Figure 5.11 shows the linear interpolation algorithm used to return the interpolated latent representations for a specified number of interpolation points ($\boldsymbol{n}$).

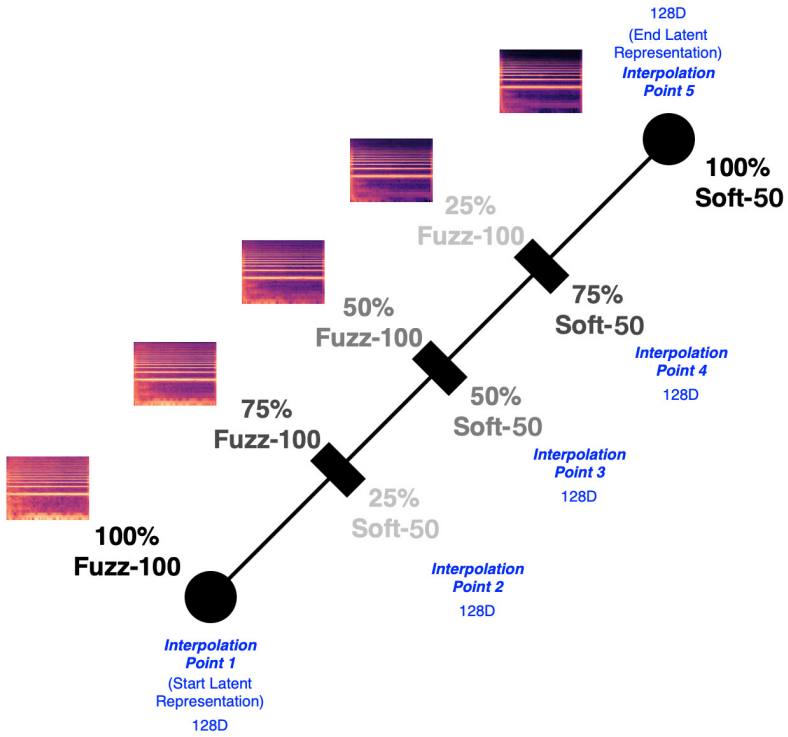


Figure 5.10: Timbre interpolation in the latent space.

```
def perform_linear_interpolation(start_v, target_v, n):
    ratios = np.linspace(0, 1, num=n)
    interpolated_latent_representations = list()
    for ratio in ratios:
        v = (1.0 - ratio) * start_v + ratio * target_v
        interpolated_latent_representations.append(v)
    return np.asarray(interpolated_latent_representations)
```

Figure 5.11: Linear interpolation algorithm.

After using the linear interpolation algorithm to return the desired number of interpolated latent representations, an almost identical process to the one specified in Section 5.2.4 is used to generate audio files for the start sound, target sound, and all the desired interpolated sounds in between. The key difference to note when generating audio from the interpolated spectrograms is that a linear interpolation between the min-max values for the start and target sounds is performed. This results in the min-max values for the interpolated sounds also following a smooth transition from the start sound's characteristics to the target sound's characteristics, helping to promote appropriate dynamic ranges for the interpolated sounds when the denormalisation and `librosa.db_to_amplitude` post-processing steps are applied. Figure 5.12 shows an overview of the timbre interpolation procedure between two specified sounds within the timbre generation system.

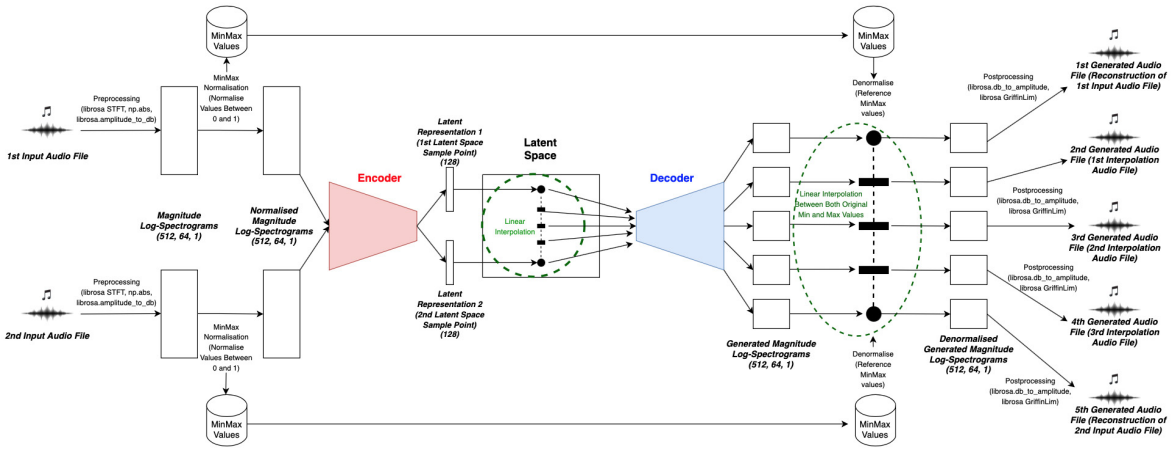


Figure 5.12: Timbre interpolation with the timbre generation system.

Chapter 6

Evaluation & discussion

This chapter details the procedures and experiments used to evaluate the timbre recognition system, the timbre generation system, and its interpolation capabilities. The results of these evaluation procedures and their experiments are also presented and discussed.

6.1 Evaluating the timbre recognition system

To thoroughly evaluate the timbre recognition system (see Chapter 4), it was necessary to test all 19 timbre recognition models on audio files of monophonic guitar notes originating from a different electric guitar model than the Fender Stratocaster used for the SemanticTimbreDataset. This evaluation is crucial to observe whether the timbre recognition models can predict timbre magnitudes for the SemanticTimbreDataset's 19 timbre descriptors on different sounds and 'transfer' their learning capabilities to a different electric guitar. It was decided to evaluate the timbre recognition models using monophonic guitar notes from a Gibson Les Paul electric guitar. This decision was made to prioritise and maximise this evaluation's external validity because the Gibson Les Paul is widely considered the next most influential and popular electric guitar model worldwide after the Fender Stratocaster [152, 67, 139, 76, 187].

6.1.1 Creating the Gibson Les Paul semantic timbre test dataset

The Les Paul electric guitar sounds that make up this new evaluation dataset were obtained from the Legacy Guitar VST plugin [79] found in Logic Pro [81]. Clean, untouched recordings of monophonic notes from pitches E2-D#5 from these Gibson samples were collected by triggering MIDI notes lasting the same 5 seconds as the Stratocaster recordings within Logic Pro. The pitches E2-D#5 were recorded rather than pitches E2-D6 as used for the SemanticTimbreDataset to facilitate evenly numbered quantities of notes in the low register (E2-D#3), medium register (E3-D#4), and high register (E4-D#5) that would

provide a fair evaluation for experiments that investigate the effect of monophonic note pitch on timbre recognition.

The same Guitar Rig [83] pedal parameters shown in Table 3.8 used to create the SemanticTimbreDataset were applied to the Gibson recordings to create the diverse timbral characteristics for the rest of this evaluation dataset.

6.1.2 Experimental procedure for evaluating timbre recognition

Two sets of experiments were conceived to evaluate the timbre recognition system. The first set of 19 experiments focused on measuring the same test metrics calculated after each of the 19 timbre recognition models' training procedures (see Section 4.3) but on the newly acquired test dataset of Gibson Les Paul monophonic guitar notes. Each experiment tests a timbre recognition model's ability to predict the magnitude of the model's corresponding timbre descriptor on the subset of Gibson Les Paul test sounds that represent the relevant timbre descriptor. Furthermore, the Les Paul notes for each timbre descriptor subgroup were also separated into three registers of pitch: low (E2-D#3), medium (E3-D#4), and high (E4-D#5). The second set of experiments replicated the first, except only distinct pitch register subgroups (low, medium, high) were selected as test data for the timbre recognition models. This pitch register separation offers additional insight into the effect of a note's pitch register on the timbre recognition models' ability to predict the 20 timbre descriptors' magnitudes accurately. The second set of experiments was motivated by previous work presenting conflicting findings on the relationship between humans' timbre and pitch perception [155, 128, 169, 176, 125], particularly between pitch and the brightness/warmth timbre dimension [29, 33].

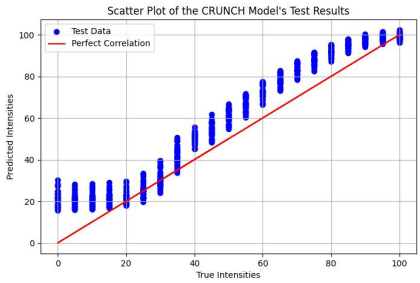
6.1.3 Timbre recognition evaluation results & discussion

The calculated test metric results for the first set of 19 experiments across all the timbre recognition models can be seen in Table 6.1, and the scatter plots of the true and predicted timbre descriptor magnitudes for each timbre recognition model's experiment can be seen in Figures 6.1-6.4.

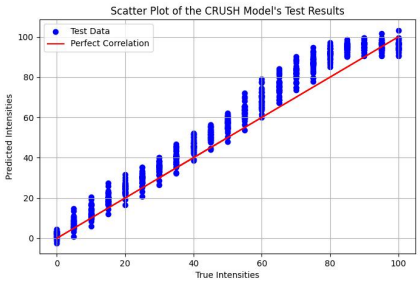
Looking at Table 6.1, all timbre recognition models except one predict the timbre magnitudes on Les Paul notes with similar patterns to their predictions on the Stratocaster training test set shown in Table 4.1, except the Les Paul predictions result in slightly higher RMSE and lower $\boldsymbol{R^2}$ results. This correlation shows that the timbre recognition models can transfer their learnt capabilities to another electric guitar, even if the predictions for different guitars are slightly more widespread. The 'fat' model's RMSE and $\boldsymbol{R^2}$ score are surprisingly inaccurate, but then observing Figure 6.2c reveals that the 'fat' model does indeed correctly predict the increasing timbre magnitudes; however, its predictions

Table 6.1: Timbre recognition models' evaluation metrics (all pitches).

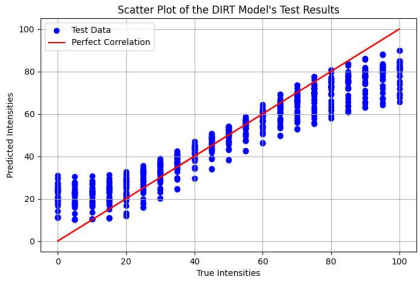
Timbre Descriptor	Test RMSE	Test R^2 Score	Pearson Correlation Coefficient (p-value)
Bright	21.902	0.477	0.864 (p: < 0.001)
Crunch	11.091	0.866	0.979 (p: < 0.001)
Crush	7.822	0.933	0.987 (p: < 0.001)
Dark	27.623	0.168	0.771 (p: < 0.001)
Dirt	11.192	0.863	0.962 (p: < 0.001)
Fat	37.075	-0.500	0.911 (p: < 0.001)
Flutter	5.893	0.962	0.991 (p: < 0.001)
Fuzz	7.068	0.946	0.982 (p: < 0.001)
Jitter	7.788	0.934	0.967 (p: < 0.001)
Punch	23.219	0.412	0.746 (p: < 0.001)
Resonant	6.905	0.948	0.990 (p: < 0.001)
Sharp	29.732	0.036	0.213 (p: < 0.001)
Shimmer	20.073	0.560	0.922 (p: < 0.001)
Smooth	27.897	0.151	0.597 (p: < 0.001)
Soft	27.265	0.189	0.534 (p: < 0.001)
Stutter	7.705	0.935	0.984 (p: < 0.001)
Thin	16.507	0.703	0.856 (p: < 0.001)
Tight	14.066	0.784	0.921 (p: < 0.001)
WahWah	7.567	0.938	0.983 (p: < 0.001)



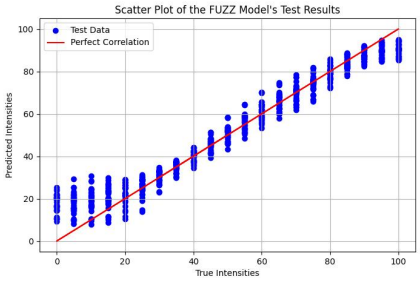
(a) Crunch



(b) Crush

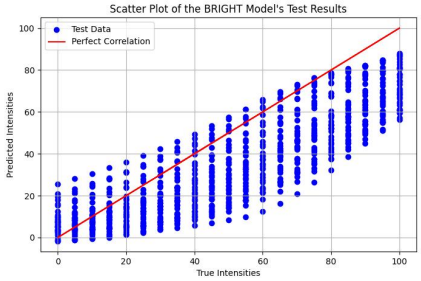


(c) Dirt

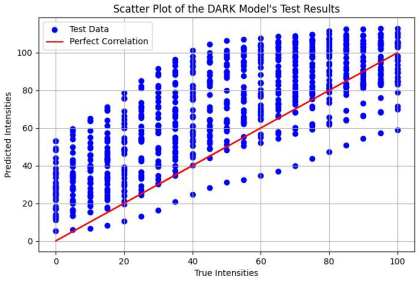


(d) Fuzz

Figure 6.1: DistortionFX models' timbre magnitude predictions.

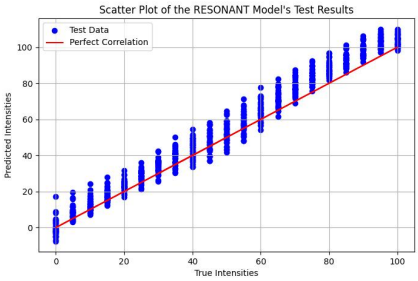


(a) Bright

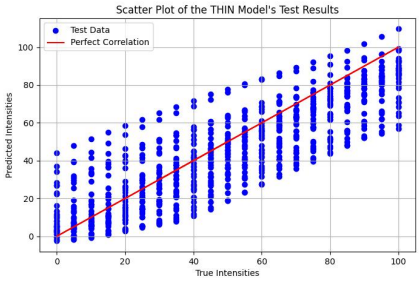


(b) Dark

(c) Fat

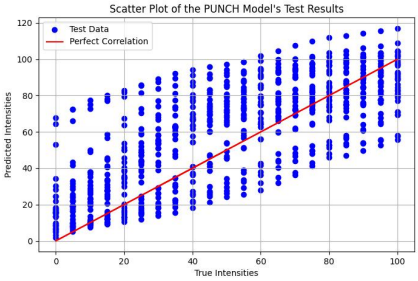


(d) Resonant

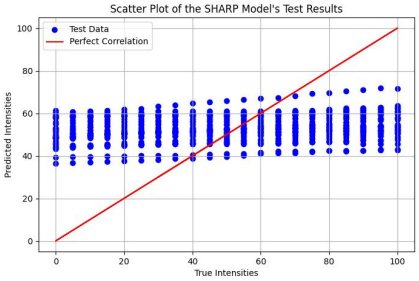


(e) Thin

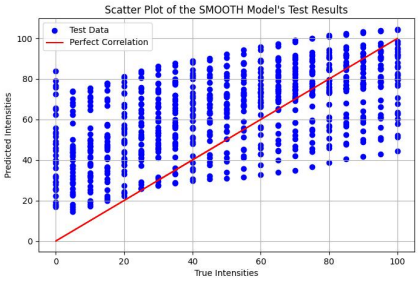
Figure 6.2: FilterFX models' timbre magnitude predictions.



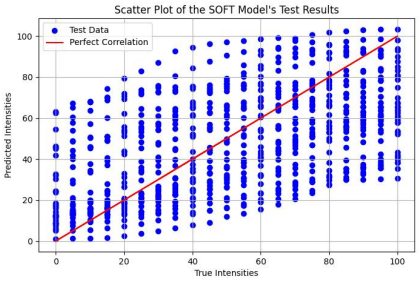
(a) Punch



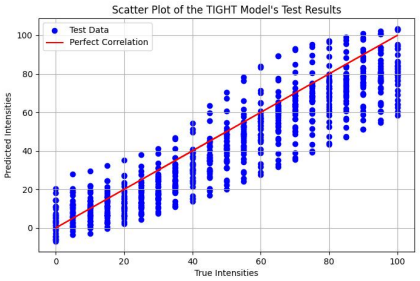
(b) Sharp



(c) Smooth

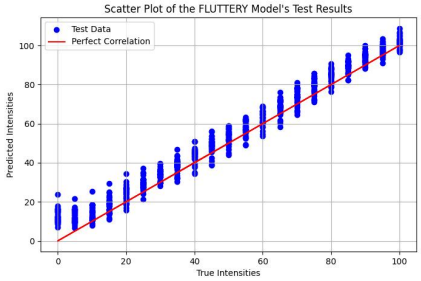


(d) Soft

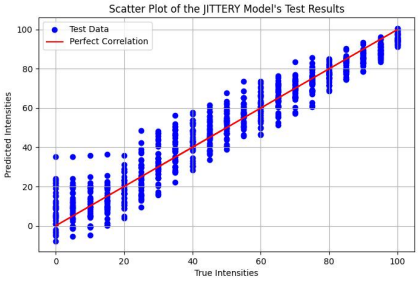


(e) Tight

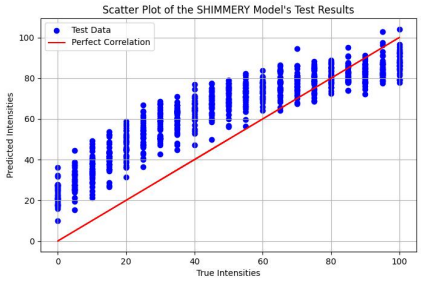
Figure 6.3: DynamicsFX models' timbre magnitude predictions.



(a) Flutter

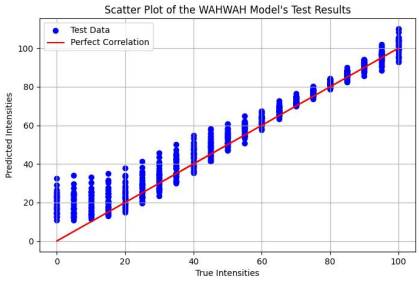


(b) Jitter



(c) Shimmer

(d) Stutter



(e) WahWah

Figure 6.4: OscillationFX models' timbre magnitude predictions.

are shifted higher than the expected value range between 0-100. Its Pearson correlation coefficient further confirms this. This phenomenon may be occurring due to the differing harmonics of the Gibson Les Paul versus the Fender Stratocaster, and the saturation effect applied to 'fatten' sounds may exaggerate these differences further. Therefore, these results indicate that a 'fat' sound for a given guitar may be quite specific to its harmonics, hinting that there are no general 'fat' guitar sounds, but rather, each guitar has its definition for what a fat sound is.

Interestingly, when observing Tables 6.1 & 4.1 and Figures 6.2a & 6.2b, the 'bright' and 'dark' timbre recognition models generally predicted the Les Paul notes to be less bright and darker than the Stratocaster notes. These results indicate that the timbre recognition system estimates the Les Paul to be a darker guitar than the Stratocaster, which agrees with the thoughts of many guitarists and guitar enthusiasts [36, 49, 186, 172, 149]. This shows that the timbre recognition models can transfer their abilities to other guitars to the point where they can highlight timbral differences between the Fender Stratocaster and Les Paul (or other) guitars.

For the second set of experiments, each model's test results for notes in the low, medium, and high pitch registers can be seen in Tables 6.2-6.5.

Table 6.2: DistortionFX models' evaluation metrics.

Timbre Descriptor	Pitch Register	RMSE	$\boldsymbol{R^2}$ Score	Pearson Corr Coefficient (p-value)
Crunch	Low	9.708	0.897	0.980 (p: < 0.001)
	Medium	11.250	0.862	0.980 (p: < 0.001)
	High	12.174	0.838	0.981 (p: < 0.001)
Crush	Low	6.947	0.947	0.992 (p: < 0.001)
	Medium	7.167	0.944	0.987 (p: < 0.001)
	High	9.161	0.908	0.985 (p: < 0.001)
Dirt	Low	13.116	0.812	0.964 (p: < 0.001)
	Medium	9.375	0.904	0.985 (p: < 0.001)
	High	10.764	0.874	0.981 (p: < 0.001)
Fuzz	Low	6.350	0.956	0.987 (p: < 0.001)
	Medium	6.912	0.948	0.987 (p: < 0.001)
	High	7.860	0.933	0.986 (p: < 0.001)

There were not notable differences across pitch registers for the DistortionFX and OscillationFX timbre descriptor recognition models. However, for the FilterFX timbre descriptor recognition models, there were significant differences in RMSE and $\boldsymbol{R^2}$ scores across pitch registers, with the 'bright', 'dark', 'fat', and 'resonant' models getting worse as the pitch register got higher. Nevertheless, their Pearson correlation coefficients remained similar, which indicates that the models perceived these timbral characteristics differently across pitch registers but could still model the timbre magnitude correlation. A

Table 6.3: FilterFX models' evaluation metrics.

Timbre Descriptor	Pitch Register	RMSE	R^2 Score	Pearson Corr Coefficient (p-value)
Bright	Low	13.729	0.794	0.914 (p: < 0.001)
	Medium	20.832	0.527	0.921 (p: < 0.001)
	High	28.577	0.109	0.941 (p: < 0.001)
Dark	Low	16.392	0.707	0.860 (p: < 0.001)
	Medium	24.208	0.361	0.874 (p: < 0.001)
	High	37.873	-0.565	0.888 (p: < 0.001)
Fat	Low	28.123	0.137	0.952 (p: < 0.001)
	Medium	37.448	-0.530	0.952 (p: < 0.001)
	High	43.937	-1.106	0.966 (p: < 0.001)
Resonant	Low	4.959	0.973	0.992 (p: < 0.001)
	Medium	6.854	0.950	0.989 (p: < 0.001)
	High	8.454	0.922	0.994 (p: < 0.001)
Thin	Low	17.928	0.649	0.806 (p: < 0.001)
	Medium	11.504	0.856	0.927 (p: < 0.001)
	High	19.070	0.603	0.941 (p: < 0.001)

Table 6.4: DynamicsFX models' evaluation metrics.

Timbre Descriptor	Pitch Register	RMSE	R^2 Score	Pearson Corr Coefficient (p-value)
Punch	Low	32.566	-0.157	0.833 (p: < 0.001)
	Medium	15.413	0.741	0.914 (p: < 0.001)
	High	17.868	0.652	0.855 (p: < 0.001)
Sharp	Low	28.905	0.089	0.362 (p: < 0.001)
	Medium	29.987	0.019	0.193 (p: < 0.001)
	High	30.286	-0.001	0.209 (p: < 0.001)
Smooth	Low	21.365	0.502	0.744 (p: < 0.001)
	Medium	27.970	0.147	0.804 (p: < 0.001)
	High	33.105	-0.196	0.688 (p: < 0.001)
Soft	Low	27.337	0.185	0.714 (p: < 0.001)
	Medium	23.649	0.390	0.839 (p: < 0.001)
	High	30.391	-0.008	0.394 (p: < 0.001)
Tight	Low	12.185	0.838	0.968 (p: < 0.001)
	Medium	11.308	0.861	0.933 (p: < 0.001)
	High	17.810	0.654	0.881 (p: < 0.001)

Table 6.5: OscillationFX models' evaluation metrics.

Timbre Descriptor	Pitch Register	RMSE	R^2 Score	Pearson Corr Coefficient (p-value)
Flutter	Low	6.376	0.956	0.993 (p: < 0.001)
	Medium	5.120	0.971	0.992 (p: < 0.001)
	High	6.109	0.959	0.992 (p: < 0.001)
Jitter	Low	9.842	0.894	0.968 (p: < 0.001)
	Medium	7.020	0.946	0.978 (p: < 0.001)
	High	5.984	0.961	0.981 (p: < 0.001)
Shimmer	Low	19.964	0.565	0.923 (p: < 0.001)
	Medium	18.591	0.623	0.937 (p: < 0.001)
	High	21.553	0.493	0.912 (p: < 0.001)
Stutter	Low	6.513	0.954	0.980 (p: < 0.001)
	Medium	5.924	0.962	0.995 (p: < 0.001)
	High	10.029	0.890	0.991 (p: < 0.001)
WahWah	Low	6.140	0.959	0.986 (p: < 0.001)
	Medium	5.934	0.962	0.988 (p: < 0.001)
	High	9.943	0.892	0.984 (p: < 0.001)

similar pattern occurs with the 'punch' and 'smooth' models. Reymore et al. found that various semantic timbre descriptors used by participants for timbre perception had strong positive and negative correlations with pitch register [155], and these experimental results, particularly the 'bright' and 'dark' results, empirically reinforce their findings.

6.2 Evaluating the timbre generation system

The evaluation of the timbre generation system concerns the ability of the timbre generation VAE (see Chapter 5) to generate an acoustically pleasing and realistic monophonic guitar note sound to a specified timbre descriptor, timbre magnitude, and pitch. Since the SemanticTimbreDataset already contains guitar sounds to all possible specifications, this evaluation could also test the VAE's ability to accurately *reconstruct* guitar sounds to resemble the sounds in the dataset. This evaluation was split into two components: an objective evaluation via regression, and a subjective perceptual evaluation with human participants. Generative models can be evaluated objectively and subjectively due to the content they produce [151].

6.2.1 Generating test sounds for evaluation

The VAE was prompted to generate a single guitar note in the pitch of E4 for every distinct timbre descriptor category within the SemanticTimbreDataset at timbre magnitudes of 25%,

50%, 75%, and 100% for each descriptor (except 'Clean', which was generated once with 0% timbre magnitude). This resulting dataset was called the '**VAE-GeneratedTestSet**' [21] and was used for both the objective and perceptual evaluation components. Selected examples from the VAE-GeneratedTestSet are visualised as spectrograms in Appendix C. The single E4 pitch was selected for all notes to ensure consistency and minimise bias across pitch and loudness when human participants listened to sounds for the perceptual evaluation. E4 is also mid-way through the pitch register for electric guitars, so its choice over other pitches further minimised bias towards very low or high pitches.

6.2.2 Experimental procedures for evaluating timbre generation

6.2.2.1 Objective evaluation via regression

The objective evaluation component was facilitated by re-training the same timbre recognition regression models described in Chapter 4 to recognise timbre from 0.74-second time windows of audio to match the length of audio files the timbre generation system generates. This approach enabled a neutral, objective perspective to evaluate the timbre generation system's capability to reconstruct guitar notes that accurately incorporate the various timbre characteristics described by each timbre descriptor within the SemanticTimbreDataset and their respective timbre magnitudes. Furthermore, the same regression test metrics used in Section 4.3 were again used to measure the recognition models' performance for predicting timbre magnitudes for generated sounds from each timbre descriptor category. Using the same regression test metrics and timbre recognition models as before also permits feasibility for direct comparison to the timbre recognition's results on the original data, giving insight into the VAE's reconstruction capabilities.

6.2.2.2 Perceptual evaluation

The goal for the perceptual evaluation was to determine the timbre generation VAE's capability to reconstruct perceptually accurate guitar sounds to the same timbral specification as mentioned before. Twenty human participants from the University of Cambridge community were recruited to judge the VAE-generated guitar sounds in an ethically reviewed and approved experiment. 18/20 of these participants regularly play musical instruments, and 8/20 play the electric guitar. For each of the 77 sounds in the VAE-GeneratedTestSet, participants were asked to listen to the original corresponding sound in the SemanticTimbreDataset followed by a 3-second pause and then the generated test sound. Immediately following each pair of sounds, participants were asked to rate the second VAE-generated sound based on Figure 6.5's criteria.

The participant ratings between 1-5 for each generated sound were then averaged to produce a mean opinion score (MOS) for each VAE-generated sound's reconstruction

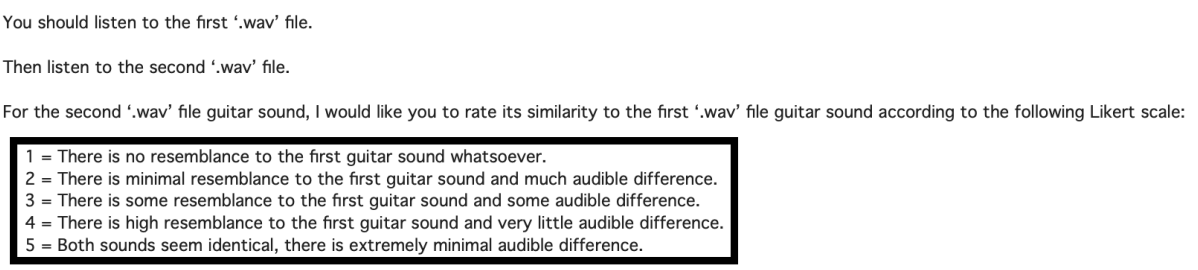


Figure 6.5: Participant criteria.

quality and similarity. Many studies use the MOS metric on a 1-5 scale to qualitatively evaluate their generative models'/systems' reconstruction quality [39, 105, 32, 19] and this study follows suit. This perceptual evaluation is necessary since timbre is primarily a perceptual and qualitative characteristic of sound that is best judged via human perception.

6.2.3 Timbre generation evaluation results

6.2.3.1 Objective evaluation via regression results & discussion

The calculated regression test metric results for the set of 19 experiments across all timbre recognition models performed on the corresponding timbre descriptor groups within the VAE-GeneratedTestSet can be seen in Table 6.6.

Table 6.6: Timbre recognition models' evaluation metrics on VAE-GeneratedTestSet.

Timbre Descriptor	Test RMSE	Test $\boldsymbol{R^2}$ Score	Pearson Correlation Coefficient (p-value)
Bright	10.762	0.883	0.927 (p: < 0.001)
Crunch	7.895	0.936	0.964 (p: < 0.001)
Crush	6.263	0.952	0.977 (p: < 0.001)
Dark	16.519	0.698	0.858 (p: < 0.001)
Dirt	8.012	0.932	0.967 (p: < 0.001)
Fat	12.517	0.824	0.931 (p: < 0.001)
Flutter	14.824	0.759	0.842 (p: < 0.001)
Fuzz	5.939	0.958	0.981 (p: < 0.001)
Jitter	11.813	0.849	0.926 (p: < 0.001)
Punch	14.573	0.771	0.884 (p: < 0.001)
Resonant	5.752	0.967	0.986 (p: < 0.001)
Sharp	28.128	0.136	0.456 (p: < 0.001)
Shimmer	9.281	0.902	0.949 (p: < 0.001)
Smooth	15.342	0.733	0.878 (p: < 0.001)
Soft	14.179	0.785	0.887 (p: < 0.001)
Stutter	20.123	0.555	0.783 (p: < 0.001)
Thin	7.145	0.942	0.974 (p: < 0.001)
Tight	18.627	0.621	0.732 (p: < 0.001)
WahWah	21.791	0.497	0.903 (p: < 0.001)

Comparing the results from Table 6.6 to Table 4.1, it is clear that most models have similar metrics values, which indicates the timbre generation VAE can reconstruct sounds the timbre recognition models perceive to be similar to the original sounds in the SemanticTimbreDataset, providing evidence for strong reconstruction. However, the 'flutter', 'stutter', 'tight', and 'wahwah' models have drastically worse metrics on the VAE-generated sounds than the original SemanticTimbreDataset sounds, hinting that the VAE struggled to reconstruct sounds to embody these timbral characteristics accurately. Intriguingly, the timbral characteristics described by these terms all make nuanced changes to audio in quick time-frames, which results in their corresponding spectrograms containing small, high-frequency pockets of crucial information, as seen in Appendix B. VAEs are known to generate blurry output images compared to GANs which are known to produce sharper clearer images [44, 98, 73, 203, 146, 130]. Given this evidence, it is possible that the timbre-generation VAE could not accurately generate crisp spectrograms containing the necessary high-frequency content required to reconstruct sounds with 'stutter' or 'wahwah', and the VAE's generated spectrograms for these descriptors in Appendix C indicate this was the case.

6.2.3.2 Perceptual evaluation results & discussion

Table 6.7 shows the human participants' mean opinion scores for each guitar sound's reconstruction quality and similarity in the VAE-GeneratedTestSet compared to the corresponding original sound in the SemanticTimbreDataset.

Participants generally rated most of the VAE-reconstructed sounds as very similar to the original sounds with high-quality reconstructions, as indicated by many mean opinion scores between 4-5, confirming that most sounds were perceptually reconstructed with high accuracy. Strikingly, the four weakest timbre descriptor reconstructions with average mean opinion scores below 4 were the same timbre descriptors highlighted in the adjacent objective evaluation (tight, flutter, stutter, and wahwah). This points towards the previously mentioned issue of VAEs' tendencies to produce blurry images [44, 98, 73, 203, 146, 130] carrying over spectrogram inaccuracies into human auditory perception.

6.3 Evaluating timbre interpolation

This section aims to evaluate the VAE's ability to correctly 'merge' timbre characteristics that can be described by two timbre descriptors in the SemanticTimbreDataset from both objective and perceptual perspectives. Similar to the timbre generation evaluation, this evaluation aimed to be comprehensive, so it was split into two components: an objective evaluation via classification and a subjective perceptual evaluation with human

Table 6.7: Mean opinion scores for the timbre generation system's reconstructions of E4 notes.

Timbre Group	Timbre Descriptor	Timbre Magnitude 25	50	75	100	Average MOS
DistortionFX	Clean	N/A	N/A	N/A	4.65	4.65
	Crunch	4.15	3.90	4.40	4.55	4.25
	Crush	4.70	4.75	4.15	4.15	4.44
	Dirt	4.55	3.80	4.05	4.15	4.14
	Fuzz	4.75	4.75	4.65	4.70	4.71
FilterFX	Bright	4.85	4.50	4.30	4.10	4.44
	Dark	4.25	4.00	4.20	4.90	4.34
	Fat	4.10	3.45	4.35	4.30	4.05
	Resonant	4.75	4.95	4.80	4.80	4.83
	Thin	4.80	4.90	4.90	4.85	4.86
DynamicsFX	Punch	4.65	4.30	4.15	4.85	4.49
	Sharp	4.55	4.50	3.75	3.80	4.15
	Smooth	4.00	4.45	4.15	4.20	4.20
	Soft	4.55	4.70	4.85	4.85	4.74
	Tight	3.10	2.90	2.90	2.75	2.91
OscillationFX	Flutter	4.75	2.90	2.70	2.25	3.15
	Jitter	4.80	4.25	4.40	3.55	4.25
	Shimmer	5.00	5.00	5.00	4.85	4.96
	Stutter	3.05	2.90	2.80	2.20	2.74
	WahWah	4.00	2.85	2.65	2.30	2.95

participants. Again, this was done due to timbre's qualitative and quantitative nature, and previously benchmarked evaluation procedures [151].

6.3.1 Generating timbre interpolation data for evaluation

The VAE was prompted to generate monophonic guitar notes in the pitch of E4 via a 5-point linear interpolation of its latent space between two timbre descriptors specified to have timbre magnitudes of 100 in the following scenarios:

- **Scenario 1**: The top-2 performing timbre descriptor reconstructions from each timbre group in the perceptual evaluation detailed in Section 6.2.2.2. Table 6.7 shows that the top-2 timbre descriptors for each timbre group are: DistortionFX - 'Clean' & 'Fuzz', FilterFX - 'Resonant' & 'Thin', DynamicsFX - 'Punch' & 'Soft', OscillationFX - 'Jitter' & 'Shimmer'. This results in 4 interpolations (Clean-Fuzz, Resonant-Thin, Punch-Soft, and Shimmer-Jitter).
- **Scenario 2**: All pair combinations of the top performing timbre descriptor reconstructions from all timbre groups in the perceptual evaluation detailed in Section 6.2.2.2. Table 6.7 shows that the top timbre descriptors for each timbre group are: DistortionFX - 'Fuzz', FilterFX - 'Thin', DynamicsFX - 'Soft', OscillationFX - 'Shimmer'. This results in 6 interpolations (Fuzz-Thin, Fuzz-Soft, Fuzz-Shimmer, Thin-Soft, Thin-Shimmer, and Soft-Shimmer).

A fixed single pitch of E4 for all interpolations was chosen for the same reasons explained in Section 6.2.1. The timbre magnitude for each relevant timbre descriptor was fixed to 100 to ensure a reasonable number of judgments for human participants during the perceptual experiment while still obtaining critical insights into how the VAE models timbre throughout interpolations between timbre descriptors. Including all possible combinations of all timbre magnitudes would significantly increase the number of judgments required, exposing human participants to listening fatigue [7, 129] given the scope of this project's allocated timeline. For similar reasons, it was decided to perform 5-point interpolations, resulting in 5 audio samples for each interpolation between a pair of descriptors. A 5-point interpolation still provides plenty of granular insight into the interpolation between each pair of descriptors while keeping the number of experimental judgments for participants reasonable. This resulting dataset was called the '**VAE-InterpolatedTestSet**' [21] and was used for the objective and perceptual evaluation components. Appendix D contains spectrograms of VAE-InterpolatedTestSet's audio.

6.3.2 Experimental procedure for the objective evaluation via classification

A CNN classifier was trained on audio samples from each relevant timbre descriptor with timbre magnitudes of 100 from the SemanticTimbreDataset to perform multi-class classification and classify which timbre descriptor best describes a given guitar sound. This classifier was implemented in Keras [25], and its architecture and training metrics can be seen in Appendix E. When evaluating the accuracy of a given timbre interpolation, the goal was to determine whether specific timbral characteristics that correspond to the interpolation's relevant timbre descriptors are present in a given sound, and to what degree throughout the generated audio samples (points) of the interpolation. The probabilities supplied by the `softmax` activation function [92] from the output layer of the trained multi-class classifier aptly convey how confident the classifier is for classifying these timbral characteristics in each file, which elegantly fulfils this evaluation's goal. Hence, for every interpolation present in the VAE-InterpolatedTestSet, the classifier's probabilities for each of the two relevant timbre descriptor classes for each of the five audio samples was noted as an objective measure of the VAE's ability to interpolate between sounds described by two timbre descriptors.

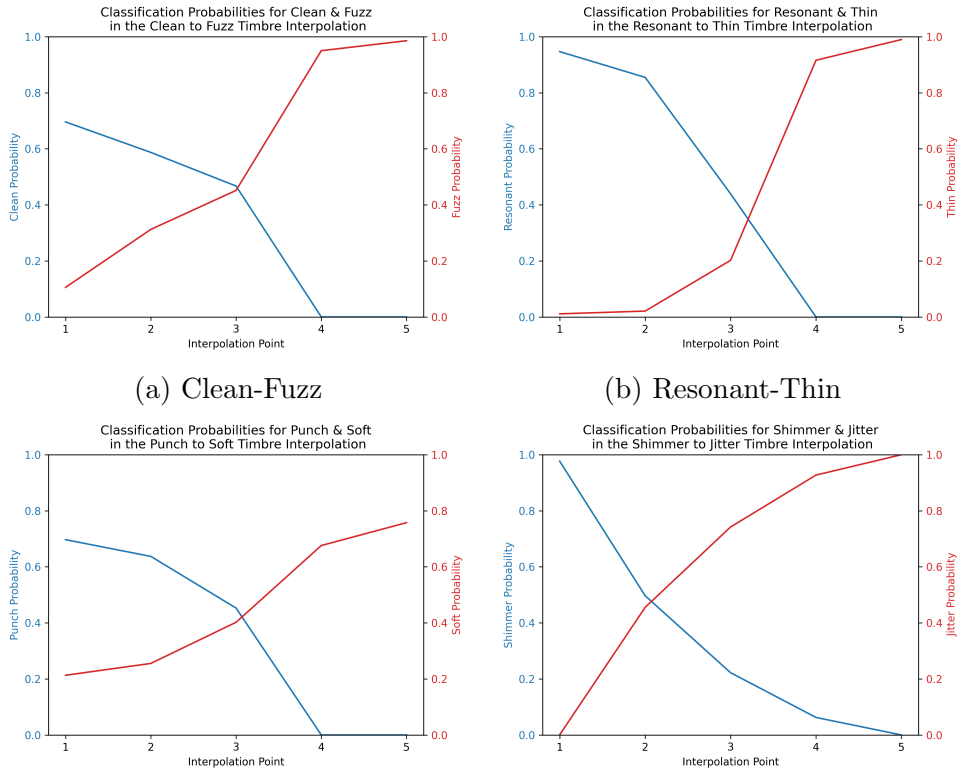
6.3.3 Objective evaluation via classification results & discussion

Figure 6.6 shows the classifier's probabilities for interpolations between the timbre descriptor pairs obtained via scenario 1. Figure 6.7 shows the classifier's probabilities for interpolations between the timbre descriptor pairs derived via scenario 2.

All ten interpolations display the classifier's probabilities for classifying the start and target timbre descriptors crossing between interpolation points 2-4. This objectively confirms that the classifier correctly predicts the start timbre for interpolation point 1 and target timbre for interpolation point 5, and its `softmax` probabilities decrease for interpolation points 2-4 accordingly as the timbral characteristics merge into an unknown unseen class. Some interpolations, such as those in Figures 6.6a, 6.6c, 6.7a, 6.7c, and 6.7e, display the probability crossover closer to interpolation point 3, which displays a more definitive judgment for the expected proportions of each timbre descriptor throughout the interpolation.

6.3.4 Experimental procedure for the perceptual evaluation

This evaluation aims to verify whether the VAE can seamlessly blend and balance the specified start/target timbre descriptors in the interpolated outputs according to their positions along the linear interpolation trajectory. Doing so ensures that each generated



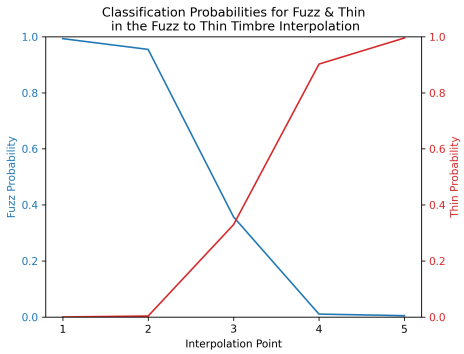
(a) Clean-Fuzz

(b) Resonant-Thin

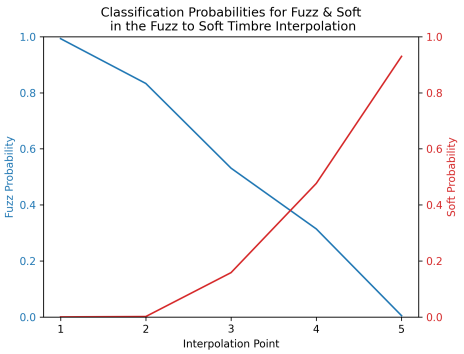
(c) Punch-Soft

(d) Shimmer-Jitter

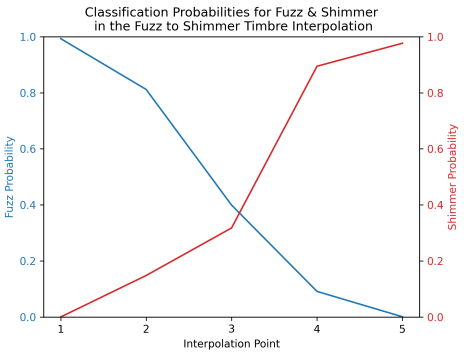
Figure 6.6: Classification probabilities for interpolations between the top 2 descriptors for each timbre group.



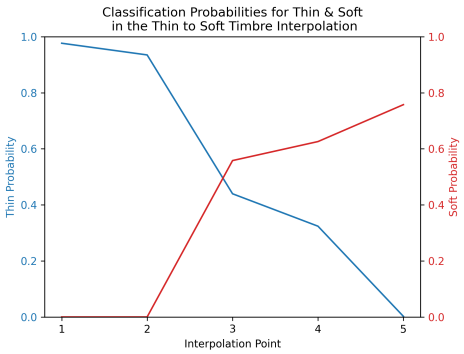
(a) Fuzz-Thin



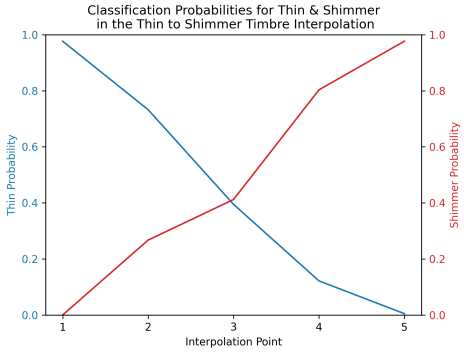
(b) Fuzz-Soft



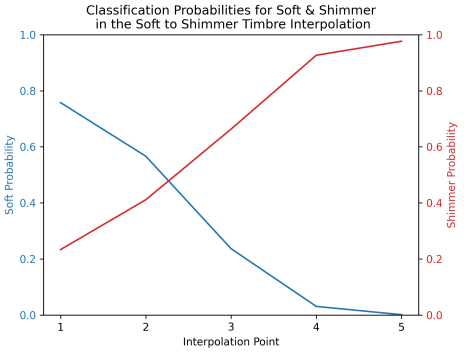
(c) Fuzz-Shimmer



(d) Thin-Soft



(e) Thin-Shimmer



(f) Soft-Shimmer

Figure 6.7: Classification probabilities for interpolations between the top descriptors of each timbre group.

audio file reflects the intended timbral qualities and maintains the correct proportional representation of these qualities as dictated by the interpolation parameters. In addition to verifying the accuracy of single linear timbre interpolations, assessing the VAE's performance across multiple interpolation tasks was critical to ensure that the model does not confuse or improperly blend timbral characteristics from unrelated interpolations.

To address this, the evaluation was built around a perceptual experiment asking humans to correctly place audio files in terms of perceptual distance from three distinct interpolations from three distinct pairs of a set of timbre descriptors. The three timbre interpolations with the closest probability convergence to the central data point and the highest start/target probabilities from the objective evaluation via classification were selected to provide the audio samples for the experiment. Figures 6.6 and 6.7 show that the best-performing interpolations based on this criteria were Fuzz-Shimmer, Fuzz-Thin, and Shimmer-Thin. These interpolations were selected for the perceptual experiment.

To begin the experiment, participants listened to the three original audio samples at the start and end of each interpolation: the Fuzz, Shimmer, and Thin sounds with a timbre magnitude of 100. They were then asked to place the interpolated audio samples based on perceptual timbral distance from the original three sounds. Figure 6.8 shows the triangle template from which participants were asked to place the interpolated sounds in such a manner.

The same twenty participants recruited for the timbre generation perceptual experiment were also recruited for this ethically reviewed and approved experiment. Three descriptors and three 5-point interpolations were used to appropriately scope the experiment's expectations of human participants. Three 5-point interpolations resulted in each participant placing 12 audio files in terms of perceptual distance, which provides a good balance between experimental rigour and participants' expenditure. This experimental approach helped determine the model's robustness and capacity to segregate and apply the correct timbral characteristics specific to each interpolation path.

6.3.5 Perceptual evaluation results & discussion

Conceptually, asking a participant to place the audio files in terms of perceptual timbre distance on the provided triangle template (see Figure 6.8) is akin to a multi-class classification problem where participants are 'classifying' the positions of audio samples on the triangle. For this reason, a confusion matrix was used to visualise the rankings for all interpolated audio files between the three interpolations across all twenty participants. Figure 6.9 shows this confusion matrix.

The confusion matrix shows that all participants correctly placed half of the interpolated sounds on the triangle template. When some participants misplaced sounds, they never misplaced sounds further than one place from their true placements on the interpolation

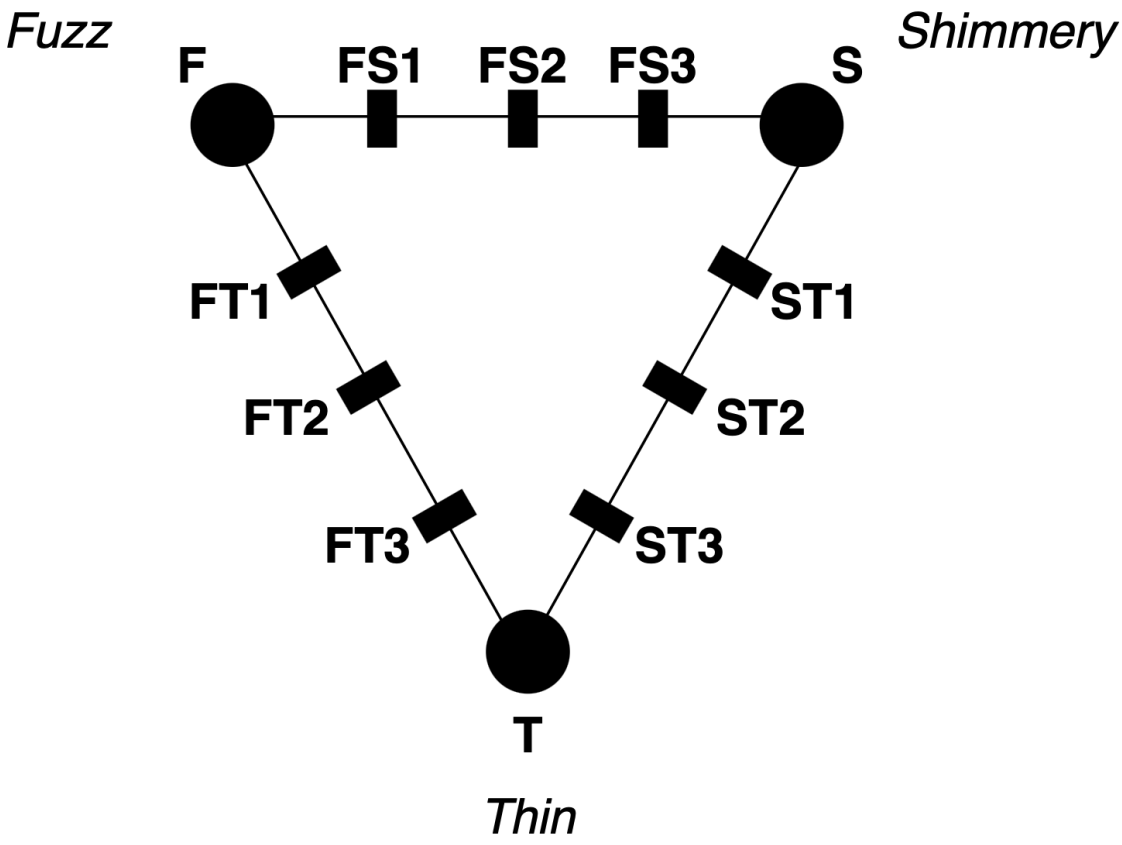


Figure 6.8: Triangle template.

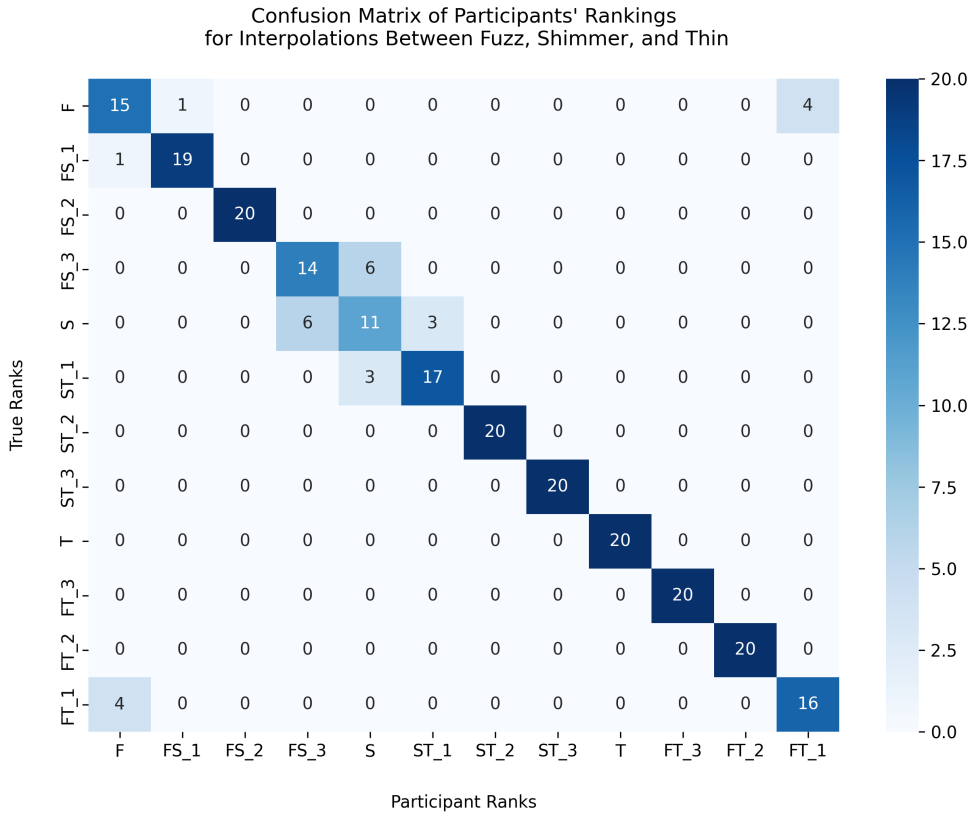


Figure 6.9: Participants' placements for interpolated audio samples. Refer to Figure 6.8 for class locations on the experiment's triangle.

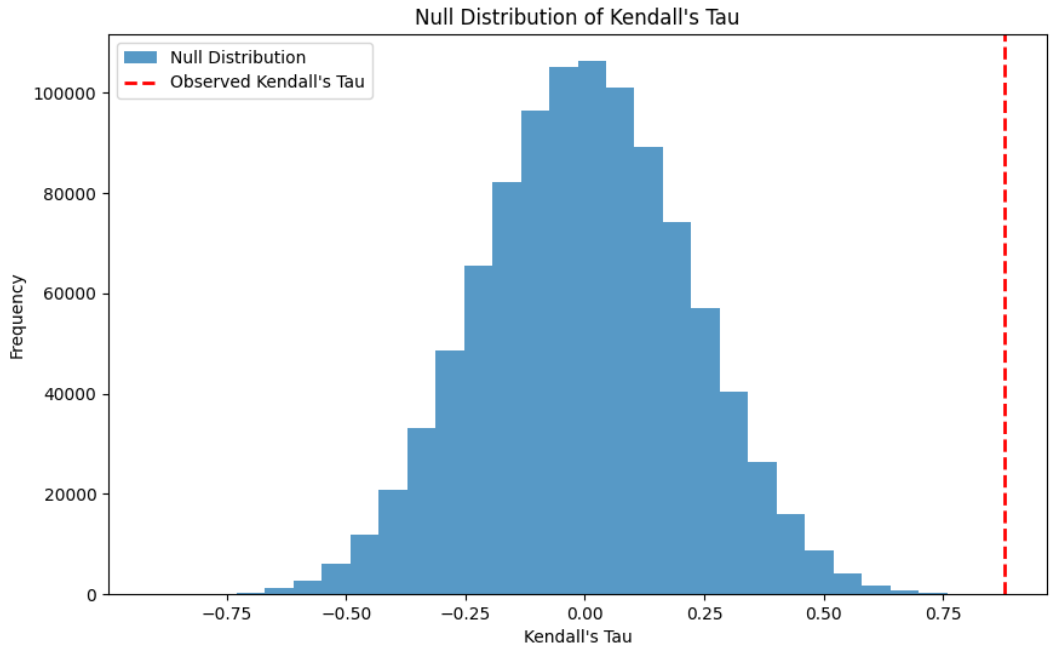


Figure 6.10: Kendall's Tau null distribution.

scale. These incorrect 'neighbour' placements all occurred around the F and S locations, indicating that the 'fuzz' and 'shimmer' audio samples were perceptually harder to distinguish from their neighbours in their respective interpolations. Overall, these results indicate that humans could perceptually organise the interpolated sounds in terms of perceptual distance from one another and that even the few errors in judgement were minimised to merely swapping close timbral neighbours along the interpolation.

To statistically validate the significance of the participants' rankings, a Kendall's Tau correlation coefficient [90, 150] was calculated between the sounds' true ranks and aggregated participant ranks. The participant ranks were aggregated by averaging their ranks for each item and converting these averages into discrete ranks. The observed Kendall's Tau for the aggregated ranks was 0.879. To test the null hypothesis that the participant rankings are no better than random, a permutation test for Kendall's Tau was conducted [72, 166]. The true ranks were randomly permuted 1,000,000 times to generate a null distribution of Kendall's Tau values. Figure 6.10 shows the resulting null distribution and the position of the observed Kendall's Tau value.

The observed Kendall's Tau was compared to the null distribution, resulting in a **p-value < 0.001**. The null hypothesis was therefore rejected, indicating that the participant rankings are significantly better than random rankings, providing strong evidence that participants accurately ranked the sounds.

Chapter 7

Conclusion

This dissertation made significant strides in semantic timbre recognition and generation by developing and evaluating systems designed to recognise, generate, and interpolate the timbre of monophonic electric guitar sounds via semantic descriptors. By focusing on semantic timbre descriptors relevant to the electric guitar, this project not only enhances the field of music technology but also provides valuable tools for audio processing and music production.

The SemanticTimbreDataset introduced in Chapter 3 marks a pivotal achievement in providing a structured and detailed basis for studying timbre. It is specifically tailored to capture a broad range of timbral nuances through twenty semantic descriptors relevant to electric guitar sounds, making it an invaluable resource for training sophisticated machine learning models.

The timbre recognition system's CNNs, detailed and evaluated in Chapters 4 and 6, successfully recognise various timbral characteristics from the SemanticTimbreDataset. Most models achieve RMSE values below 15 and $\boldsymbol{R^2}$ scores above 0.73 across sounds produced by the two most popular electric guitars worldwide. This system sets a new standard in timbre recognition by focusing on subtle nuances within the timbre of a single instrument described by descriptors, aligning closely with the practical needs of music producers who fine-tune individual instrument sounds for harmonious track mixing through natural language.

The timbre generation system detailed and evaluated in Chapters 5 and 6 showcases a VAE that generates new guitar sounds with specified timbral qualities and interpolates between them. This exploration of timbre interpolation represents a significant advancement in the field. By manipulating the latent space of the VAE, this project successfully demonstrated how different timbral characteristics described by descriptors such as 'bright' and 'dark' can be blended seamlessly. This method showcases the versatility of VAEs in audio processing and demonstrates a new standard for timbre manipulation in digital music production, offering unprecedented control and creativity in sound design.

The practical implications of this research are vast and exciting. For music producers and sound engineers, the ability to finely tune and creatively manipulate timbre with natural language can transform the music production process, making it more intuitive, accessible, and aligned with artistic visions. Furthermore, the methodologies developed here could be adapted for use in other digital audio applications, potentially leading to innovations in how audio is synthesised, modified, and implemented across various media.

7.1 Limitations & future work

Despite these advancements, the project faced obstacles that provide directions for future research.

The SemanticTimbreDataset labels each sound with only one timbre descriptor and its magnitude. Future datasets could include labels for multiple timbre descriptors and their magnitudes per sound, facilitating the development of more complex models like conditional VAEs [173], which may learn the interdependencies between different timbral characteristics more effectively.

The scope of timbre generation was constrained by available computational resources and the project timeline. Future work could expand the training data to cover a broader range of pitches and timbre magnitudes, exploring interpolations beyond the fixed pitches of Western staff notation [60] and timbre magnitude parameters. The timbre generation VAE can currently interpolate between pitches and various timbre magnitudes, so an expanded evaluation could investigate these properties.

A VAE architecture was chosen for the timbre generation system mainly due to VAEs' interpolation capabilities [101, 156, 9, 136, 31] and training stability [11, 110, 211, 130] while considering the project's timeline. However, given the tendency of VAEs to produce outputs that might lack sharpness, as evidenced in previous work [44, 98, 73, 203, 146, 130], investigating GANs for generating spectrograms could yield higher-quality audio synthesis. Comparing the timbral clarity and authenticity of GAN-generated sounds against VAE outputs would be a valuable area of further study.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `https://www.tensorflow.org/`. Software available from tensorflow.org.

[2] André Almeida, Emery Schubert, and Joe Wolfe. Timbre Vibrato Perception and Description. *Music Perception*, 38(3):282–292, February 2021. ISSN 0730-7829, 1533-8312. doi: 10.1525/mp.2021.38.3.282. URL `https://online.ucpress.edu/mp/article/38/3/282/116128/Timbre-Vibrato-Perception-and-Description`.

[3] Vox Amps. The story of wah wah, September 2021. URL `https://www.voxamps.co.uk/blogs/updates/the-story-of-wah-wah`. Accessed on February 23rd, 2024.

[4] Sanjeev Arora, Andrej Risteski, and Yi Zhang. Do GANs learn the distribution? some theory and empirics. In *Proceedings of the International Conference on Learning Representations 2018*, 2018. URL `https://openreview.net/forum?id=BJehNfW0-`.

[5] Michael Astley-Brown. Ibanez Tube Screamer Mini review, September 2021. URL `https://www.musicradar.com/reviews/guitars/ibanez-tube-screamer-mini-631948`.

[6] Mathieu Barthet, Richard Kronland-Martinet, and Sølvi Ystad. Consistency of timbre patterns in expressive music performance. In *9th International Conference on Digital Audio Effects*, pages 19–25, September 2006. URL `https://hal.science/hal-00463315/`.

[7] R. Baselmans, N. H. van Schijndel, and R. P. N. Duisters. Measuring the Effect of Signal-To-Noise Ratio on Listening Fatigue (Technical Note PR-TN

2010/00235). Technical Report PR-TN 2010/00235, Philips Research Europe, 2010. URL https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=9016eaecb2e26319603d1a301c4cc6e66e523cdd.

[8] Sam Bell. Boss BD-2 Blues Driver - REVIEW, December 2020. URL https://www.guitarinteractivemagazine.com/review/boss-bd-2-blues-driver/.

[9] David Berthelot, Colin Raffel, Aurko Roy, and Ian J. Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer. In *ICLR (Poster)*, 2019. URL https://openreview.net/forum?id=S1fQSiCcYm.

[10] Maciej Blaszke and Bożena Kostek. Musical instrument identification using deep learning approach. *Sensors*, 22(8):3033, April 2022. ISSN 1424-8220. doi: 10.3390/s22083033. URL http://dx.doi.org/10.3390/s22083033.

[11] Russell Sammut Bonnici, Martin Benning, and Charalampos Saitis. Timbre Transfer with Variational Auto Encoding and Cycle-Consistent Adversarial Networks. In *2022 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, Padua, Italy, July 2022. IEEE. ISBN 978-1-72818-671-9. doi: 10.1109/IJCNN55064.2022.9892107. URL https://ieeexplore.ieee.org/document/9892107/.

[12] Juan J. Bosch, Jordi Janer, Ferdinand Fuhrmann, and Perfecto Herrera. A Comparison of Sound Segregation Techniques for Predominant Instrument Recognition in Musical Audio Signals. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR 2012)*, pages 559–564, 2012. URL https://ismir2012.ismir.net/event/papers/559_ISMIR_2012.pdf.

[13] David M. Brewster. *Introduction to Guitar Tone & Effects: An Essential Manual for Getting the Best Sounds from Electric Guitars, Amplifiers, Effect Pedals, and Digital Processors*. Guitar Educational Series. Hal Leonard, 2003. ISBN 9780634060465. URL https://books.google.co.uk/books?id=q99-bY3cL8YC.

[14] Michel Buffa and Jerome Lebrun. Real time tube guitar amplifier simulation using WebAudio. In *Web Audio Conference 2017 – Collaborative Audio #WAC2017*, Queen Mary Research Online (QMRO) repository. http://qmro.qmul.ac.uk/xmlui/handle/123456789/26089, London, United Kingdom, August 2017. Queen Mary University of London. URL https://hal.univ-cotedazur.fr/hal-01589229.

[15] Michel Buffa and Jerome Lebrun. Real-time emulation of a marshall jcm 800 guitar tube amplifier, audio fx pedals, in a virtual pedal board. In *Companion Proceedings of the The Web Conference 2018*, WWW '18, page 179–182, Republic

and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee. ISBN 9781450356404. doi: 10.1145/3184558.3186973. URL https://doi.org/10.1145/3184558.3186973.

[16] Michel Buffa and Jerome Lebrun. Webaudio virtual tube guitar amps and pedal board design. In *Web Audio Conference 2018*. HAL, 2018.

[17] Juan José Burred, Axel Robel, and Thomas Sikora. Dynamic spectral envelope modeling for timbre analysis of musical instrument sounds. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):663–674, 2010. doi: 10.1109/TASL.2009.2036300.

[18] Marcelo Caetano, Charalampos Saitis, and Kai Siedenburg. Audio content descriptors of timbre. In Kai Siedenburg, Charalampos Saitis, Stephen McAdams, Arthur N. Popper, and Richard R. Fay, editors, *Timbre: Acoustics, Perception, and Cognition*, pages 297–333. Springer International Publishing, 2019. ISBN 978-3-030-14832-4. doi: 10.1007/978-3-030-14832-4_11. URL https://doi.org/10.1007/978-3-030-14832-4_11.

[19] Antoine Caillon and Philippe Esling. Rave: A variational autoencoder for fast and high-quality neural audio synthesis. *CoRR*, abs/2111.05011, 2021. URL https://arxiv.org/abs/2111.05011.

[20] Joseph Manfredi Cameron. Spectrograms of Electric Guitar Notes in the Semantic-TimbreDataset, May 2024. URL https://doi.org/10.5281/zenodo.11398030.

[21] Joseph Manfredi Cameron. VAE-Generated Monophonic Electric Guitar Notes, May 2024. URL https://doi.org/10.5281/zenodo.11398170.

[22] Joseph Manfredi Cameron. Monophonic Electric Guitar Notes to Train a Timbre Generation VAE, May 2024. URL https://doi.org/10.5281/zenodo.11398253.

[23] Shan Carter and Michael Nielsen. Using Artificial Intelligence to Augment Human Intelligence. *Distill*, 2(12):10.23915/distill.00009, December 2017. ISSN 2476-0757. doi: 10.23915/distill.00009. URL https://distill.pub/2017/aia.

[24] Alexander U. Case, Agnieszka Roginska, Justin D. Mathew, and Jim Anderson. Electric guitar - A blank canvas for timbre and tone. *Proceedings of Meetings on Acoustics*, 19(1):015039, May 2013. ISSN 1939-800X. doi: 10.1121/1.4800310. URL https://doi.org/10.1121/1.4800310.

[25] Francois Chollet and Others. Keras 2.15.0, December 2023. URL https://github.com/keras-team/keras.

[26] Marco Comunità, Dan Stowell, and Joshua D. Reiss. Guitar effects recognition and parameter estimation with convolutional neural networks. *Journal of the Audio Engineering Society*, 69(7/8):594–604, 2021. URL https://www.aes.org/e-lib/browse.cfm?elib=21124.

[27] Chris Corfield. Best guitar VSTs 2024: guitar plugins and software to supercharge your guitar recordings. *MusicRadar*, February 2023. URL https://www.musicradar.com/news/best-guitar-vsts-and-guitar-plugins.

[28] Boss (Roland Corporation). Boss Blues Driver BD-2 Pedal. https://www.boss.info/global/products/bd-2/, 2023. Accessed: 07/12/2023.

[29] Marion Cousineau, Samuele Carcagno, Laurent Demany, and Daniel Pressnitzer. What is a melody? On the relationship between pitch and brightness of timbre. *Frontiers in Systems Neuroscience*, 7, 2014. ISSN 1662-5137. doi: 10.3389/fnsys.2013.00127. URL http://journal.frontiersin.org/article/10.3389/fnsys.2013.00127/abstract.

[30] Antonia Creswell, Tom White, Vincent Dumoulin, Kai Arulkumaran, Biswa Sengupta, and Anil A. Bharath. Generative Adversarial Networks: An Overview. *arXiv:1710.07035 [cs]*, October 2017. URL http://arxiv.org/abs/1710.07035. arXiv: 1710.07035.

[31] Paulino Cristovao, Hidemoto Nakada, Yusuke Tanimura, and Hideki Asoh. Generating in-between images through learned latent space representation using variational autoencoders. *IEEE Access*, 8:149456–149467, 2020. doi: 10.1109/ACCESS.2020.3016313.

[32] Alexandre Défossez, Neil Zeghidour, Nicolas Usunier, Leon Bottou, and Francis Bach. Sing: symbol-to-instrument neural generator. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 9055–9065, Red Hook, NY, USA, 2018. Curran Associates Inc.

[33] Laurent Demany and Catherine Semal. Pitch versus brightness of timbre: Detecting combined shifts in fundamental and formant frequency. *Music Perception: An Interdisciplinary Journal*, 11(1):1–13, 1993. ISSN 07307829, 15338312. URL http://www.jstor.org/stable/40285596.

[34] Prafulla Dhariwal, Heewoo Jun, Christine Payne, Jong Wook Kim, Alec Radford, and Ilya Sutskever. Jukebox: A generative model for music. *arXiv preprint arXiv:2005.00341*, 2020.

[35] Alastair C. Disley, David M. Howard, and Andy Hunt. Timbral Description of Musical Instruments. In *Proceedings of the 9th International Conference on Music Perception and Cognition*, pages 61–68, Bologna, Italy, January 2006.

[36] Jayson Kerr Dobney, Craig J. Inciardi, Anthony DeCurtis, Alan Di Perna, David Fricke, Holly George-Warren, and Matthew W. Hill. The Quintessential Quartet. In *Play It Loud: Instruments of Rock & Roll*, pages 17–52. The Metropolitan Museum of Art, New York, 2019. ISBN 978-1-58839-666-2. OCLC: on1048937652.

[37] Felix A. Dobrowohl, Andrew J. Milne, and Roger T. Dean. Timbre preferences in the context of mixing music. *Applied Sciences*, 9(8):1695, April 2019. ISSN 2076-3417. doi: 10.3390/app9081695. URL `https://www.mdpi.com/2076-3417/9/8/1695`.

[38] Emily I. Dolan and Alexander Rehding, editors. *The Oxford handbook of timbre*. Oxford handbooks. Oxford University Press, New York, 2021. ISBN 978-0-19-063722-4.

[39] Chris Donahue, Julian McAuley, and Miller Puckette. Adversarial audio synthesis. In *Proceedings of the International Conference on Learning Representations 2019*, 2019. doi: 10.48550/ARXIV.1802.04208. URL `https://openreview.net/forum?id=ByMVTsR5KQ`.

[40] A. R. Duchossoir. *The Fender Stratocaster*. Hal Leonard, Milwaukee, WI, rev. ; 40th anniversary ed edition, 1994. ISBN 978-0-7935-4735-7. OCLC: 33099787.

[41] Simon Duggal. *Transients*, pages 197–199. Springer Nature Switzerland, 2024. ISBN 978-3-031-40067-4. doi: 10.1007/978-3-031-40067-4_17. URL `https://doi.org/10.1007/978-3-031-40067-4_17`.

[42] P. Dutilleux, K. Dempwolf, M. Holters, and U. Zölzer. *Nonlinear Processing*, chapter 4, pages 101–138. John Wiley and Sons, Ltd, 2011. ISBN 9781119991298. doi: https://doi.org/10.1002/9781119991298.ch4. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119991298.ch4`.

[43] P. Dutilleux, M. Holters, S. Disch, and U. Zölzer. *Modulators and Demodulators*, chapter 3, pages 83–99. John Wiley and Sons, Ltd, 2011. ISBN 9781119991298. doi: https://doi.org/10.1002/9781119991298.ch3. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119991298.ch3`.

[44] Mohamed El-Kaddoury, Abdelhak Mahmoudi, and Mohammed Majid Himmi. Deep Generative Models for Image Generation: A Practical Comparison Between Variational Autoencoders and Generative Adversarial Networks. In Éric Renault, Selma Boumerdassi, Cherkaoui Leghris, and Samia Bouzefrane, editors, *Mobile, Secure,*

and Programmable Networking, pages 1–8. Springer International Publishing, 2019. ISBN 978-3-030-22885-9.

[45] Taffeta M. Elliott, Liberty S. Hamilton, and Frédéric E. Theunissen. Acoustic structure of the five perceptual dimensions of timbre in orchestral instrument tones. *The Journal of the Acoustical Society of America*, 133(1):389–404, January 2013. ISSN 0001-4966, 1520-8524. doi: 10.1121/1.4770244. URL https://pubs.aip.org/jasa/article/133/1/389/929560/Acoustic-structure-of-the-five-perceptual.

[46] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Mohammad Norouzi, Douglas Eck, and Karen Simonyan. Neural audio synthesis of musical notes with WaveNet autoencoders. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1068–1077, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL https://proceedings.mlr.press/v70/engel17a.html.

[47] Robert Fink, Melinda Latour, and Zachary Wallmark, editors. *The Relentless Pursuit of Tone: Timbre in Popular Music*, volume 1. Oxford University Press, October 2018. ISBN 978-0-19-998522-7. doi: 10.1093/oso/9780199985227.001.0001. URL https://academic.oup.com/book/10894.

[48] Robert Fink, Zachary Wallmark, and Melinda Latour. Chapter 1 Introduction: Chasing the Dragon: In Search of Tone in Popular Music. In *The Relentless Pursuit of Tone: Timbre in Popular Music*. Oxford University Press, October 2018. ISBN 9780199985227. doi: 10.1093/oso/9780199985227.003.0001. URL https://doi.org/10.1093/oso/9780199985227.003.0001.

[49] Rod Fogg and Dave Hunter. The story of the electric guitar. In *The Electric Guitar Handbook: [A Complete Course in Modern Technique and Styles]*, pages 5–23. Backbeat, Milwaukee, Wis., 1st ed edition, 2009. ISBN 978-0-87930-989-3. OCLC: 632820551.

[50] Claudia Fritz, Alan F. Blackwell, Ian Cross, Jim Woodhouse, and Brian C. J. Moore. Exploring violin sound quality: Investigating English timbre descriptors and correlating resynthesized acoustical modifications with perceptual properties. *The Journal of the Acoustical Society of America*, 131(1):783–794, January 2012. ISSN 0001-4966, 1520-8524. doi: 10.1121/1.3651790. URL https://pubs.aip.org/jasa/article/131/1/783/824051/Exploring-violin-sound-quality-Investigating.

[51] Luis-Manuel Garcia. Beats, flesh, and grain: sonic tactility and affect in electronic

dance music. *Sound Studies*, 1(1):59–76, 2015. doi: 10.1080/20551940.2015.1079072. URL https://doi.org/10.1080/20551940.2015.1079072.

[52] Martin Gerber. Timbre: Understanding and Manipulating Texture. *Music 101, Sessionville*, August 2014. URL https://sessionville.com/articles/timbre-understanding-and-manipulating-texture.

[53] Rolf Inge Godoy, Marc Leman, Tor Halmrast, Knut Guettler, and Rolf Bader. Gesture and Timbre. In *Musical Gestures: Sound, Movement, and Meaning*, pages 183–211. Routledge, illustrated edition, 2010. ISBN 978-1-135-18363-9.

[54] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Chapter 14: Autoencoders. In *Deep Learning*, pages 499–523. MIT Press, 2016. http://www.deeplearningbook.org.

[55] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Chapter 9: Convolutional networks. In *Deep Learning*, pages 326–366. MIT Press, 2016. http://www.deeplearningbook.org.

[56] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[57] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative Adversarial Networks. *arXiv:1406.2661 [stat.ML]*, 2014. doi: 10.48550/ARXIV.1406.2661. URL https://arxiv.org/abs/1406.2661. Publisher: arXiv Version Number: 1.

[58] Alex Gounaropoulos and Colin Johnson. Synthesising timbres and timbre-changes from adjectives/adverbs. In Franz Rothlauf, Jürgen Branke, Stefano Cagnoni, Ernesto Costa, Carlos Cotta, Rolf Drechsler, Evelyne Lutton, Penousal Machado, Jason H. Moore, Juan Romero, George D. Smith, Giovanni Squillero, and Hideyuki Takagi, editors, *Applications of Evolutionary Computing*, pages 664–675, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. ISBN 978-3-540-33238-1.

[59] Jacek Grekow and Teodora Dimitrova-Grekow. Monophonic music generation with a given emotion using conditional variational autoencoder. *IEEE Access*, 9:129088–129101, 2021. doi: 10.1109/ACCESS.2021.3113829.

[60] James Grier. *Musical Notation in the West*. Cambridge Introductions to Music. Cambridge University Press, 2021. ISBN 9780521898164.

[61] Daniel W. Griffin and Jae S. Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243, 1984. doi: 10.1109/TASSP.1984.1164317.

[62] Karlheinz Gröchenig. The short-time fourier transform. In *Foundations of Time-Frequency Analysis*, pages 37–58. Birkhäuser Boston, Boston, MA, 2001. ISBN 978-1-4612-0003-1. doi: 10.1007/978-1-4612-0003-1_4. URL https://doi.org/10.1007/978-1-4612-0003-1_4.

[63] Donald E. Hall. *Musical Acoustics*. Brooks/Cole Publishing Company, 2nd edition edition, 1991. ISBN 978-0-534-13248-4.

[64] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020. doi: 10.1038/s41586-020-2649-2. URL https://doi.org/10.1038/s41586-020-2649-2.

[65] Peter M. C. Harrison. Chapter 9: Timbre. In *Music and Science*. GitHub Pages, 2024. URL https://pmcharrison.github.io/intro-to-music-and-science/timbre.html.

[66] Hermann Helmholtz. Force, Pitch, and Quality - in Chapter 1: On the Sensation of Sound in General. In *On the Sensations of Tone As A Physiological Basis for the Theory of Music*, page 10. Dover Publications, 2nd edition edition, June 1954. ISBN 978-0-486-60753-5.

[67] Jan-Peter Herbst. Empirical Explorations of Guitar Players' Attitudes Towards Their Equipment and the Role of Distortion in Rock Music. *Current Musicology*, page No 105 (2019): Current Musicology, September 2019. doi: 10.7916/CM.V0I105.5404. URL https://journals.library.columbia.edu/index.php/currentmusicology/article/view/5404. Publisher: Current Musicology.

[68] Carlos Hernandez-Olivan and Jose R. Beltran. Timbre Classification of Musical Instruments with a Deep Learning Multi-Head Attention-Based Model. *CoRR*, 2021. doi: 10.48550/ARXIV.2107.06231. URL https://arxiv.org/abs/2107.06231.

[69] Shawn Hershey, Sourish Chaudhuri, Daniel P. W. Ellis, Jort F. Gemmeke, Aren Jansen, R. Channing Moore, Manoj Plakal, Devin Platt, Rif A. Saurous, Bryan Seybold, Malcolm Slaney, Ron J. Weiss, and Kevin Wilson. Cnn architectures for large-scale audio classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 131–135, 2017. doi: 10.1109/ICASSP.2017.7952132.

[70] Reemt Hinrichs, Kevin Gerkens, Alexander Lange, and Jörn Ostermann. Convolutional neural networks for the classification of guitar effects and extraction of the parameter settings of single and multi-guitar effects from instrument mixes. *EURASIP Journal on Audio, Speech, and Music Processing*, 2022(1):28, October 2022. ISSN 1687-4722. doi: 10.1186/s13636-022-00257-4. URL https://asmp-eurasipjournals.springeropen.com/articles/10.1186/s13636-022-00257-4.

[71] Patricia A. Holmes. An exploration of musical communication through expressive use of timbre: The performer's perspective. *Psychology of Music*, 40(3):301–323, May 2012. ISSN 0305-7356, 1741-3087. doi: 10.1177/0305735610388898. URL http://journals.sagepub.com/doi/10.1177/0305735610388898.

[72] Charles A. Holt and Sean P. Sullivan. Permutation tests for experimental data. *Experimental Economics*, 26(4):775–812, September 2023. ISSN 1386-4157, 1573-6938. doi: 10.1007/s10683-023-09799-6. URL https://link.springer.com/10.1007/s10683-023-09799-6.

[73] Huaibo Huang, Zhihang Li, Ran He, Zhenan Sun, and Tieniu Tan. Introvae: Introspective variational autoencoders for photographic image synthesis. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper_files/paper/2018/file/093f65e080a295f8076b1c5722a46aa2-Paper.pdf.

[74] Sicong Huang, Qiyang Li, Cem Anil, Xuchan Bao, Sageev Oore, and Roger B. Grosse. TimbreTron: A WaveNet(CycleGAN(CQT(Audio))) Pipeline for Musical Timbre Transfer. In *International Conference on Learning Representations 2019*, May 2019. URL https://openreview.net/forum?id=S1lvm305YQ.

[75] Dave Hunter. *The Electric Guitar Sourcebook: How to Find the Sounds You Like*. Backbeat Books, San Francisco, 1st ed edition, 2006. ISBN 978-0-87930-886-5. OCLC: ocm68046196.

[76] Dave Hunter. *The Fender stratocaster: the life and times of the world's greatest guitar and its players*. Voyageur Press, Minneapolis, MN, 2013. ISBN 978-0-7603-4484-2.

[77] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55.

[78] Ibanez. Ibanez TSMINI Tube Screamer Guitar Pedal, 2024. URL https://www.ibanez.com/usa/products/detail/ts_mini_01.html.

[79] Apple Inc. Legacy Guitar in Logic Pro for Mac, 2023. URL https://support.apple.com/en-ph/guide/logicpro/lgsife1e9c56/mac.

[80] Apple Inc. Pedalboard overview in Logic Pro for Mac, 2023. URL https://support.apple.com/en-sa/guide/logicpro/lgcef14a2681/mac.

[81] Apple Inc. Logic Pro, 2023. URL https://www.apple.com/uk/logic-pro/.

[82] Native Instruments. ADSR explained: How to control synth envelopes in your music, June 2023. URL https://blog.native-instruments.com/adsr-explained/.

[83] Native Instruments. Guitar Rig 7 Pro, December 2023. URL https://www.native-instruments.com/en/products/komplete/guitar/guitar-rig-7-pro/.

[84] Native Instruments. Guitar Rig Manual: Components Reference, December 2023. URL https://native-instruments.com/ni-tech-manuals/guitar-rig-manual/en/components-reference.

[85] Kristoffer Jensen. Irregularities, Noise and Random Fluctuations in Musical Sounds. *The Journal of Music and Meaning*, 2, 2004. URL http://www.musicandmeaning.net/issues/showArticle.php?artID=2.3.

[86] Joe@t.blog. Wah't is the wah-wah?, August 2017. URL https://www.thomann.de/blog/en/wah-wah/#:~:text=The%20principle%20of%20the%20wah,%20or%20œwow%20effect. Accessed on February 23rd, 2024.

[87] Colin G. Johnson and Alex Gounaropoulos. Timbre Interfaces using Adjectives and Adverbs. In Norbert Schnell, editor, *Proceedings of the 2006 Conference on New Interfaces for Musical Expression*, pages 101–102, Paris France, 2006. ISBN 2-84426-314-3. URL https://kar.kent.ac.uk/14472/1/Timbre_interfaces_using_adjectives_and_adverbs.pdf.

[88] Malcolm John Joyce. The Fender Stratocaster Electric Guitar: A Case Study for Both Nontransferable and Transferable Skills Learning in a Generalist Electronic Engineering Cohort. *IEEE Transactions on Education*, 53(3):397–404, August 2010. ISSN 0018-9359, 1557-9638. doi: 10.1109/TE.2009.2025369. URL http://ieeexplore.ieee.org/document/5286255/.

[89] Henrik Jürgens, Reemt Hinrichs, and Jörn Ostermann. Recognizing guitar effects and their parameter settings. In *Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx2020)*, Vienna, Austria, 2020. URL https://www.dafx.de/paper-archive/details.php?id=prT5ii5OJlX8Y3yAoQj-2A.

[90] M. G. Kendall. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93, 1938. ISSN 00063444. URL http://www.jstor.org/stable/2332226.

[91] Roger A. Kendall and Edward C. Carterette. Verbal attributes of simultaneous wind instrument timbres: I. von bismarck's adjectives. *Music Perception: An Interdisciplinary Journal*, 10(4):445–467, 1993. ISSN 07307829, 15338312. URL http://www.jstor.org/stable/40285583.

[92] Keras. Keras 2 Layer activations: Keras 2 API Documentation, December 2023. URL https://keras.io/2.16/api/layers/activations/.

[93] Keras. Keras 2 Adam Optimizer: Keras 2 API Documentation, December 2023. URL https://keras.io/2.16/api/optimizers/adam/.

[94] Keras. Keras 2 API Documentation, December 2023. URL https://keras.io/2.16/api/.

[95] Keras. Keras 2 The Model Class: Keras 2 API Documentation, December 2023. URL https://keras.io/2.16/api/models/model/.

[96] Keras. Keras 2 Model Training APIs: Keras 2 API Documentation, December 2023. URL https://keras.io/2.16/api/models/model_training_apis/.

[97] Keras. Keras 2 Mean Squared Error: Keras 2 API Documentation, December 2023. URL https://keras.io/2.16/api/losses/regression_losses/#meansquarederror-class.

[98] Salman H. Khan, Munawar Hayat, and Nick Barnes. Adversarial training of variational auto-encoders for high fidelity image generation. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1312–1320, 2018. doi: 10.1109/WACV.2018.00148.

[99] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]*, December 2014. URL http://arxiv.org/abs/1412.6980. arXiv: 1412.6980.

[100] Diederik P. Kingma and Max Welling. Auto-Encoding Variational Bayes. In *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[101] Diederik P. Kingma and Max Welling. An Introduction to Variational Autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, 2019. ISSN 1935-8237, 1935-8245. doi: 10.1561/2200000056. URL http://www.nowpublishers.com/article/Details/MAL-056.

[102] Durk P Kingma, Tim Salimans, and Max Welling. Variational dropout and the local reparameterization trick. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/bc7316929fe1545bf0b98d114ee3ecb8-Paper.pdf.

[103] Allen Kozzin. Guitar. a modest proposal: Serious music for the electric guitar. *Music Journal*, 35(4):26, Apr 01 1977. URL https://ezp.lib.cam.ac.uk/login?url=https://www.proquest.com/scholarly-journals/guitar-modest-proposal-serious-music-electric/docview/1290707325/se-2. Last updated - 2013-02-21.

[104] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 25, page 1097–1105. Curran Associates, Inc., 2012. URL https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf.

[105] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brebisson, Yoshua Bengio, and Aaron Courville. Melgan: generative adversarial networks for conditional waveform synthesis. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, Red Hook, NY, USA, 2019. Curran Associates Inc.

[106] Megan L. Lavengood. The cultural significance of timbre analysis: A case study in 1980s pop music, texture, and narrative. *Music Theory Online*, 26(3), September 2020. ISSN 1067-3040. doi: 10.30535/mto.26.3.3. URL https://mtosmt.org/issues/mto.20.26.3/mto.20.26.3.lavengood.html.

[107] Samuel K. Lavine. "Are You Experienced'? The Life, Music, and Legacy of Jimi Hendrix. *Student Publications at Gettysburg College*, 1086, 2023. URL https://cupola.gettysburg.edu/student_scholarship/1086/.

[108] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, R. Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. In D. Touretzky, editor, *Advances in Neural Information Processing Systems*, volume 2. Morgan-Kaufmann, 1989. URL https://proceedings.neurips.cc/paper_files/paper/1989/file/53c3bce66e43be4f209556518c2fcb54-Paper.pdf.

[109] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521 (7553):436–444, May 2015. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature14539. URL https://www.nature.com/articles/nature14539.

[110] Je-Yeol Lee and Sang-Il Choi. Improvement of learning stability of generative adversarial network using variational learning. *Applied Sciences*, 10(13), 2020. ISSN 2076-3417. doi: 10.3390/app10134528. URL https://www.mdpi.com/2076-3417/10/13/4528.

[111] Librosa. Librosa Core IO and DSP: Librosa 0.10.1 Documentation, August 2023. URL https://librosa.org/doc/0.10.1/core.html.

[112] Librosa. Librosa Display: Librosa 0.10.1 Documentation, August 2023. URL https://librosa.org/doc/0.10.1/display.html.

[113] Librosa. librosa.griffinlim : Librosa 0.10.1 Documentation, August 2023. URL https://librosa.org/doc/0.10.1/generated/librosa.griffinlim.html#librosa.griffinlim.

[114] Ronald Light. *Pedal Culture: Guitar Effects Pedals as Cultural Artifacts*. Rowman & Littlefield Publishers, Inc., December 2021. ISBN 978-1-4930-6079-5.

[115] Line6. Line 6 Spider V20 MkII Amplifier. https://uk.line6.com/spider-v-mkii/spider-20/, 2023. Accessed: 07/12/2023.

[116] Xiaoluan Liu, Yi Xu, Kai Alter, and Jyrki Tuomainen. Emotional connotations of musical instrument timbre in comparison with emotional speech prosody: Evidence from acoustics and event-related potentials. *Frontiers in Psychology*, 9:737, May 2018. ISSN 1664-1078. doi: 10.3389/fpsyg.2018.00737. URL http://journal.frontiersin.org/article/10.3389/fpsyg.2018.00737/full.

[117] Xubo Liu, Turab Iqbal, Jinzheng Zhao, Qiushi Huang, Mark D. Plumbley, and Wenwu Wang. Conditional sound generation using neural discrete time-frequency representation learning. In *2021 IEEE 31st International Workshop on Machine Learning for Signal Processing (MLSP)*. IEEE, October 2021. doi: 10.1109/mlsp52302.2021.9596430. URL http://dx.doi.org/10.1109/MLSP52302.2021.9596430.

[118] Dave Lockwood. Compression and Limiting. *Home and Studio Recording*, August 1984. URL http://www.muzines.co.uk/articles/compression-and-limiting/4098?theme=2.

[119] Looperman. What is a VST Plugin or VSTi Instrument, May 2012. URL https://www.looperman.com/blog/detail/55/what-is-a-vst-plugin-or-vsti-instrument.

[120] P Manisha and Sujit Gujar. Generative Adversarial Networks (GANs): What it can generate and What it cannot?, 2018. URL https://arxiv.org/abs/1804.00140. Version Number: 2.

[121] Yan Maresz. On computer-assisted orchestration. *Contemporary Music Review*, 32 (1):99–109, 2013. doi: 10.1080/07494467.2013.774515. URL https://doi.org/10.1080/07494467.2013.774515.

[122] Matplotlib. Matplotlib savefig: Matplotlib 3.8.4 Documentation, April 2024. URL https://matplotlib.org/stable/api/_as_gen/matplotlib.pyplot.savefig.html.

[123] Stephen McAdams. Musical timbre perception. In *The Psychology of Music*, pages 35–67. Elsevier, 2013. ISBN 978-0-12-381460-9. doi: 10.1016/B978-0-12-381460-9.00002-X. URL https://linkinghub.elsevier.com/retrieve/pii/B978012381460900002X.

[124] Stephen McAdams. The Perceptual Representation of Timbre. In Kai Siedenburg, Charalampos Saitis, Stephen McAdams, Arthur N. Popper, and Richard R. Fay, editors, *Timbre: Acoustics, Perception, and Cognition*, volume 69, pages 23–57. Springer International Publishing, Cham, 2019. ISBN 978-3-030-14831-7 978-3-030-14832-4. doi: 10.1007/978-3-030-14832-4_2. URL http://link.springer.com/10.1007/978-3-030-14832-4_2. Series Title: Springer Handbook of Auditory Research.

[125] Stephen McAdams and Bruno L. Giordano. 72 The perception of musical timbre. In *Oxford Handbook of Music Psychology*. Oxford University Press, 12 2008. ISBN 9780199298457. doi: 10.1093/oxfordhb/9780199298457.013.0007. URL https://doi.org/10.1093/oxfordhb/9780199298457.013.0007.

[126] Stephen McAdams and Kai Siedenburg. Perception and Cognition of Musical Timbre. In *Foundations in Music Psychology: Theory and Research*, pages 71–120. The MIT Press, Cambridge, Massachusetts, USA, March 2019. ISBN 978-0-262-03927-7.

[127] Brian McFee, Matt McVicar, Daniel Faronbi, Iran Roman, Matan Gover, Stefan Balke, Scott Seyfarth, Ayoub Malek, Colin Raffel, Vincent Lostanlen, Benjamin van Niekirk, Dana Lee, Frank Cwitkowitz, Frank Zalkow, Oriol Nieto, Dan Ellis, Jack Mason, Kyungyun Lee, Bea Steers, ..., and Waldir Pimenta. librosa/librosa: 0.10.1 (0.10.1), 2023. URL https://doi.org/10.5281/zenodo.8252662.

[128] Robert D. Melara and Lawrence E. Marks. Interaction among auditory dimensions: Timbre, pitch, and loudness. *Perception & Psychophysics*, 48(2):169–178, March

1990. ISSN 0031-5117, 1532-5962. doi: 10.3758/BF03207084. URL https://link.springer.com/10.3758/BF03207084.

[129] Brandi Murphy. *Listening Fatigue in College Students*. PhD Dissertation, Texas Tech University, May 2021. URL https://ttu-ir.tdl.org/items/a44c1e51-6130-49be-8c9c-d53ec710cbb9.

[130] Habibeh Naderi, Behrouz Haji Soleimani, and Stan Matwin. Generating high-fidelity images with disentangled adversarial vaes and structure-aware loss. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8, 2020. doi: 10.1109/IJCNN48605.2020.9207056.

[131] Theodora Nestorova. Vibrato. *Timbre and Orchestration Writings*, June 2022. URL https://timbreandorchestration.org/writings/timbre-lingo/2022/6/27/vibrato.

[132] NumPy. Numpy abs: NumPy 1.26 Documentation, September 2023. URL https://numpy.org/doc/stable/reference/generated/numpy.absolute.html.

[133] NumPy. Numpy save: NumPy 1.26 Documentation, September 2023. URL https://numpy.org/doc/stable/reference/generated/numpy.save.html.

[134] Sound on Sound. Wow and Flutter, 2024. URL https://www.soundonsound.com/glossary/wow-flutter.

[135] OpenAI. Jukebox, April 2020. URL https://openai.com/index/jukebox.

[136] Alon Oring, Zohar Yakhini, and Yacov Hel-Or. Autoencoder image interpolation by shaping the latent space. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8281–8290. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/oring21a.html.

[137] Jyri Pakarinen and David T. Yeh. A review of digital techniques for modeling vacuum-tube guitar amplifiers. *Computer Music Journal*, 33(2):85–100, 2009. ISSN 01489267, 15315169. URL http://www.jstor.org/stable/40301029.

[138] The pandas development team. pandas-dev/pandas: Pandas, February 2020. URL https://doi.org/10.5281/zenodo.3509134.

[139] Arthur Paté, Benoît Navarret, Régis Dumoulin, Jean-Loic Le Carrou, Benoît Fabre, and Vincent Doutaut. About the electric guitar: a cross-disciplinary context for an acoustical study. In Société Française d'Acoustique, editor, *Acoustics 2012*, Nantes, France, April 2012. URL https://hal.science/hal-00810875.

[140] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[141] Hegel Pedroza, Gerardo Meza, and Iran R. Roman. EGFxSet: Electric guitar tones processed through real effects of distortion, modulation, delay and reverb, September 2022. URL `https://doi.org/10.5281/zenodo.7044411`.

[142] Geoffroy Peeters, Bruno L. Giordano, Patrick Susini, Nicolas Misdariis, and Stephen McAdams. The Timbre Toolbox: Extracting audio descriptors from musical signals. *The Journal of the Acoustical Society of America*, 130(5):2902–2916, 11 2011. ISSN 0001-4966. doi: 10.1121/1.3642604. URL `https://doi.org/10.1121/1.3642604`.

[143] Henri Penttinen, Vesa Välimäki, and Matti Karjalainen. A digital filtering approach to obtain a more acoustic timbre for an electric guitar. In *2000 10th European Signal Processing Conference*, pages 1–4, 2000.

[144] Nathanaël Perraudin, Peter Balazs, and Peter L. Søndergaard. A fast griffin-lim algorithm. In *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 1–4, October 2013. doi: 10.1109/WASPAA.2013.6701851.

[145] Peshawa J Muhammad Ali and Rezhna Hassan Faraj. Data Normalization and Standardization: A Technical Report. *Machine Learning Technical Reports*, 1(1): 1–6, 2014. doi: 10.13140/RG.2.2.28948.04489. URL `http://rgdoi.net/10.13140/RG.2.2.28948.04489`.

[146] Antoine Plumerault, Hervé Le Borgne, and Céline Hudelot. Avae: Adversarial variational auto encoder. In *2020 25th International Conference on Pattern Recognition (ICPR)*, pages 8687–8694, 2021. doi: 10.1109/ICPR48806.2021.9412727.

[147] Stewart Pollens. *Pitch Notation Conventions*, page xxi–xxii. Cambridge University Press, 2022.

[148] Jordi Pons, Olga Slizovskaia, Rong Gong, Emilia Gómez, and Xavier Serra. Timbre analysis of music audio signals with convolutional neural networks. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 2744–2748, 2017. doi: 10.23919/EUSIPCO.2017.8081710.

[149] Ian S. Port. Prologue. In *The Birth of Loud: Leo Fender, Les Paul, and the Guitar-Pioneering Rivalry That Shaped Rock 'n' Roll*, pages 1–6. Scribner, New York, London, Toronto, Sydney, New Delhi, 2019. ISBN 978-1-5011-4165-2.

[150] Llukan Puka. Kendall's tau. In Miodrag Lovric, editor, *International Encyclopedia of Statistical Science*, pages 713–715. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. ISBN 978-3-642-04898-2. doi: 10.1007/978-3-642-04898-2_324. URL https://doi.org/10.1007/978-3-642-04898-2_324.

[151] Hendrik Purwins, Bo Li, Tuomas Virtanen, Jan Schlüter, Shuo-Yiin Chang, and Tara Sainath. Deep learning for audio signal processing. *IEEE Journal of Selected Topics in Signal Processing*, 13(2):206–219, 2019. doi: 10.1109/JSTSP.2019.2908700.

[152] Thierry Rayna and Ludmila Striukova. Engineering vs. Craftsmanship: Innovation in the Electric Guitar Industry (1945-1984). *SSRN Electronic Journal*, 2008. ISSN 1556-5068. doi: 10.2139/ssrn.1353905. URL http://www.ssrn.com/abstract=1353905.

[153] Ali Razavi, Aäron van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, volume 32, Red Hook, NY, USA, 2019. Curran Associates Inc. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/5f8e2fa1718d1bbcadf1cd9c7a54fb8c-Paper.pdf.

[154] Lindsey Reymore and David Huron. Using Auditory Imagery Tasks to Map the Cognitive Linguistic Dimensions of Musical Instrument Timbre Qualia. *Psychomusicology: Music, Mind, and Brain*, 30(3):124–144, September 2020. ISSN 2162-1535, 0275-3987. doi: 10.1037/pmu0000263. URL http://doi.apa.org/getdoi.cfm?doi=10.1037/pmu0000263.

[155] Lindsey Reymore, Jason Noble, Charalampos Saitis, Caroline Traube, and Zachary Wallmark. Timbre Semantic Associations Vary Both Between and Within Instruments. *Music Perception*, 40(3):253–274, February 2023. ISSN 0730-7829, 1533-8312. doi: 10.1525/mp.2023.40.3.253. URL https://online.ucpress.edu/mp/article/40/3/253/195233/Timbre-Semantic-Associations-Vary-Both-Between-and.

[156] Adam Roberts, Jesse Engel, and Douglas Eck, editors. *Hierarchical Variational Autoencoders for Music*, 2017. URL https://nips2017creativity.github.io/doc/Hierarchical_Variational_Autoencoders_for_Music.pdf.

[157] Bill Robertson. Q: What determines the quality of musical notes? *Science and Children*, 51(6):77–81, 02 2014. URL https://ezp.lib.cam.ac.uk/login?url=https://www.proquest.com/scholarly-journals/q-what-determines-quality-musical-notes/docview/1498085252/se-2. Copyright - Copyright National Science Teachers Association Feb 2014; Document feature - Illustrations; Photographs; Last updated - 2023-12-04; CODEN - SCICBN.

[158] Charalampos Saitis and Stefan Weinzierl. The Semantics of Timbre. In Kai Siedenburg, Charalampos Saitis, Stephen McAdams, Arthur N. Popper, and Richard R. Fay, editors, *Timbre: Acoustics, Perception, and Cognition*, volume 69, pages 119–149. Springer International Publishing, Cham, 2019. ISBN 978-3-030-14831-7 978-3-030-14832-4. doi: 10.1007/978-3-030-14832-4_5. URL http://link.springer.com/10.1007/978-3-030-14832-4_5. Series Title: Springer Handbook of Auditory Research.

[159] Hisao Sakai. Perceptibility of Wow and Flutter. *Journal of the Audio Engineering Society*, 18:290–298, June 1970.

[160] Justin Salamon, Christopher Jacoby, and Juan Pablo Bello. A dataset and taxonomy for urban sound research. In *Proceedings of the 22nd ACM International Conference on Multimedia*, MM '14, page 1041–1044, New York, NY, USA, 2014. Association for Computing Machinery. ISBN 9781450330633. doi: 10.1145/2647868.2655045. URL https://doi.org/10.1145/2647868.2655045.

[161] John Schneider. The Acoustic Guitar. In *The Contemporary Guitar*, pages 11–37. Rowman & Littlefield, revised edition, 2015. ISBN 978-1-4422-3790-2.

[162] John Schneider. *The Contemporary Guitar*. Rowman & Littlefield, Lanham, Maryland, revised and enlarged edition edition, 2015. ISBN 978-1-4422-3788-9 978-1-4422-3789-6 978-1-4422-3790-2.

[163] John Schneider. The Electric Guitar. In *The Contemporary Guitar*, pages 39–56. Rowman & Littlefield, revised edition, 2015. ISBN 978-1-4422-3790-2.

[164] Scikit-learn. Scikit-learn r2_score: Scikit-learn Documentation, April 2024. URL https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html.

[165] Scikit-learn. Scikit-learn train_test_split: Scikit-learn Documentation, April 2024. URL https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html.

[166] SciPy. Scipy scipy.stats.kendalltau: SciPy Documentation, April 2024. URL https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.kendalltau.html.

[167] SciPy. Scipy pearsonr: SciPy Documentation, April 2024. URL https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html.

[168] Carl E. Seashore. The Natural History of the Vibrato. *Proceedings of the National Academy of Sciences*, 17(12):623–626, December 1931. ISSN 0027-8424, 1091-6490. doi: 10.1073/pnas.17.12.623. URL https://pnas.org/doi/full/10.1073/pnas.17.12.623.

[169] Catherine Semal and Laurent Demany. Dissociation of pitch from timbre in auditory short-term memory. *The Journal of the Acoustical Society of America*, 89(5):2404–2410, May 1991. ISSN 0001-4966, 1520-8524. doi: 10.1121/1.400928. URL https://pubs.aip.org/jasa/article/89/5/2404/959976/Dissociation-of-pitch-from-timbre-in-auditory.

[170] Kai Siedenburg, Charalampos Saitis, and Stephen McAdams. The present, past, and future of timbre research. In Kai Siedenburg, Charalampos Saitis, Stephen McAdams, Arthur N. Popper, and Richard R. Fay, editors, *Timbre: Acoustics, Perception, and Cognition*, volume 69, pages 1–19. Springer International Publishing, 2019. ISBN 978-3-030-14831-7 978-3-030-14832-4. doi: 10.1007/978-3-030-14832-4_1. URL http://link.springer.com/10.1007/978-3-030-14832-4_1. Series Title: Springer Handbook of Auditory Research.

[171] Kai Siedenburg, Charalampos Saitis, Stephen McAdams, Arthur N. Popper, and Richard R. Fay, editors. *Timbre: Acoustics, Perception, and Cognition*, volume 69 of *Springer Handbook of Auditory Research*. Springer International Publishing, Cham, 2019. ISBN 978-3-030-14831-7 978-3-030-14832-4. doi: 10.1007/978-3-030-14832-4. URL http://link.springer.com/10.1007/978-3-030-14832-4.

[172] Jacob Sobolev. Les Paul Vs Stratocaster - Which One Is More Suited For You?, 2024. URL https://rockguitaruniverse.com/les-paul-vs-stratocaster/.

[173] Kihyuk Sohn, Honglak Lee, and Xinchen Yan. Learning structured output representation using deep conditional generative models. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015. URL https://proceedings.neurips.cc/paper_files/paper/2015/file/8d55a249e6baa5c06772297520da2051-Paper.pdf.

[174] Ryan Stables, Sean Enderby, Brecht De Man, György Fazekas, and Joshua D. Reiss. SAFE: A System for the Extraction and Retrieval of Semantic Audio Descriptors. In *Proceedings of the 15th International Society for Music Information Retrieval*, Taipei, Taiwan, October 2014. URL https://www.open-access.bcu.ac.uk/id/eprint/3255.

[175] Ryan Stables, Brecht De Man, Sean Enderby, Joshua D. Reiss, György Fazekas, and Thomas Wilmering. Semantic Description of Timbral Transformations in Music Production. In *Proceedings of the 24th ACM International Conference on Multimedia*, pages 337–341, Amsterdam The Netherlands, October 2016. ACM. ISBN 978-1-4503-3603-1. doi: 10.1145/2964284.2967238. URL https://dl.acm.org/doi/10.1145/2964284.2967238.

[176] Gary E. Starr and Mark A. Pitt. Interference effects in short-term memory for timbre. *The Journal of the Acoustical Society of America*, 102(1):486–494, July 1997. ISSN 0001-4966, 1520-8524. doi: 10.1121/1.419722. URL https://pubs.aip.org/jasa/article/102/1/486/557422/Interference-effects-in-short-term-memory-for.

[177] J F Steffensen. *Interpolation*. Dover Books on Mathematics. Dover Publications, Mineola, NY, 2 edition, March 2006.

[178] Michael Stein. Automatic detection of multiple, cascaded audio effects in guitar recordings. In *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx2010)*, 2010. URL https://dafx.de/paper-archive/2010/DAFx10/Stein_DAFx10_P18.pdf.

[179] Michael Stein, Jakob Abeßer, Christian Dittmar, and Gerald Schuller. Automatic detection of audio effects in guitar and bass recordings. *Journal of the Audio Engineering Society Convention 128*, May 2010. URL https://www.aes.org/e-lib/browse.cfm?elib=15310.

[180] Steinberg. Our Technologies, 2024. URL https://www.steinberg.net/technology/.

[181] Steinberg. What is a DAW? - A Guide To The Digital Audio Workstation, 2024. URL https://www.steinberg.net/tutorials/what-is-a-daw/.

[182] Nick Stoubis. Timbre: How to Change Your Tone With Your Picking Hand, 2023. URL https://www.fender.com/articles/techniques/timbre-how-to-change-your-tone-with-your-picking-hand#:~:text=Changes%20in%20timbre%20are%20not,available%20through%20pedals%20and%20presets.

[183] Makoto Takeuchi and Haruo Saito. Absolute measurement of sampling jitter in audio equipment. *The Journal of the Acoustical Society of America*, 154(1):443–453, 07 2023. ISSN 0001-4966. doi: 10.1121/10.0020291. URL https://doi.org/10.1121/10.0020291.

[184] Hossein Talebi and Peyman Milanfar. Learning to resize images for computer vision tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 497–506, October 2021.

[185] George Tanev and Adrijan Božinovski. Virtual Studio Technology inside Music Production. In Vladimir Trajkovik and Misev Anastas, editors, *ICT Innovations 2013*, volume 231, pages 231–241. Springer International Publishing, Heidelberg, 2014. ISBN 978-3-319-01465-4 978-3-319-01466-1. doi: 10.1007/978-3-319-01466-1_22. URL https://link.springer.com/10.1007/978-3-319-01466-1_22. Series Title: Advances in Intelligent Systems and Computing.

[186] John Taylor. Fender Vs Gibson: What Are the Differences Between Them?, February 2020. URL https://www.reidys.com/blog/fender-vs-gibson-what-are-the-differences-between-them/.

[187] Editorial Team. Les Paul Guitars: What's So Special About Them?, 2023. URL https://www.geartalk.com/les-paul-guitar/.

[188] TensorFlow. TensorFlow 2.15.0 Documentation, November 2023. URL https://www.tensorflow.org/versions/r2.15/api_docs/python/tf#tensorflow.

[189] TensorFlow and Keras. Keras 2 ImageDataGenerator: TensorFlow 2 API Documentation, December 2023. URL https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator.

[190] M. Tenzer and J. Roeder. *Analytical and Cross-Cultural Studies in World Music*. Oxford University Press, USA, 2011. ISBN 9780195384581. URL https://books.google.co.uk/books?id=7FtryMm3gggC.

[191] Hoang Thanh-Tung and Truyen Tran. Catastrophic forgetting and mode collapse in gans. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–10, 2020. doi: 10.1109/IJCNN48605.2020.9207181.

[192] Play Music Today. Big Top Creepy Clown Mini Fuzz Pedal. https://www.pmtonline.co.uk/big-top-creepy-clown-mini-fuzz-pedal, 2023. Accessed: 07/12/2023.

[193] Play Music Today. PMT (Play Music Today) Music Shop Cambridge. https://www.pmtonline.co.uk/stores/cambridge/, 2023. Accessed: 07/12/2023.

[194] Play Music Today. Play Music Today Online, 2024. URL https://www.pmtonline.co.uk.

[195] B. Tolinski, A. di Perna, and C. Santana. *Play It Loud: An Epic History of the Style, Sound, and Revolution of the Electric Guitar*. Knopf Doubleday Publishing Group, 2016. ISBN 9780385541008. URL https://books.google.co.uk/books?id=o3J_CwAAQBAJ.

[196] Jeorge Tripps. Way Huge Atreides Analog Weirding Module. https://www.jimdunlop.com/way-huge-atreides-analog-weirding-module/, 2023. Accessed: 07/12/2023.

[197] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alexander Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *Arxiv*, 2016. doi: 10.48550/ARXIV.1609.03499. URL https://arxiv.org/abs/1609.03499.

[198] Aaron van den Oord, Oriol Vinyals, and Koray Kavukcuoglu. Neural discrete representation learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6309–6318, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

[199] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL http://jmlr.org/papers/v9/vandermaaten08a.html.

[200] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

[201] G. von Bismarck. Timbre of Steady Sounds: A Factorial Investigation of its Verbal Attributes. *Acta Acustica united with Acustica*, 30(3):146–159, March 1974. URL https://www.ingentaconnect.com/content/dav/aaua/1974/00000030/00000003/art00005.

[202] V. Välimäki, S. Bilbao, J. O. Smith, J. S. Abel, J. Pakarinen, and D. Berners. *Virtual Analog Effects*, chapter 12, pages 473–522. John Wiley and Sons, Ltd, 2011. ISBN 9781119991298. doi: https://doi.org/10.1002/9781119991298.ch12. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/9781119991298.ch12.

[203] Jiayu Wang, Wengang Zhou, Jinhui Tang, Zhongqian Fu, Qi Tian, and Houqiang Li. Unregularized auto-encoder with generative adversarial networks for image generation. In *Proceedings of the 26th ACM International Conference on Multimedia*, MM '18, page 709–717, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450356657. doi: 10.1145/3240508.3240569. URL https://doi.org/10.1145/3240508.3240569.

[204] Jeff R. Warren. "That worship sound": ethics, things, and shimmer reverberation. In Nathan Myrick and Mark James Porter, editors, *Ethics and Christian Musicking*, pages 73–82. Routledge, Abingdon, Oxon, 2021. ISBN 978-1-00-036012-7. OCLC: 1235278700.

[205] Wes McKinney. Data Structures for Statistical Computing in Python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 56 – 61, 2010. doi: 10.25080/Majora-92bf1922-00a.

[206] Paul White. Make Your Guitar Sound Shimmer, July 2010. URL https://www.soundonsound.com/techniques/make-your-guitar-sound-shimmer.

[207] Paul White. Creating Shimmer Reverb Effects, July 2020. URL https://www.soundonsound.com/techniques/creating-shimmer-reverb-effects.

[208] Tom White. Sampling Generative Networks, 2016. URL https://arxiv.org/abs/1609.04468. Version Number: 3.

[209] Thomas Wilmering, David Moffat, Alessia Milo, and Mark B. Sandler. A history of audio effects. *Applied Sciences*, 10(3), 2020. ISSN 2076-3417. doi: 10.3390/app10030791. URL https://www.mdpi.com/2076-3417/10/3/791.

[210] Thomas Withee. Study of the electric guitar pickup. *Individual study, University of Illinois at Urbana-Chapign*, 2002. URL https://courses.physics.illinois.edu/phys406/sp2017/Student_Projects/Spring02/TWithee/twithee_spr02_indept_study.pdf.

[211] Yuhuai Wu, Yuri Burda, Ruslan Salakhutdinov, and Roger Grosse. On the quantitative analysis of decoder-based generative models. In *Proceedings of the International Conference on Learning Representations 2017*, 2017. URL https://openreview.net/forum?id=B1M8JF9xx.

[212] William A. Yost. *Fundamentals of Hearing: An Introduction: Fifth Edition*. Brill, January 2021. ISBN 9780123704733.

[213] Asterios Zacharakis, Konstantinos Pastiadis, Joshua D. Reiss, and George Papadelis. Analysis of Musical Timbre Semantics through Metric and Non-Metric Data Reduction Techniques. In *Proceedings of the 12th International Conference on Music Perception and Cognition (ICMPC) & 8th Triennial Conference of the European Society for the Cognitive Sciences of Music (ESCOM)*, Thessaloniki, Greece, July 2012. URL http://eecs.qmul.ac.uk/~josh/documents/2012/ZacharakisPastiadisReissetal-musicaltimbre-ICMPC2012.pdf.

[214] Asterios Zacharakis, Konstantinos Pastiadis, and Joshua D. Reiss. An interlanguage study of musical timbre semantic dimensions and their acoustic correlates. *Music Perception*, 31(4):339–358, 04 2014. ISSN 0730-7829. doi: 10.1525/mp.2014.31.4.339. URL https://doi.org/10.1525/mp.2014.31.4.339.

[215] Asterios Zacharakis, Konstantinos Pastiadis, and Joshua D. Reiss. An interlanguage unification of musical timbre: Bridging semantic, perceptual, and acoustic dimensions. *Music Perception*, 32(4):394–412, 04 2015. ISSN 0730-7829. doi: 10.1525/mp.2015.32.4.394. URL https://doi.org/10.1525/mp.2015.32.4.394.

[216] Asteris I. Zacharakis, Konstantinos Pastiadis, Georgios Papadelis, and Joshua D. Reiss. An Investigation of Musical Timbre: Uncovering Salient Semantic Descriptors and Perceptual Dimensions. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 807–812. ISMIR, September 2011. doi: 10.5281/zenodo.1414964. URL https://doi.org/10.5281/zenodo.1414964.

[217] Jingjie Zhang. Shimmer Audio Effect: A Harmonic Reverberator. Final Project Report for Music 421A: Audio Applications of the FFT, Stanford University, 2018. URL https://ccrma.stanford.edu/~jingjiez/portfolio/echoing-harmonics/pdfs/Shimmer%20Audio%20Effect%20-%20A%20Harmonic%20Reverberator.pdf.

Appendix A

Acoustic characteristics of sounds described by the SemanticTimbreDataset's timbre descriptors

For future reference, here are some brief insights into the physical characteristics of sounds that are described by the SemanticTimbreDataset's timbre descriptors (see Table 3.7).

Looking at the DistortionFX descriptors, 'Clean' refers to a natural guitar note that has been untouched by any distortion effects [67], while the other descriptors refer to specific types of distortion [175] applied to guitar notes [67], with 'Fuzzy' being the most extreme [42].

Within the FilterFX descriptors, 'Bright' refers to a sound with emphasised high frequencies [53], 'Dark' describes a sound with emphasised low frequencies [162], 'Fat' refers to a sound with a large range of frequencies emphasised [84, 121], 'Thin' refers to a sound with a narrow range of frequencies remaining [63], and 'Resonant' refers to a sound that contains a very loud narrow band of frequencies [63].

The DynamicsFX descriptors generally refer to the ADSR envelope [65, 51, 125, 82] of guitar notes. 'Punchy' describes a note with a loud transient, short attack time, and short sustain [51, 84]. 'Sharp' describes a note with a loud transient, short attack time, and long sustain [82, 84]. 'Soft' describes a note with a quiet transient due to a long attack time, and short sustain [41, 84]. 'Smooth' describes a note with a quiet transient due to a long attack time, and long sustain [82, 84]. 'Tight' refers to a note that has very little dynamic range [118].

Referring to the OscillationFX descriptors, 'Fluttery' describes a note that has quick variations in pitch [134, 159, 84], 'Jittery' describes a note that contains quick oscillations of phase noise [183, 85, 84], 'Shimmering' describes a note that is mixed with extremely

quick oscillations of a pitch-shifted version of itself [217, 207, 206], 'Stuttering' describes a note that has quick variations in intensity/amplitude [84], and 'WahWah' describes a note that has slow variations in pitch [190] and is called the 'WahWah' effect [161].

Appendix B

Spectrogram images of example audio files from the SemanticTimbreDataset

Figures B.1-B.20 show examples of audio samples visualised as spectrograms from every one of the 20 timbre descriptor groups (see Table 3.7 for the 20 timbre descriptors) within the SemanticTimbreDataset. For conciseness, every figure displays a spectrogram of solely the E4 note played on a Fender Stratocaster electric guitar. The 'Clean' note displayed in Figure B.1 reflects a note with a timbre descriptor magnitude of 0. For the other 19 timbre descriptor groups in Figures B.2-B.20, two E4 notes are displayed, where one has a timbre magnitude of 50 and the other has a timbre magnitude of 100.

These spectrogram images, and all spectrogram images visualising all 275,310 audio files within the SemanticTimbreDataset, can be downloaded online [20].

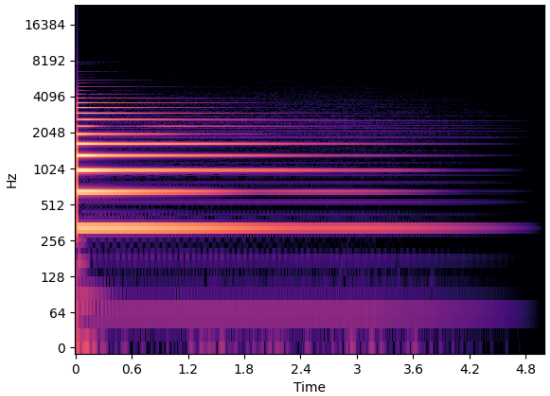
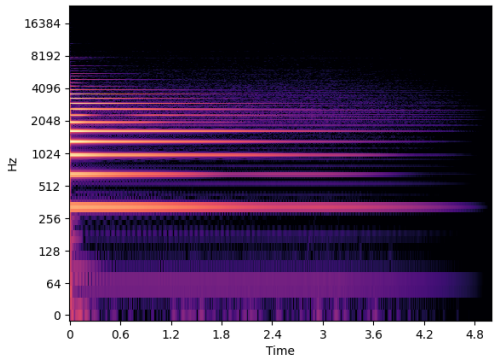
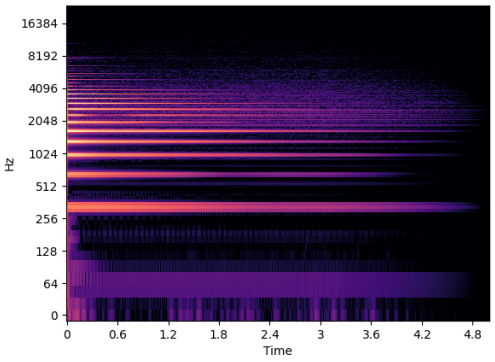


Figure B.1: 'Clean' E4 note in the SemanticTimbreDataset.

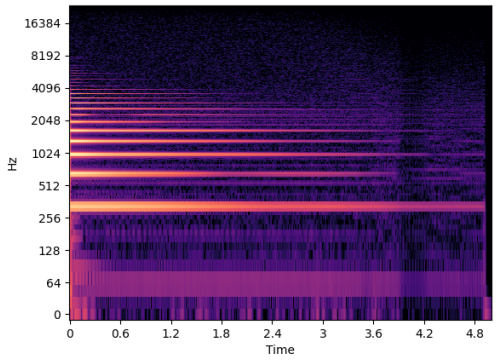


(a) 50/100 'Bright' Magnitude

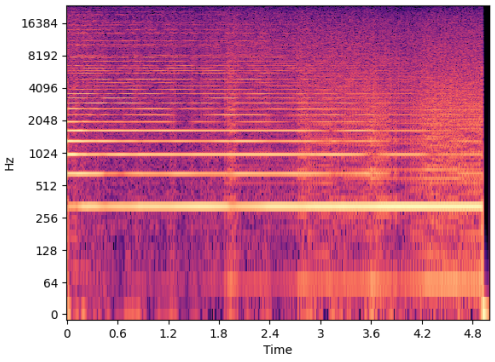


(b) 100/100 'Bright' Magnitude

Figure B.2: 'Bright' E4 notes in the SemanticTimbreDataset.

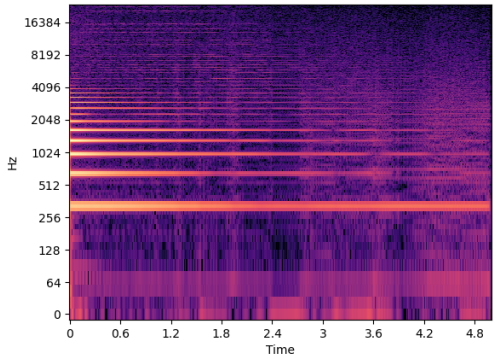


(a) 50/100 'Crunch' Magnitude

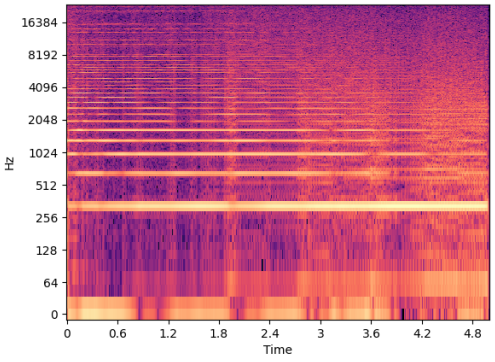


(b) 100/100 'Crunch' Magnitude

Figure B.3: 'Crunchy' E4 notes in the SemanticTimbreDataset.

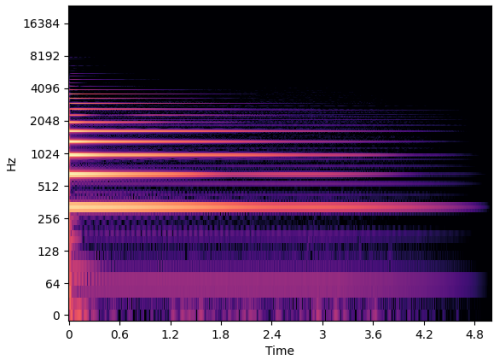


(a) 50/100 'Crush' Magnitude

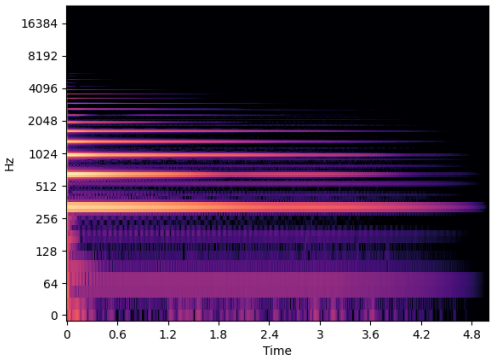


(b) 100/100 'Crush' Magnitude

Figure B.4: 'Crushed' E4 notes in the SemanticTimbreDataset.

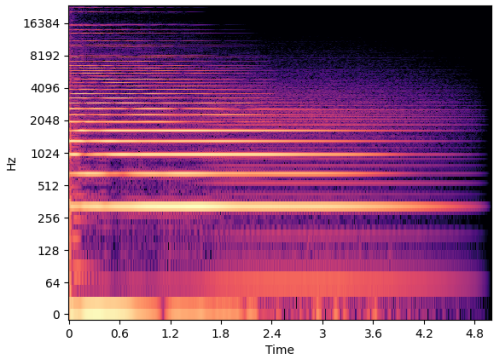


(a) 50/100 ‘Dark’ Magnitude

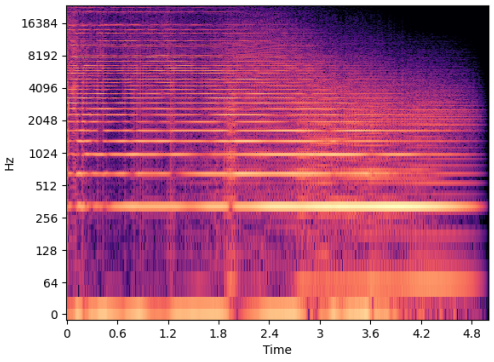


(b) 100/100 ‘Dark’ Magnitude

Figure B.5: ‘Dark’ E4 notes in the SemanticTimbreDataset.

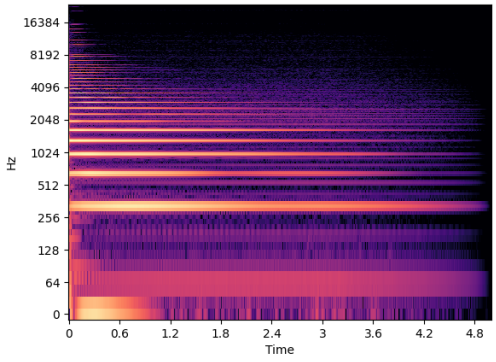


(a) 50/100 ‘Dirt’ Magnitude

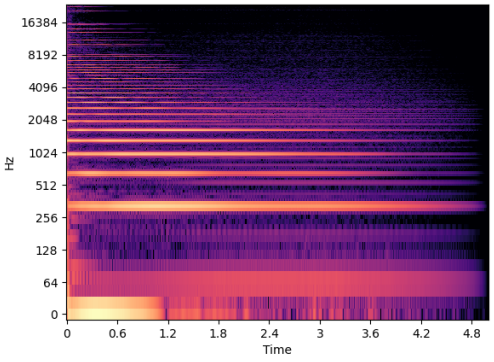


(b) 100/100 ‘Dirt’ Magnitude

Figure B.6: ‘Dirty’ E4 notes in the SemanticTimbreDataset.

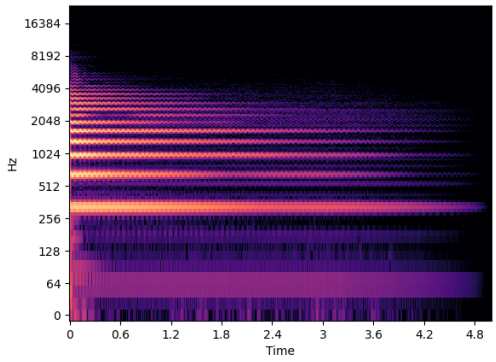


(a) 50/100 ‘Fat’ Magnitude

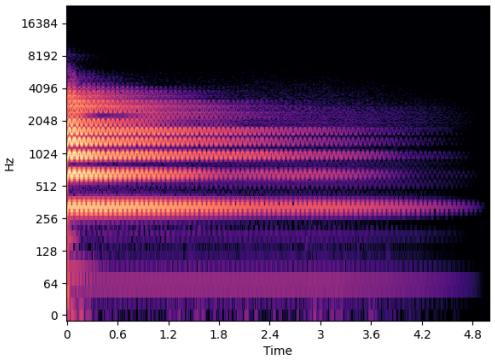


(b) 100/100 ‘Fat’ Magnitude

Figure B.7: ‘Fat’ E4 notes in the SemanticTimbreDataset.

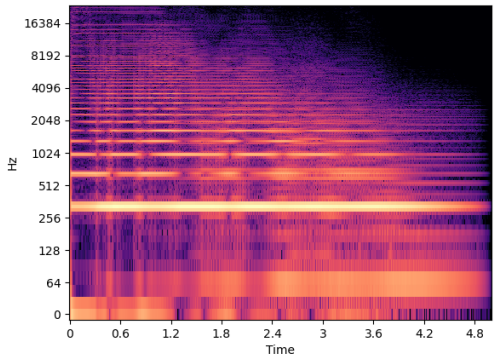


(a) 50/100 ‘Flutter’ Magnitude

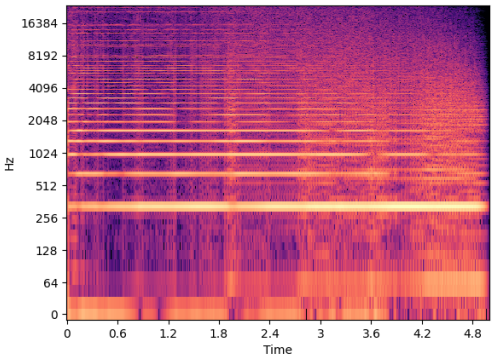


(b) 100/100 ‘Flutter’ Magnitude

Figure B.8: ‘Fluttery’ E4 notes in the SemanticTimbreDataset.

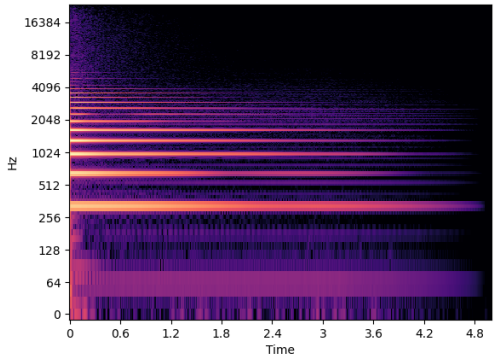


(a) 50/100 ‘Fuzz’ Magnitude

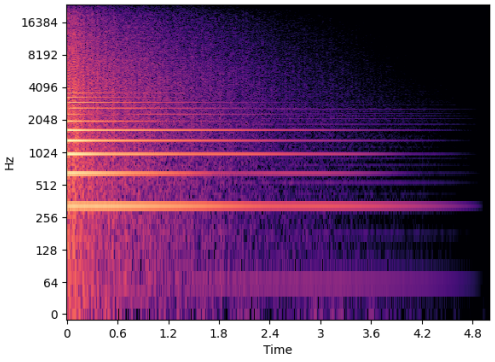


(b) 100/100 ‘Fuzz’ Magnitude

Figure B.9: ‘Fuzzy’ E4 notes in the SemanticTimbreDataset.

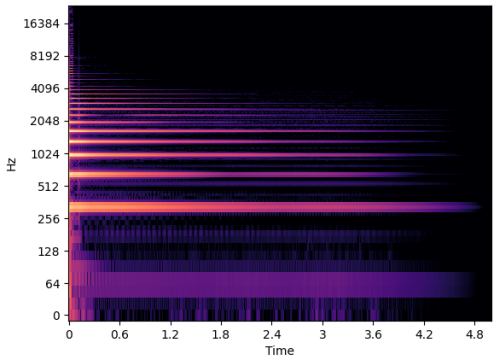


(a) 50/100 ‘Jitter’ Magnitude

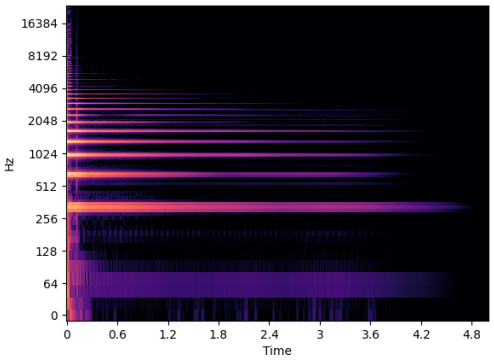


(b) 100/100 ‘Jitter’ Magnitude

Figure B.10: ‘Jittery’ E4 notes in the SemanticTimbreDataset.

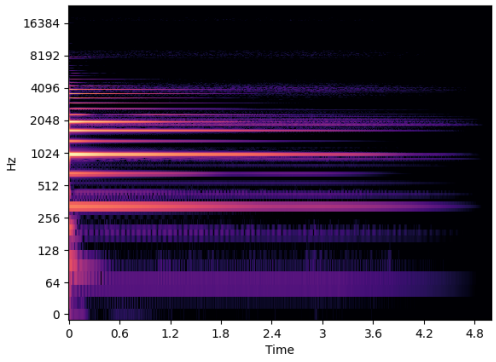


(a) 50/100 'Punch' Magnitude

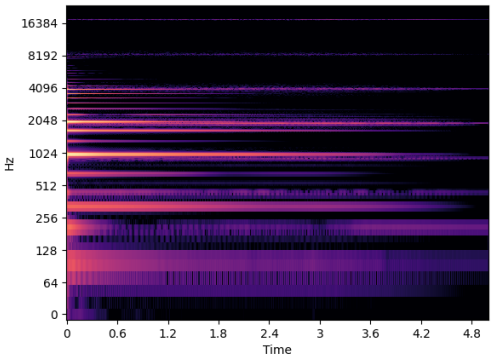


(b) 100/100 'Punch' Magnitude

Figure B.11: 'Punchy' E4 notes in the SemanticTimbreDataset.

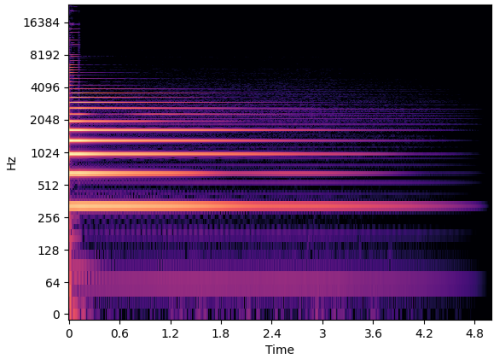


(a) 50/100 'Resonant' Magnitude

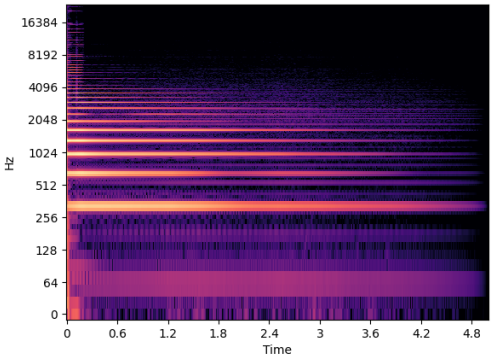


(b) 100/100 'Resonant' Magnitude

Figure B.12: 'Resonant' E4 notes in the SemanticTimbreDataset.

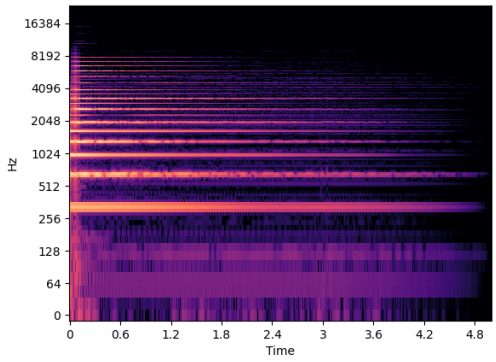


(a) 50/100 'Sharp' Magnitude

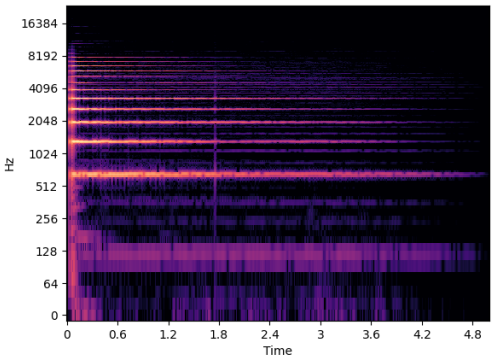


(b) 100/100 'Sharp' Magnitude

Figure B.13: 'Sharp' E4 notes in the SemanticTimbreDataset.

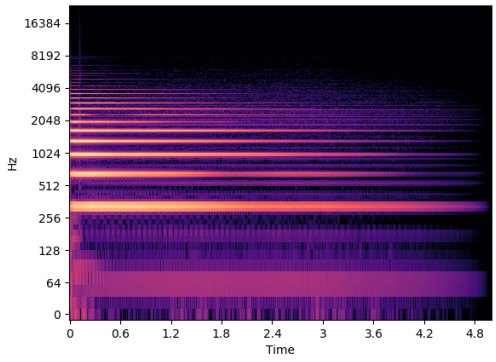


(a) 50/100 ‘Shimmer’ Magnitude

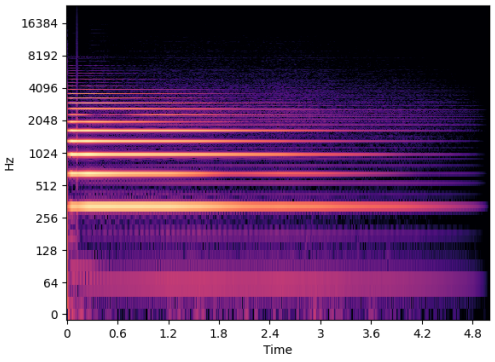


(b) 100/100 ‘Shimmer’ Magnitude

Figure B.14: ‘Shimmering’ E4 notes in the SemanticTimbreDataset.

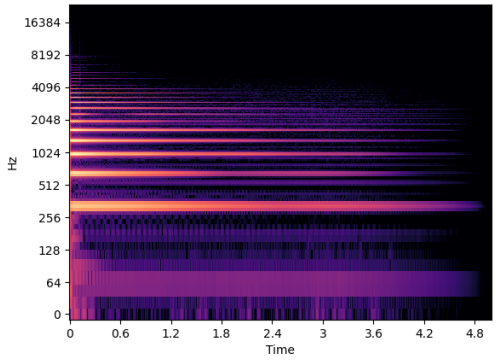


(a) 50/100 ‘Smooth’ Magnitude

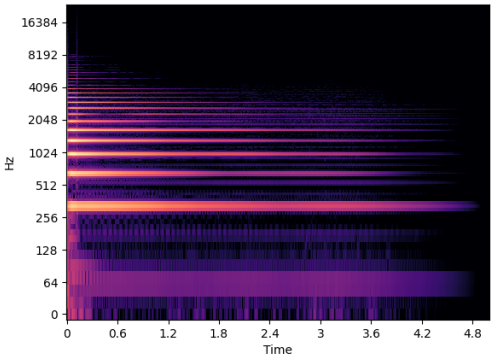


(b) 100/100 ‘Smooth’ Magnitude

Figure B.15: ‘Smooth’ E4 notes in the SemanticTimbreDataset.

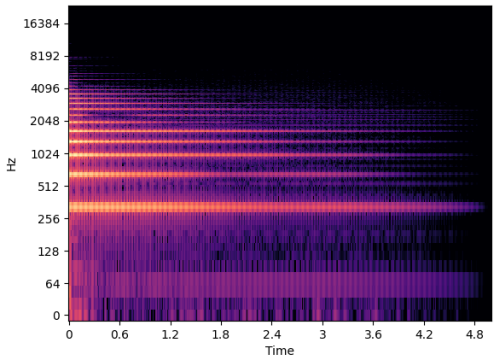


(a) 50/100 ‘Soft’ Magnitude

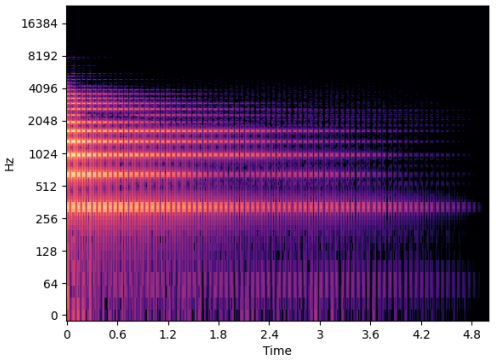


(b) 100/100 ‘Soft’ Magnitude

Figure B.16: ‘Soft’ E4 notes in the SemanticTimbreDataset.

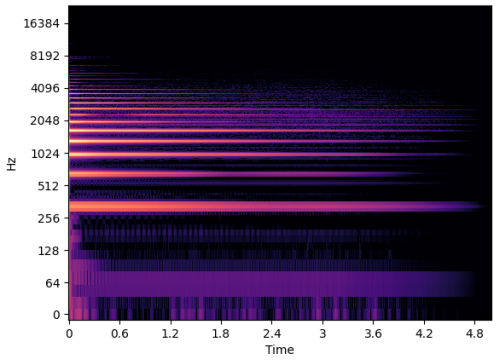


(a) 50/100 'Stutter' Magnitude

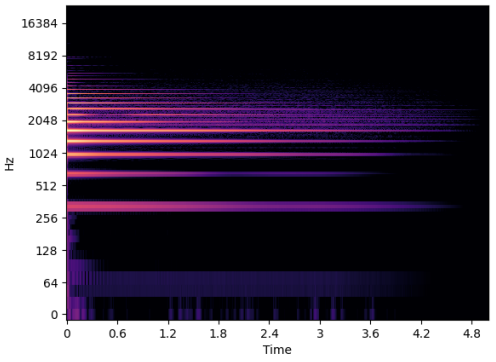


(b) 100/100 'Stutter' Magnitude

Figure B.17: 'Stuttering' E4 notes in the SemanticTimbreDataset.

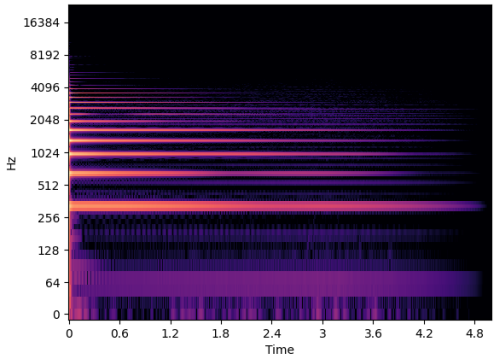


(a) 50/100 'Thin' Magnitude

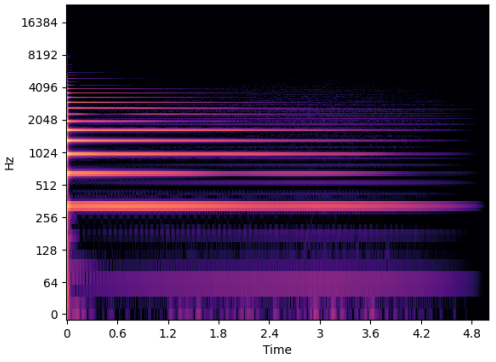


(b) 100/100 'Thin' Magnitude

Figure B.18: 'Thin' E4 notes in the SemanticTimbreDataset.

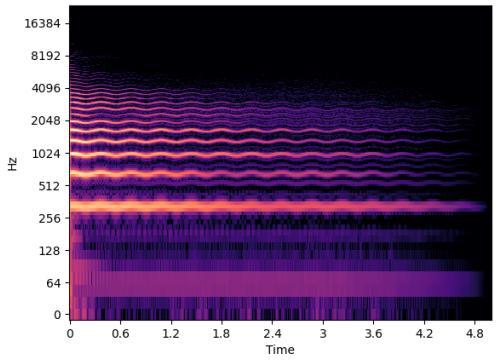


(a) 50/100 'Tight' Magnitude

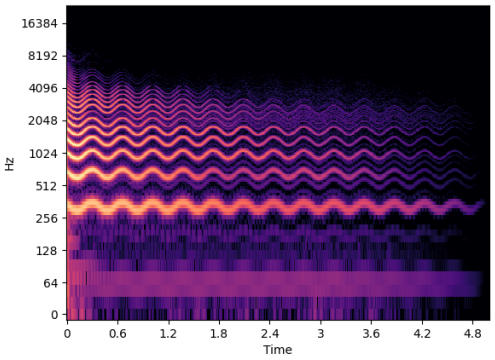


(b) 100/100 'Tight' Magnitude

Figure B.19: 'Tight' E4 notes in the SemanticTimbreDataset.



(a) 50/100 ‘WahWah’ Magnitude



(b) 100/100 ‘WahWah’ Magnitude

Figure B.20: ‘WahWah’ E4 notes in the SemanticTimbreDataset.

Appendix C

Spectrogram images of example audio files generated by the timbre generation system

Figures C.1-C.20 show spectrogram examples of audio samples generated by the timbre generation system (see Chapter 5) when prompted to synthesise E4 guitar notes with timbral characteristics described by every one of the 20 timbre descriptors (see Table 3.7 for the 20 timbre descriptors) within the SemanticTimbreDataset. The 'Clean' note displayed in Figure C.1 reflects an E4 note generated by the timbre generation system with a timbre descriptor magnitude of 0. For the other 19 timbre descriptor groups in Figures C.2-C.20, two E4 notes generated by the timbre generation system are displayed, where one has a timbre magnitude of 50 and the other has a timbre magnitude of 100.

These sounds, and all other sounds generated by the timbre generation system to make up the VAE-GeneratedTestSet (see Section 6.2.1), can be downloaded online [21].

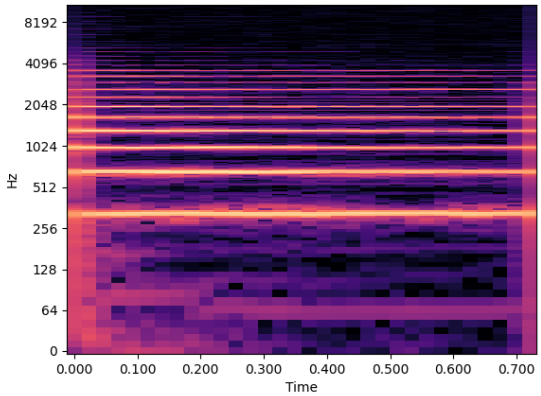
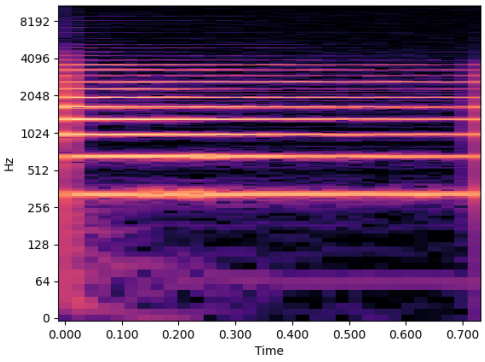
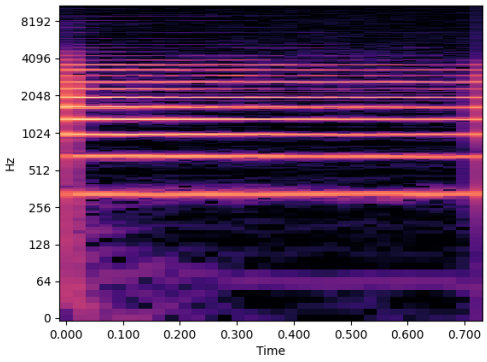


Figure C.1: 'Clean' E4 note generated by the timbre generation system.

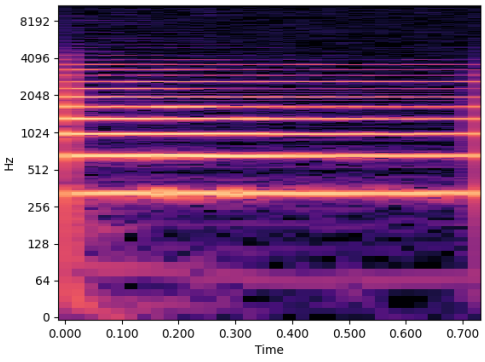


(a) 50/100 'Bright' Magnitude

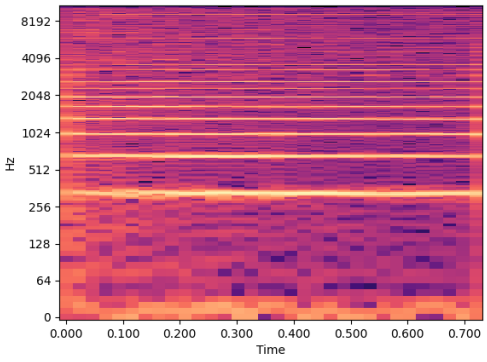


(b) 100/100 'Bright' Magnitude

Figure C.2: 'Bright' E4 notes generated by the timbre generation system.

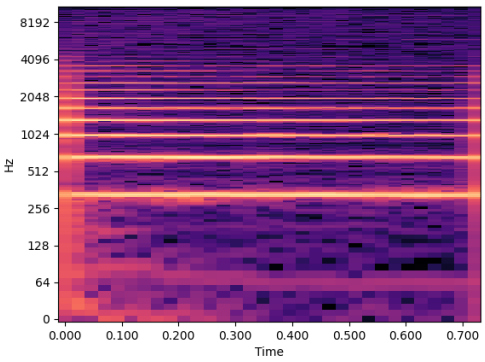


(a) 50/100 'Crunch' Magnitude

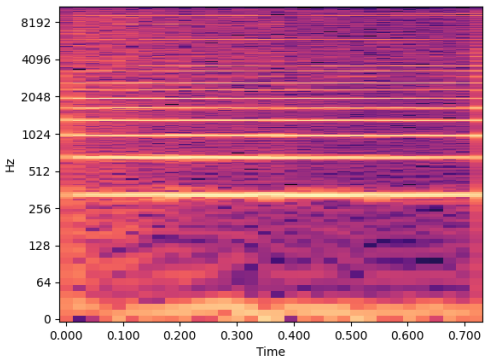


(b) 100/100 'Crunch' Magnitude

Figure C.3: 'Crunchy' E4 notes generated by the timbre generation system.

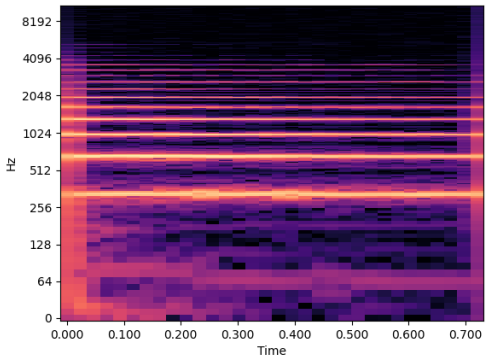


(a) 50/100 'Crush' Magnitude

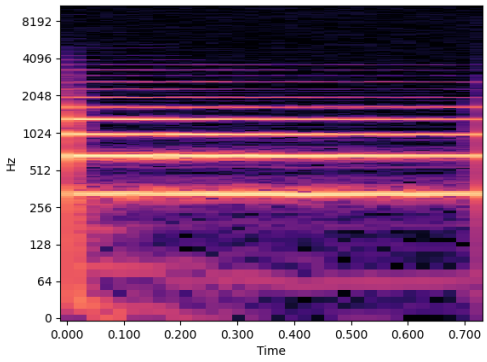


(b) 100/100 'Crush' Magnitude

Figure C.4: 'Crushed' E4 notes generated by the timbre generation system.

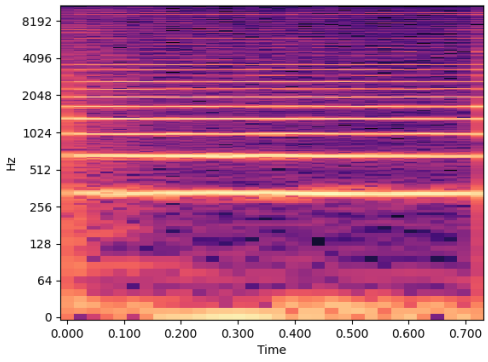


(a) 50/100 ‘Dark’ Magnitude

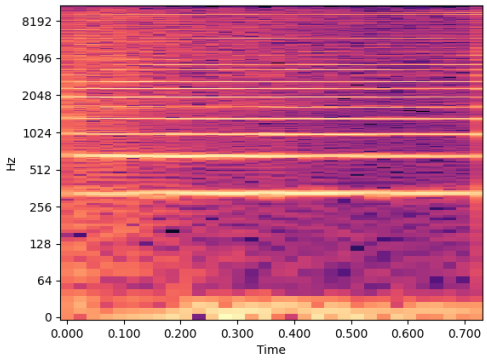


(b) 100/100 ‘Dark’ Magnitude

Figure C.5: ‘Dark’ E4 notes generated by the timbre generation system.

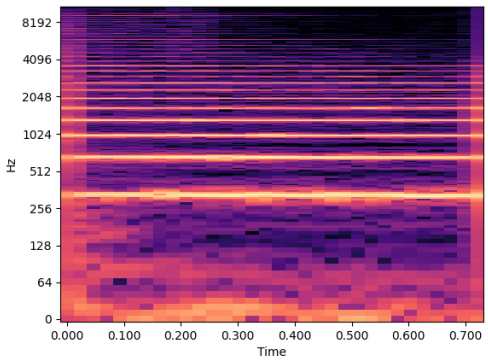


(a) 50/100 ‘Dirt’ Magnitude

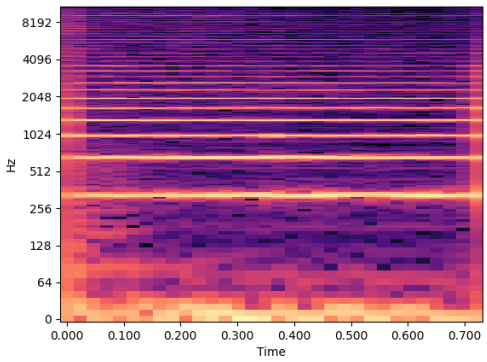


(b) 100/100 ‘Dirt’ Magnitude

Figure C.6: ‘Dirty’ E4 notes generated by the timbre generation system.

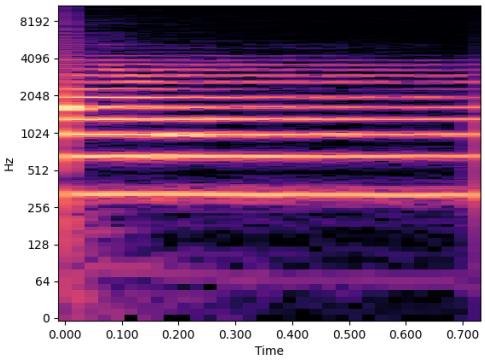


(a) 50/100 ‘Fat’ Magnitude

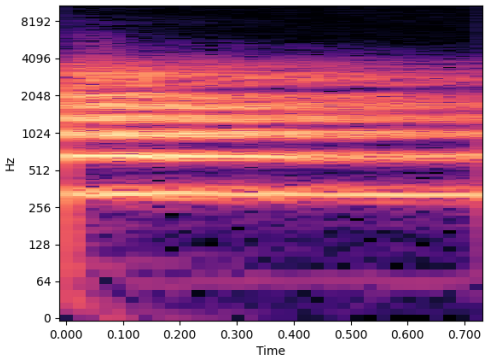


(b) 100/100 ‘Fat’ Magnitude

Figure C.7: ‘Fat’ E4 notes generated by the timbre generation system.

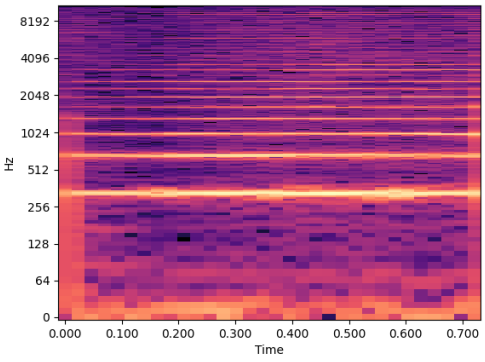


(a) 50/100 ‘Flutter’ Magnitude

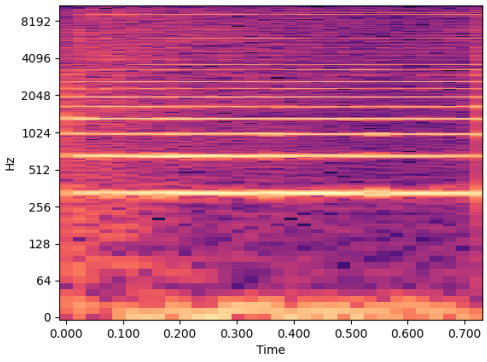


(b) 100/100 ‘Flutter’ Magnitude

Figure C.8: ‘Fluttery’ E4 notes generated by the timbre generation system.

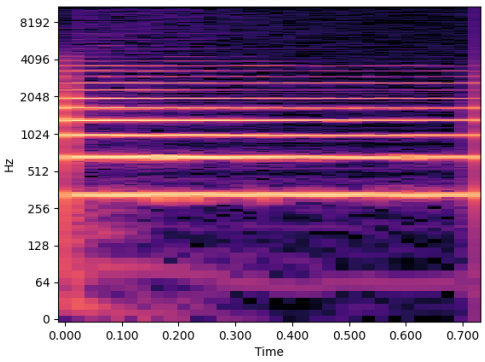


(a) 50/100 ‘Fuzz’ Magnitude

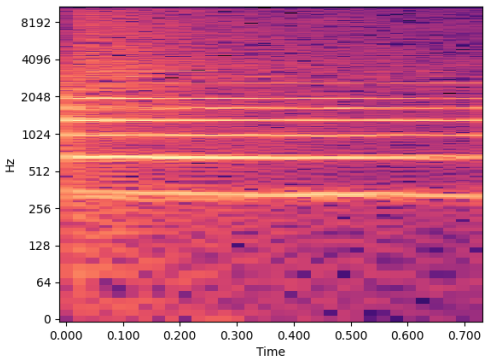


(b) 100/100 ‘Fuzz’ Magnitude

Figure C.9: ‘Fuzzy’ E4 notes generated by the timbre generation system.

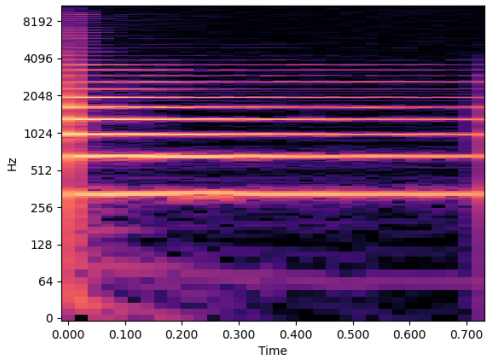


(a) 50/100 ‘Jitter’ Magnitude

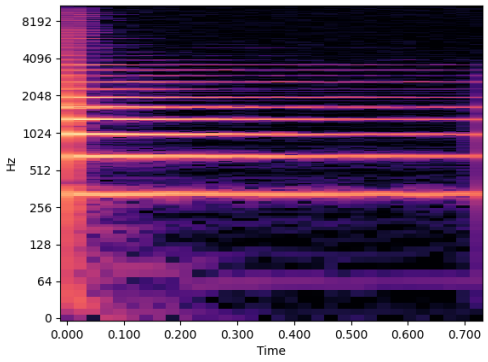


(b) 100/100 ‘Jitter’ Magnitude

Figure C.10: ‘Jittery’ E4 notes generated by the timbre generation system.

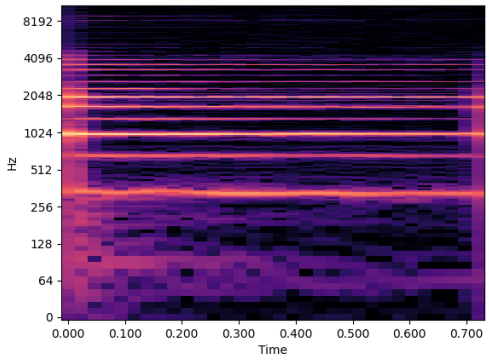


(a) 50/100 ‘Punch’ Magnitude

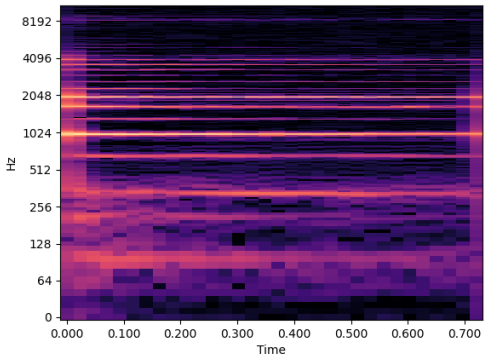


(b) 100/100 ‘Punch’ Magnitude

Figure C.11: ‘Punchy’ E4 notes generated by the timbre generation system.

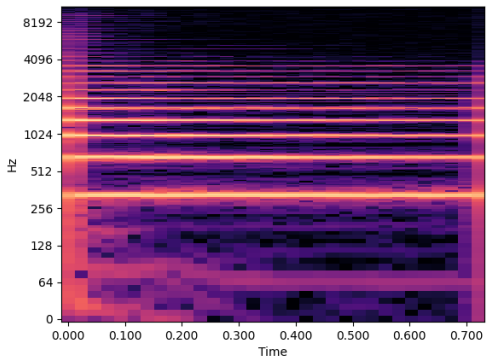


(a) 50/100 ‘Resonant’ Magnitude

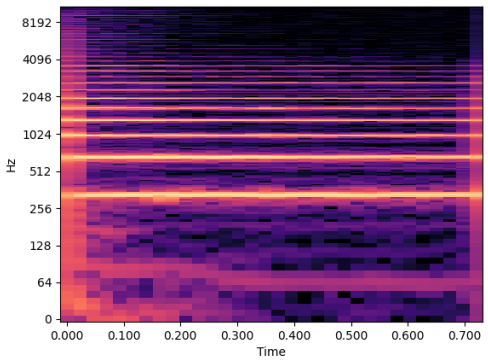


(b) 100/100 ‘Resonant’ Magnitude

Figure C.12: ‘Resonant’ E4 notes generated by the timbre generation system.

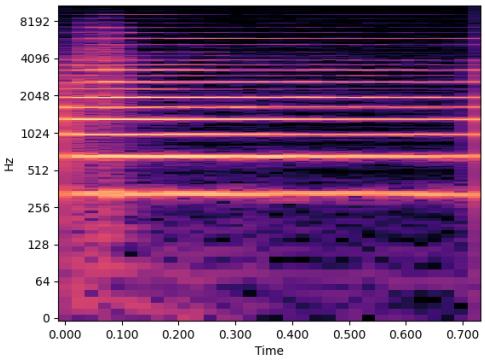


(a) 50/100 ‘Sharp’ Magnitude

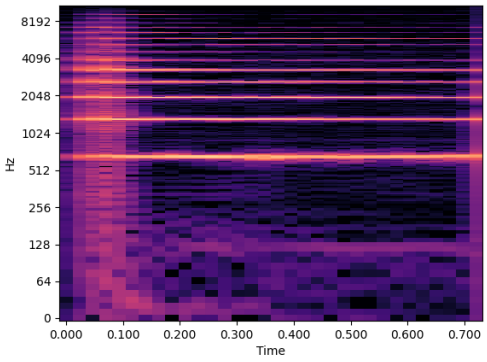


(b) 100/100 ‘Sharp’ Magnitude

Figure C.13: ‘Sharp’ E4 notes generated by the timbre generation system.

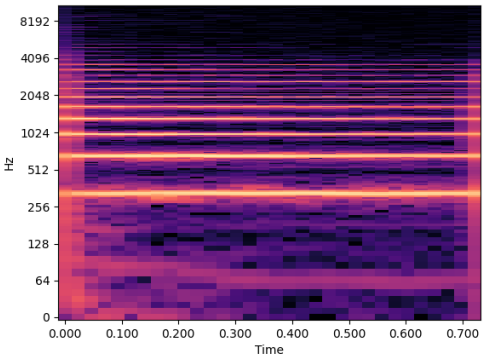


(a) 50/100 ‘Shimmer’ Magnitude

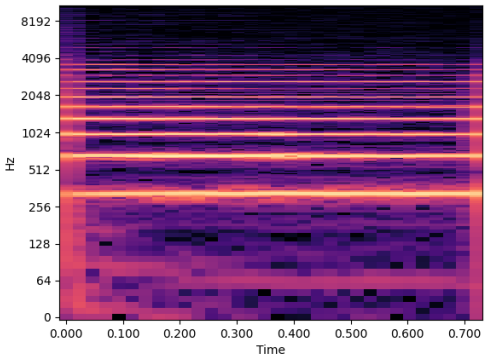


(b) 100/100 ‘Shimmer’ Magnitude

Figure C.14: ‘Shimmering’ E4 notes generated by the timbre generation system.

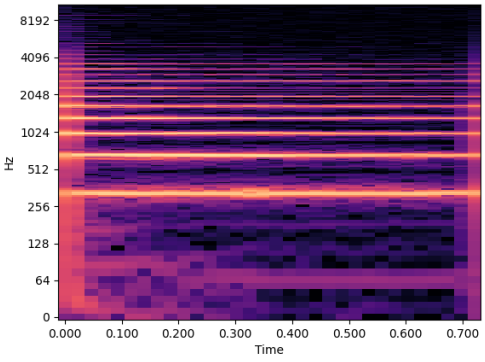


(a) 50/100 ‘Smooth’ Magnitude

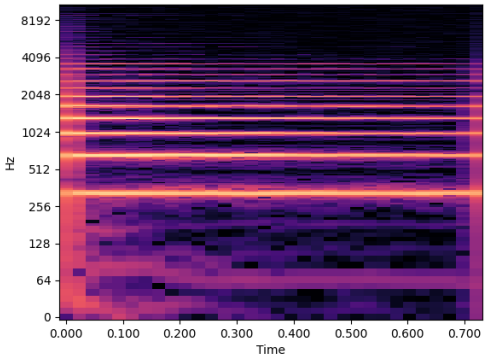


(b) 100/100 ‘Smooth’ Magnitude

Figure C.15: ‘Smooth’ E4 notes generated by the timbre generation system.

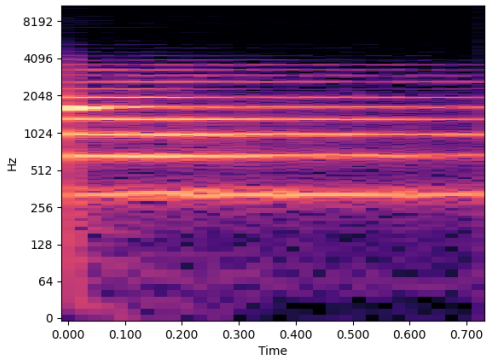


(a) 50/100 ‘Soft’ Magnitude

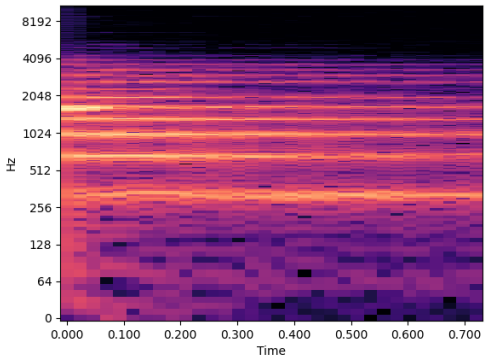


(b) 100/100 ‘Soft’ Magnitude

Figure C.16: ‘Soft’ E4 notes generated by the timbre generation system.

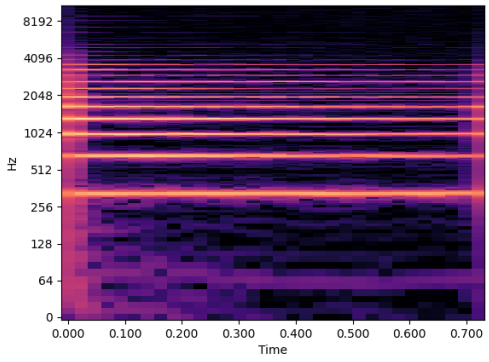


(a) 50/100 ‘Stutter’ Magnitude

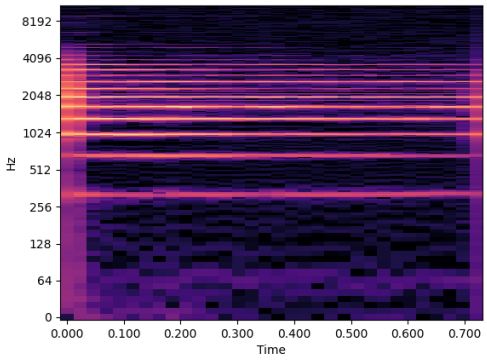


(b) 100/100 ‘Stutter’ Magnitude

Figure C.17: ‘Stuttering’ E4 notes generated by the timbre generation system.

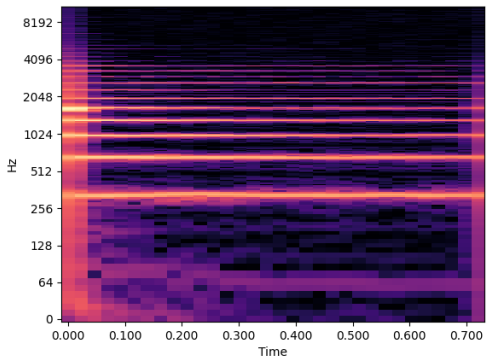


(a) 50/100 ‘Thin’ Magnitude

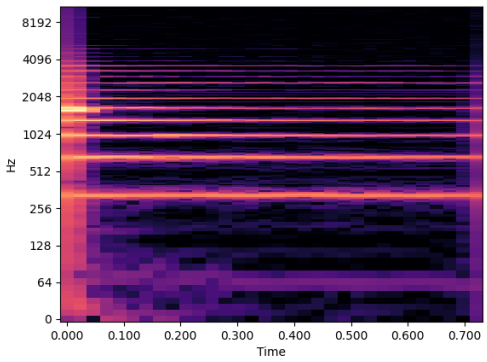


(b) 100/100 ‘Thin’ Magnitude

Figure C.18: ‘Thin’ E4 notes generated by the timbre generation system.

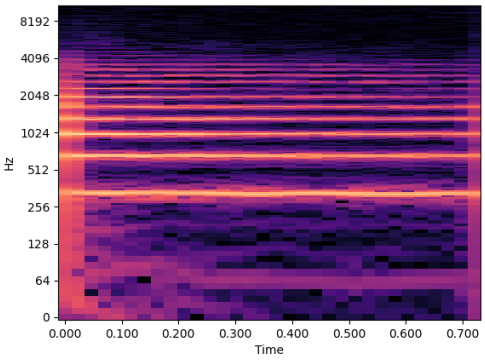


(a) 50/100 ‘Tight’ Magnitude

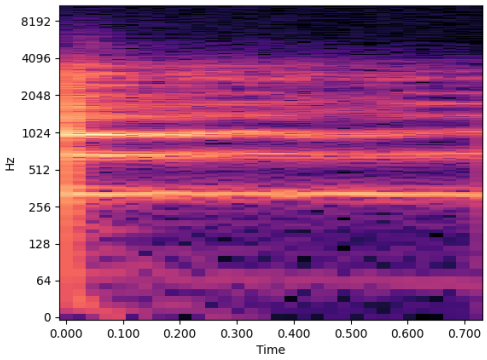


(b) 100/100 ‘Tight’ Magnitude

Figure C.19: ‘Tight’ E4 notes generated by the timbre generation system.



(a) 50/100 ‘WahWah’ Magnitude



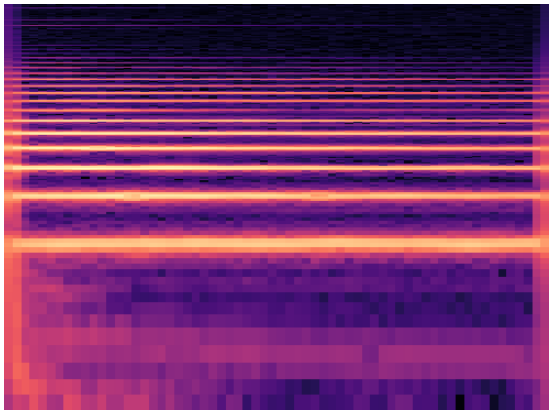
(b) 100/100 ‘WahWah’ Magnitude

Figure C.20: ‘WahWah’ E4 notes generated by the timbre generation system.

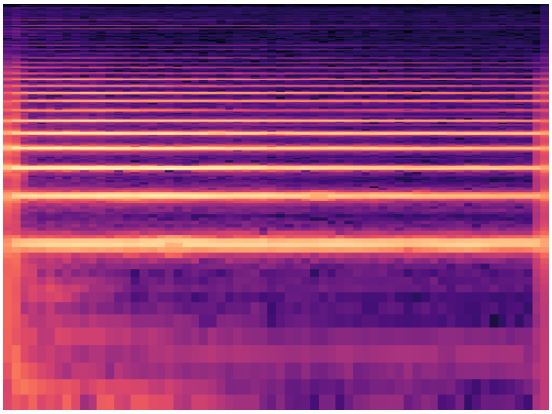
Appendix D

Spectrograms of interpolated audio files generated by the timbre generation system

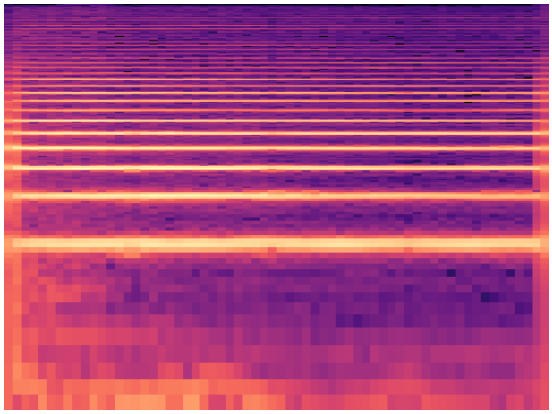
Figures D.1-D.10 illustrate all sounds from the **VAE-InterpolatedTestSet** (see Section 6.3.1) visualised as spectrograms. These sounds from the VAE-InterpolatedTestSet can be downloaded online [21].



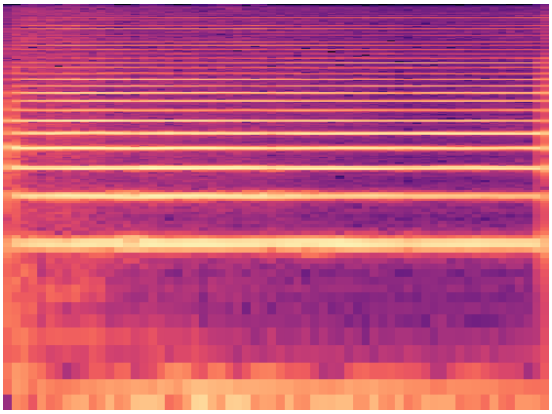
(a) Interpolation Point 1



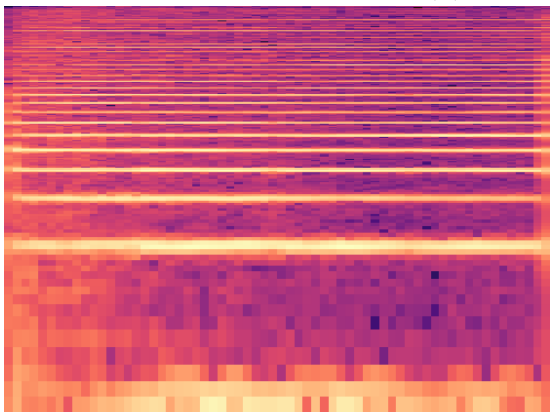
(b) Interpolation Point 2



(c) Interpolation Point 3

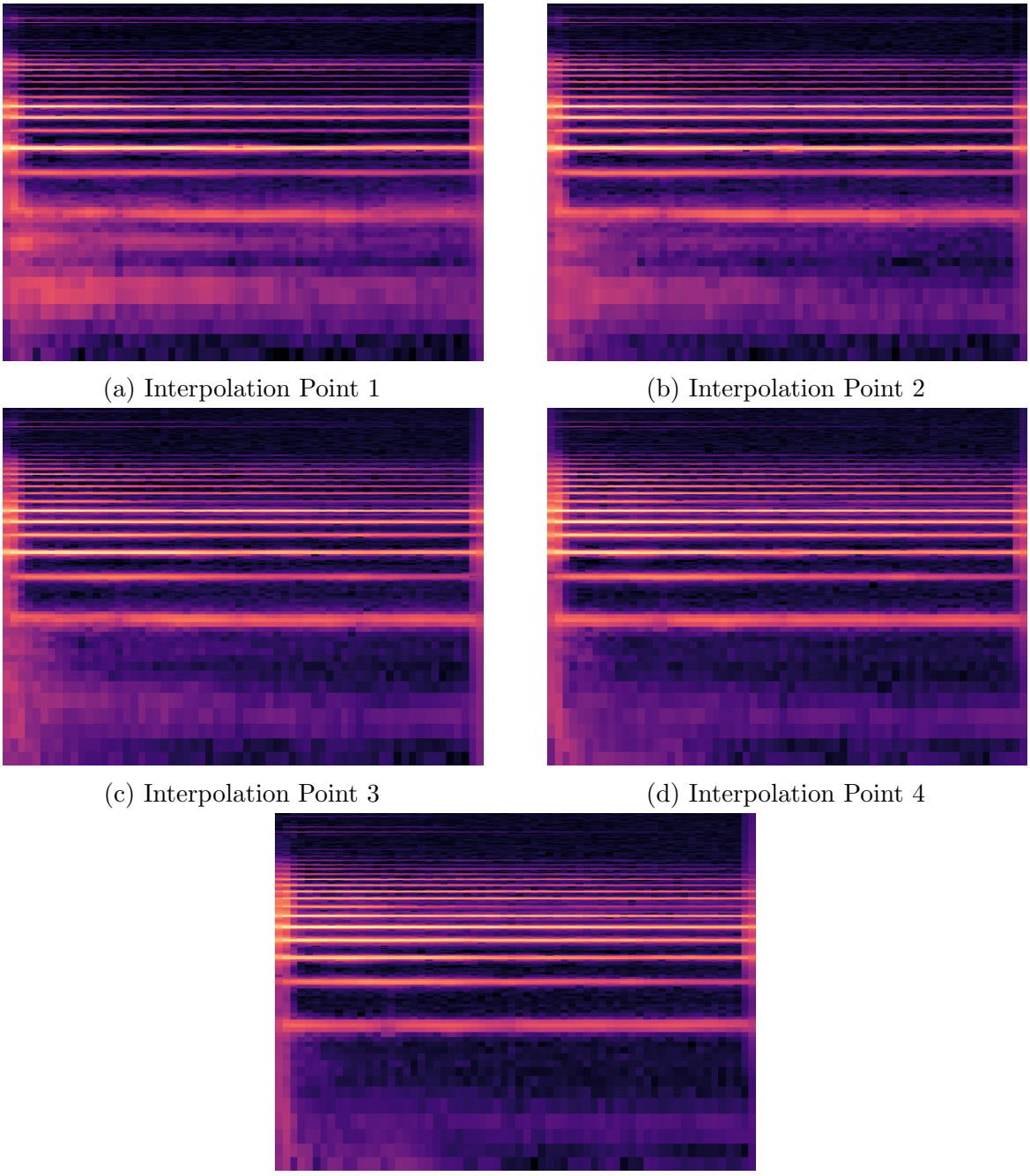


(d) Interpolation Point 4



(e) Interpolation Point 5

Figure D.1: Audio samples for the **Clean-Fuzz 5-Point Interpolation**.



(a) Interpolation Point 1

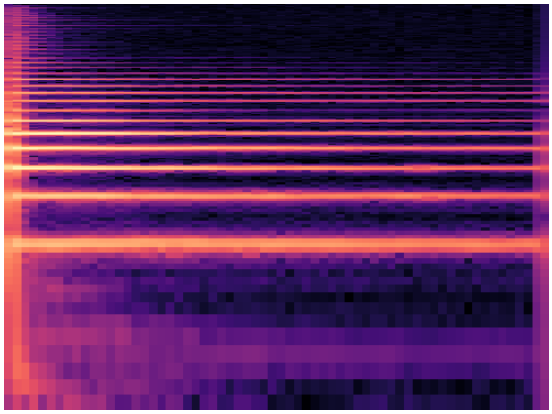
(b) Interpolation Point 2

(c) Interpolation Point 3

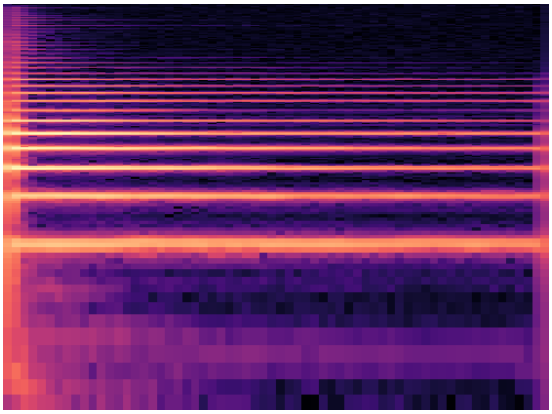
(d) Interpolation Point 4

(e) Interpolation Point 5

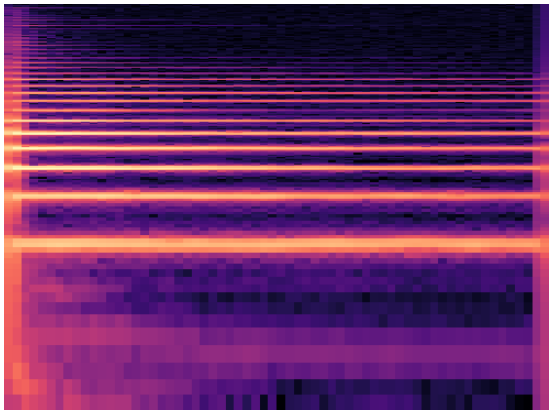
Figure D.2: Audio samples for the **Resonant-Thin 5-Point Interpolation**.



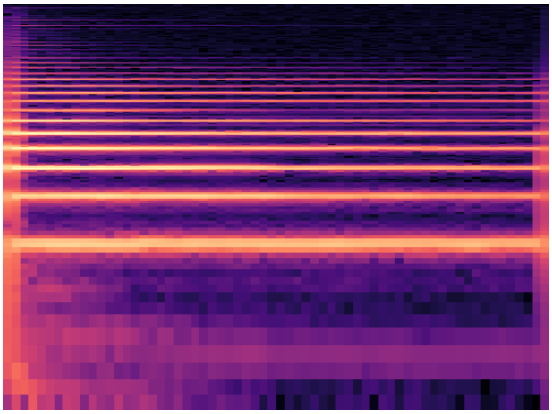
(a) Interpolation Point 1



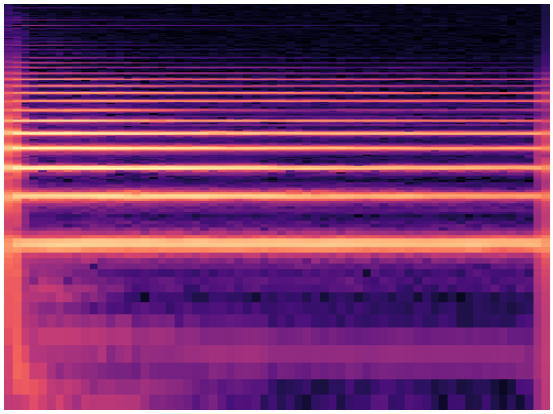
(b) Interpolation Point 2



(c) Interpolation Point 3

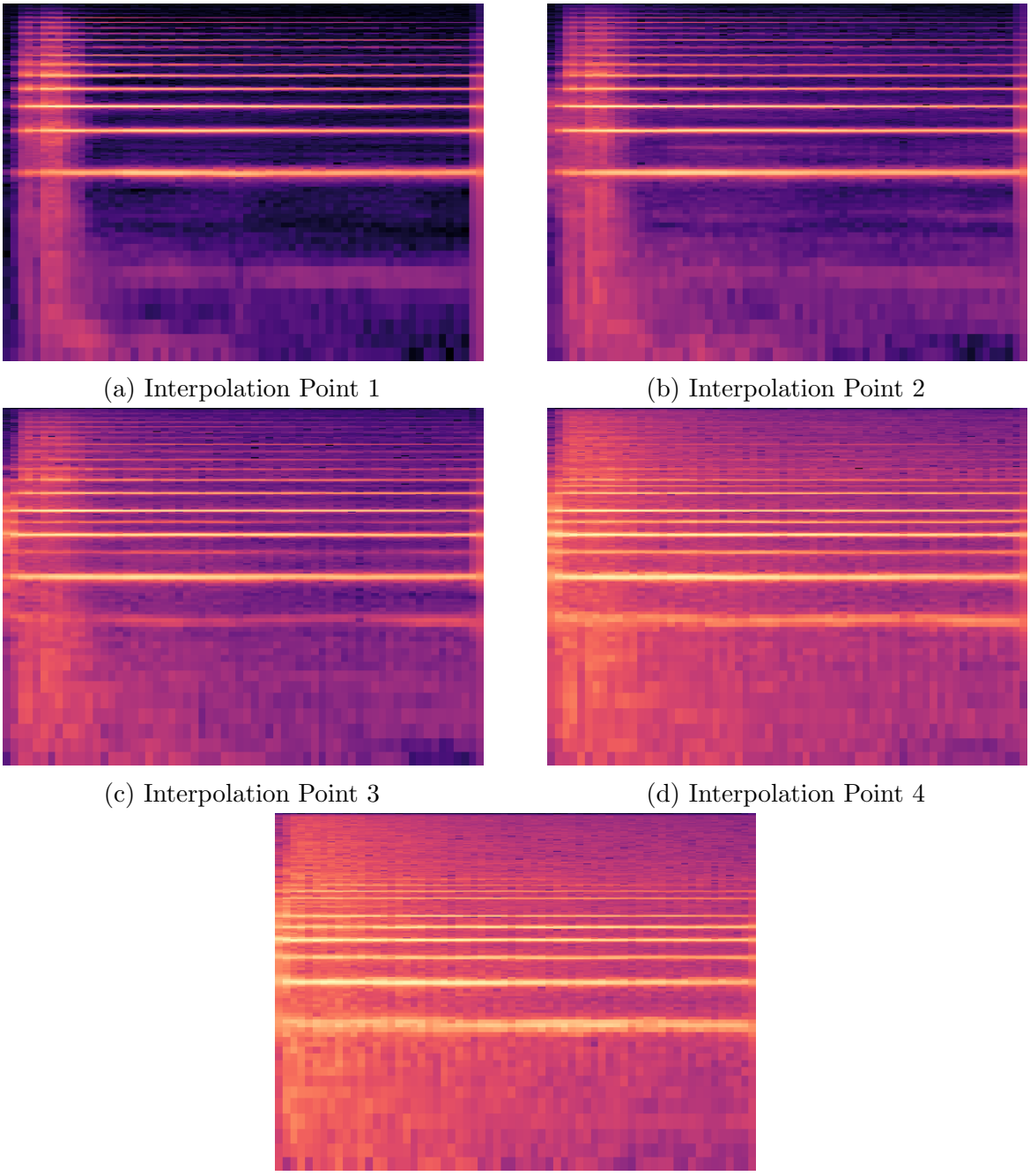


(d) Interpolation Point 4



(e) Interpolation Point 5

Figure D.3: Audio samples for the **Punch-Soft 5-Point Interpolation**.



(a) Interpolation Point 1

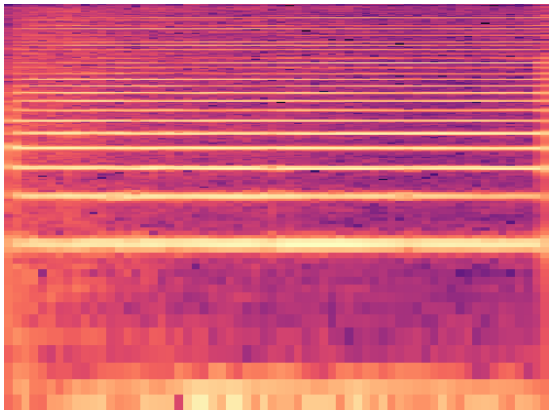
(b) Interpolation Point 2

(c) Interpolation Point 3

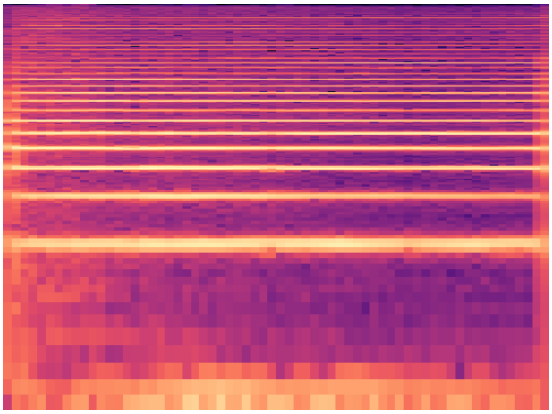
(d) Interpolation Point 4

(e) Interpolation Point 5

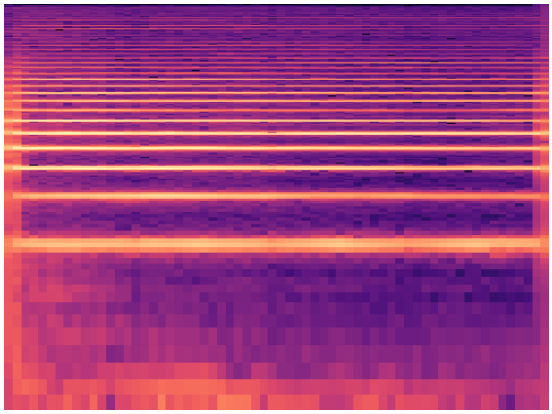
Figure D.4: Audio samples for the **Shimmer-Jitter 5-Point Interpolation**.



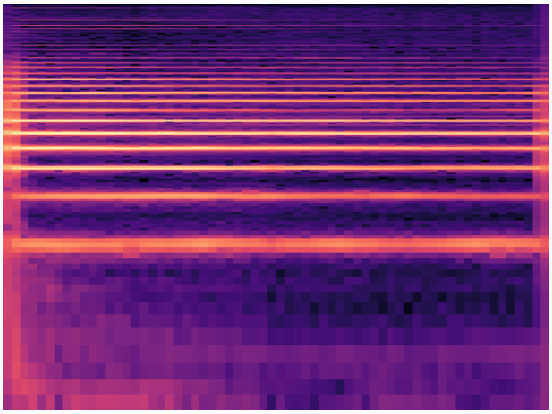
(a) Interpolation Point 1



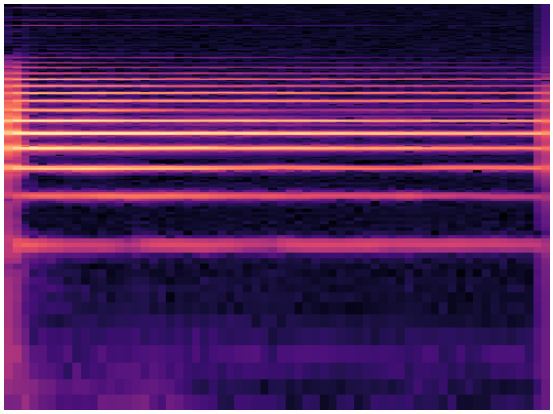
(b) Interpolation Point 2



(c) Interpolation Point 3

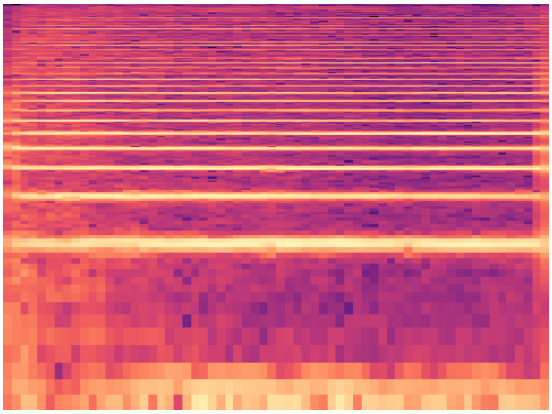


(d) Interpolation Point 4

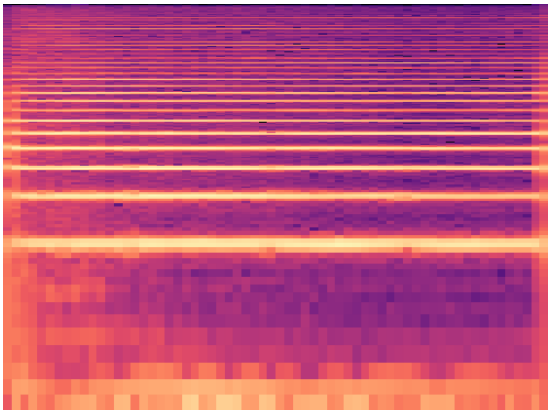


(e) Interpolation Point 5

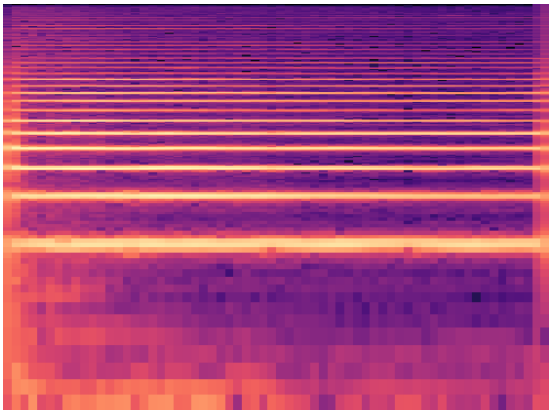
Figure D.5: Audio samples for the **Fuzz-Thin 5-Point Interpolation**.



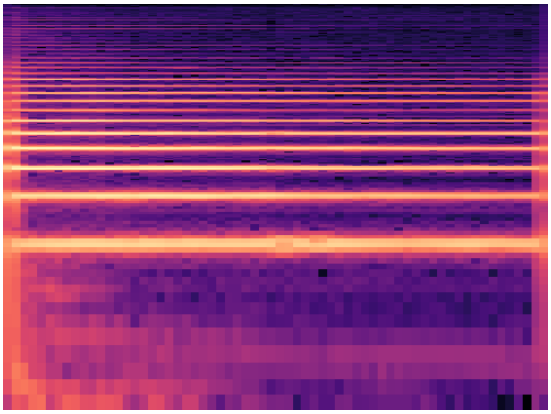
(a) Interpolation Point 1



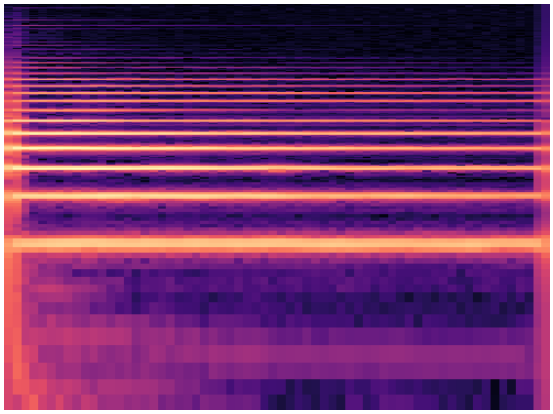
(b) Interpolation Point 2



(c) Interpolation Point 3

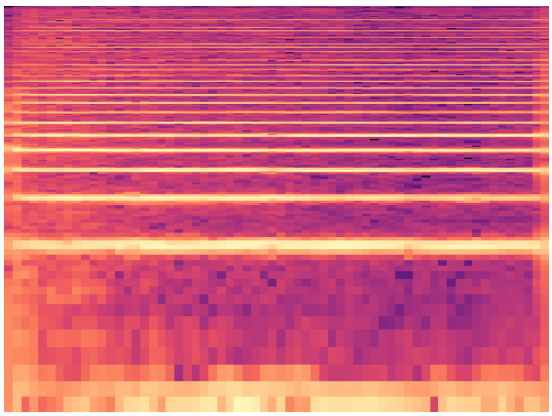


(d) Interpolation Point 4

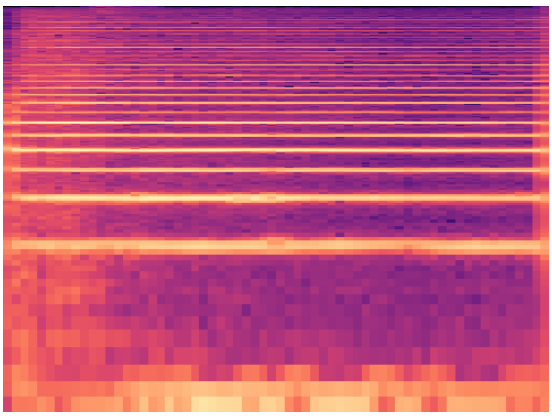


(e) Interpolation Point 5

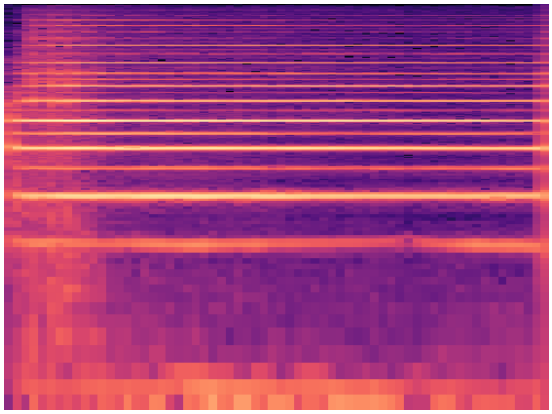
Figure D.6: Audio samples for the **Fuzz-Soft 5-Point Interpolation**.



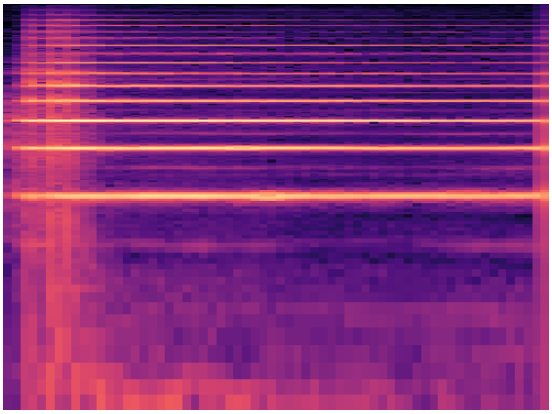
(a) Interpolation Point 1



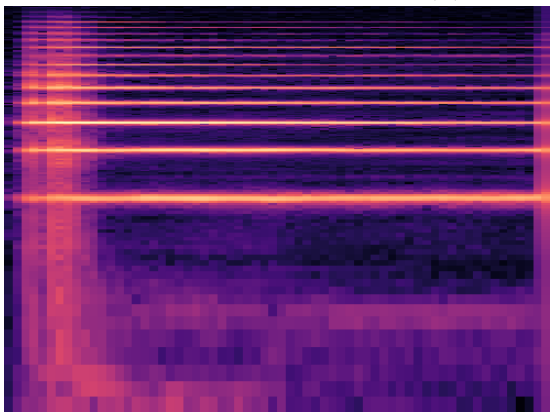
(b) Interpolation Point 2



(c) Interpolation Point 3

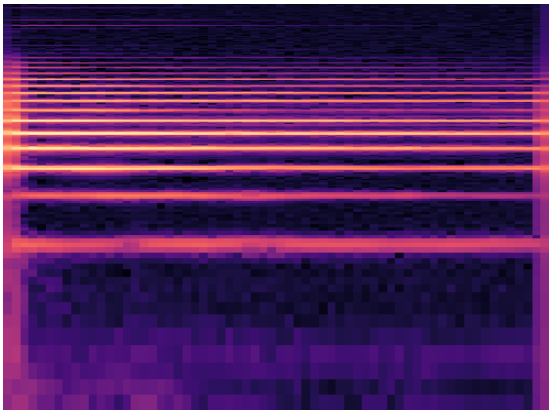


(d) Interpolation Point 4

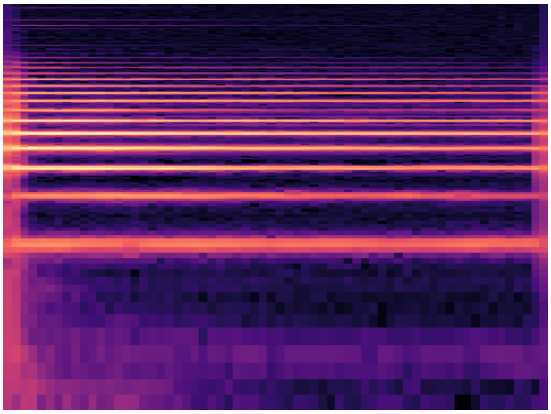


(e) Interpolation Point 5

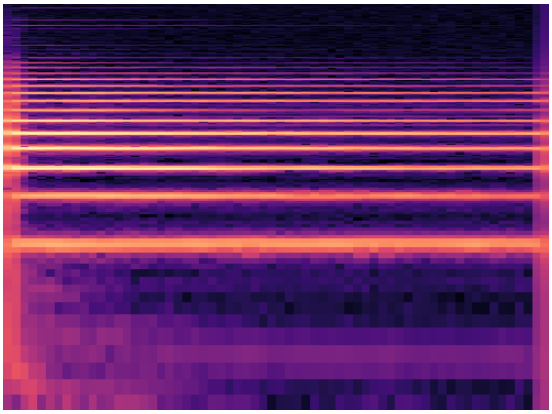
Figure D.7: Audio samples for the **Fuzz-Shimmer 5-Point Interpolation**.



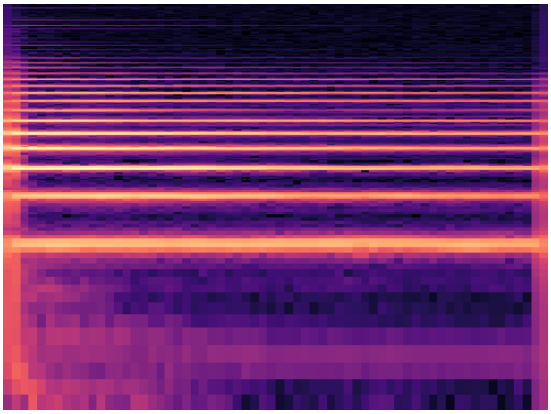
(a) Interpolation Point 1



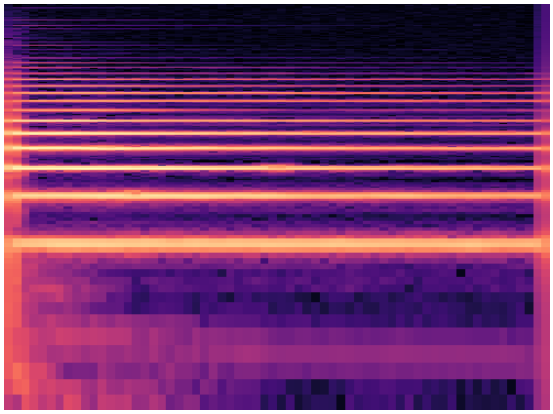
(b) Interpolation Point 2



(c) Interpolation Point 3

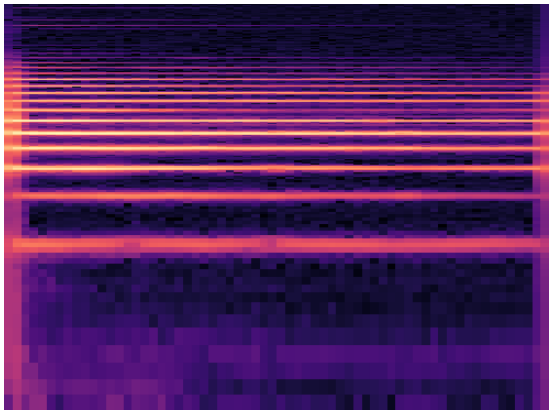


(d) Interpolation Point 4

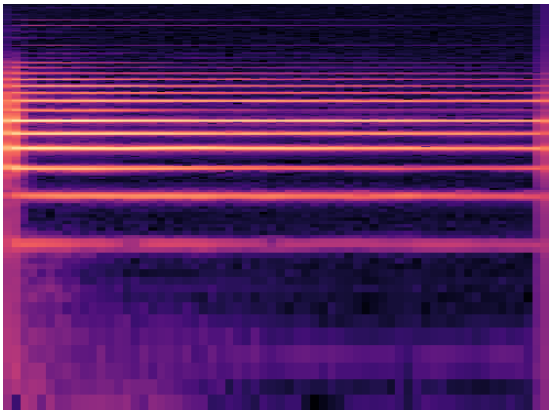


(e) Interpolation Point 5

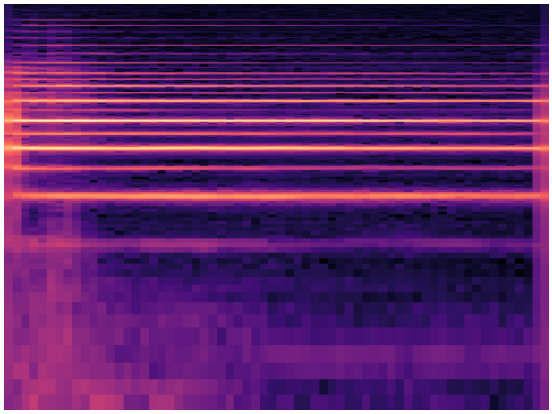
Figure D.8: Audio samples for the **Thin-Soft 5-Point Interpolation**.



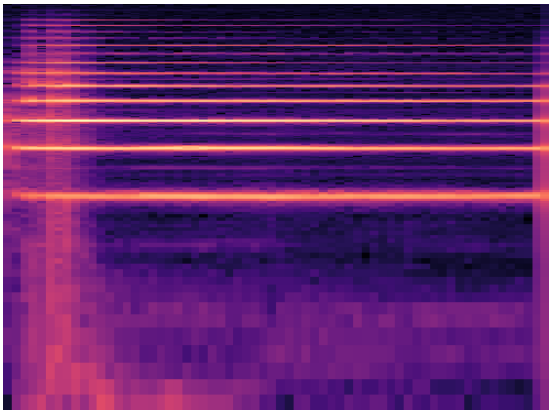
(a) Interpolation Point 1



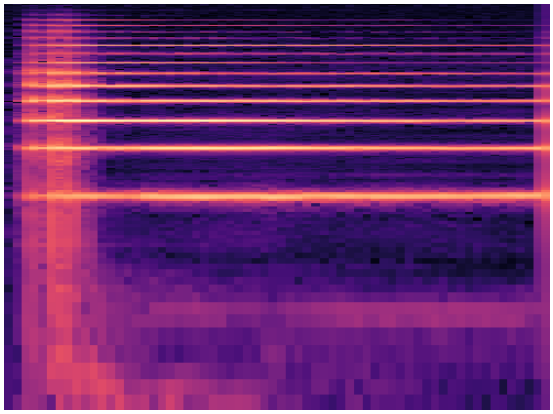
(b) Interpolation Point 2



(c) Interpolation Point 3

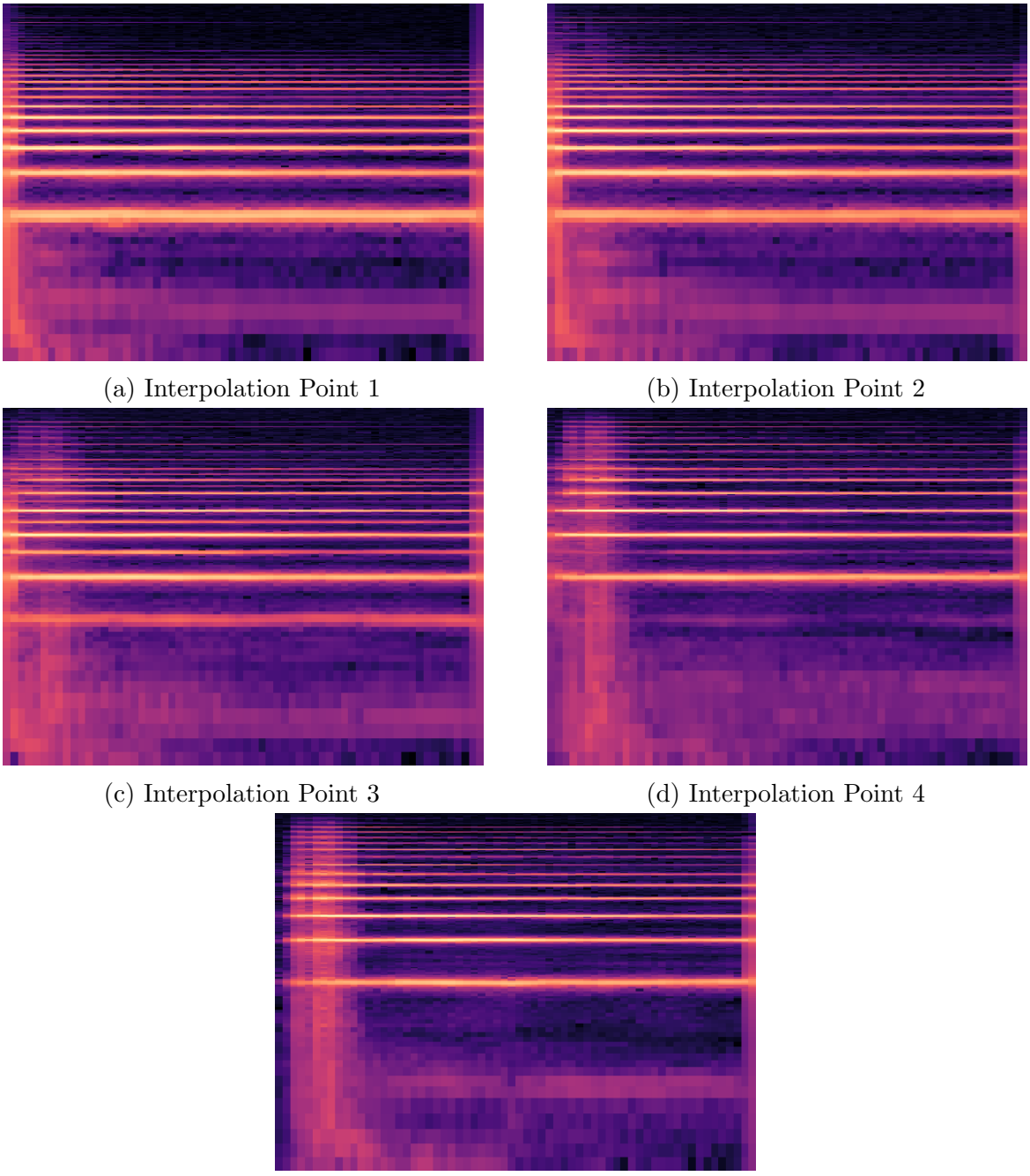


(d) Interpolation Point 4



(e) Interpolation Point 5

Figure D.9: Audio samples for the **Thin-Shimmer 5-Point Interpolation**.



(a) Interpolation Point 1

(b) Interpolation Point 2

(c) Interpolation Point 3

(d) Interpolation Point 4

(e) Interpolation Point 5

Figure D.10: Audio samples for the **Soft-Shimmer 5-Point Interpolation**.

Appendix E

Timbre classifier model architecture & training metrics

Figures E.1 & E.2 show the CNN model architecture for the timbre classifier used in the timbre interpolation objective evaluation via classification experiment detailed in Sections 6.3.2 & 6.3.3, and Figure E.3 shows its training metrics.

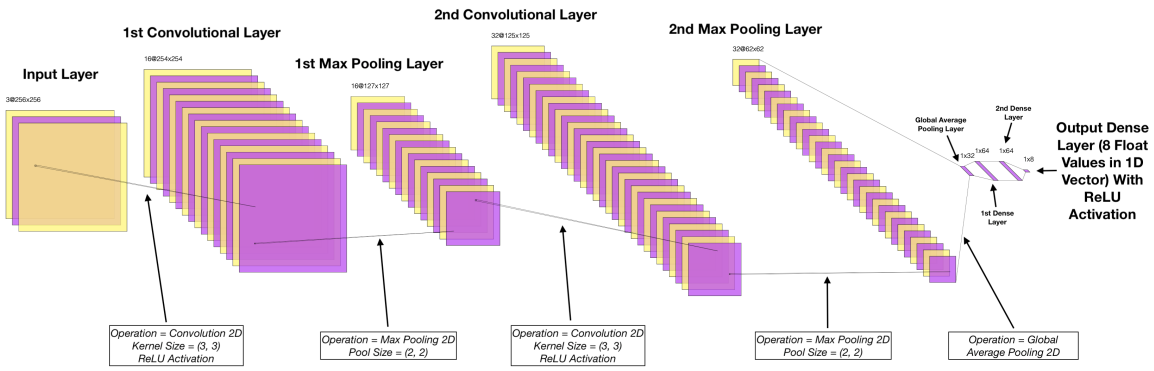


Figure E.1: Timbre classifier CNN model architecture.

```
Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 256, 256, 3)]     0

 conv2d (Conv2D)             (None, 254, 254, 16)      448

 max_pooling2d (MaxPooling2  (None, 127, 127, 16)      0
 D)

 conv2d_1 (Conv2D)           (None, 125, 125, 32)      4640

 max_pooling2d_1 (MaxPoolin  (None, 62, 62, 32)        0
 g2D)

 global_average_pooling2d (  (None, 32)                0
 GlobalAveragePooling2D)

 dense (Dense)               (None, 64)                2112

 dense_1 (Dense)             (None, 64)                4160

 dense_2 (Dense)             (None, 8)                 520

=================================================================
Total params: 11880 (46.41 KB)
Trainable params: 11880 (46.41 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure E.2: Timbre classifier CNN model architecture provided by Keras' `model.summary()` [95].

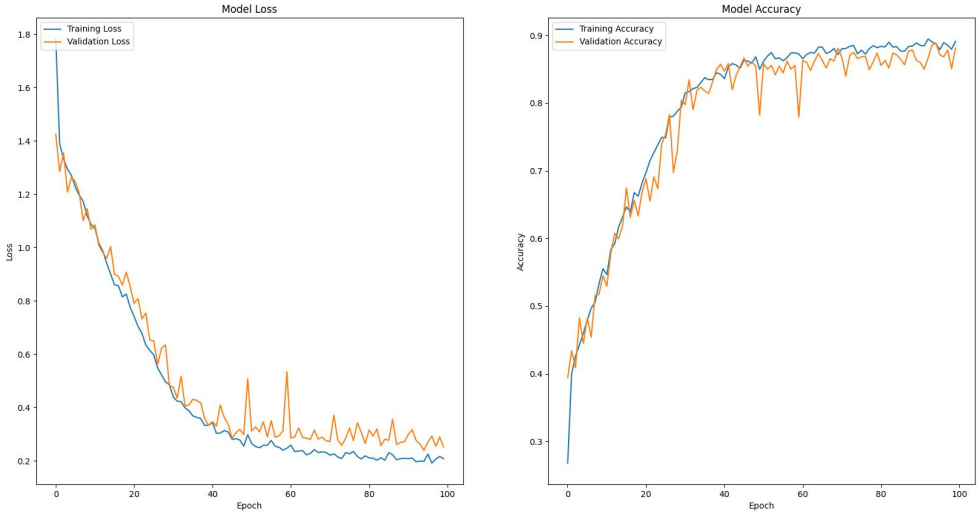


Figure E.3: Timbre classifier training metrics (training/validation loss per training epoch and training/validation accuracy per training epoch).