EVALUATING PERCEPTUAL DIFFERENCES OF TIMBRE TRANSFER BETWEEN THREE GENERATIVE NEURAL NETWORK MODELS USING AUTOENCODERS

Mikkel Sang Mee Baunsgaard

Department of Architecture, Design and Media Technology
Aalborg University
Aalborg, Denmark
mbauns18@student.aau.dk

Victor Blicher Buch

Department of Architecture, Design and Media Technology
Aalborg University
Aalborg, Denmark
vbuch18@student.aau.dk

David Mockovsky

Department of Architecture, Design and Media Technology
Aalborg University
Aalborg, Denmark
dmocko18@student.aau.dk

Oscar Bill Zhou

Department of Architecture, Design and Media Technology
Aalborg University
Aalborg, Denmark
ozhou18@student.aau.dk

ABSTRACT

Within the music industry it is very common to use either pedals or Virtual Studio Technology plugins to apply audio effects, however, it can be time consuming depending on the effect. This could be solved using machine learning, where existing research points to using Deep Neural Networks (DNN). In this paper we explored the field of DNNs for audio effects, as a means to apply timbre transfer, by creating our own variational autoencoders (VAE) and autoencoder. As a proof of concept, we specifically focused on equalization. The subjective quality was tested using MUSHRA and an A/B test, whose results showed that the generated tracks performed poorly, which indicates the models were sub-optimal. For further improvements more in-depth research is advised, while we also provide suggestions if one chooses to use VAEs for timbre transfer.

1. INTRODUCTION

Machine learning (ML) has become prominent in many fields, where the boundaries of what can be achieved with it are constantly being tested, and the music industry is no exception. A common practice within this industry is applying effects using either analog devices, such as a pedal, or digital software such as a Virtual Studio Technology plugin (VST) used within a digital audio work station (DAW). Both allow the performer to manipulate the timbre/tone in real time, whereas the latter is also commonly used in post production. This enables the producer to apply any effect, such as *double tracking*, where multiple recordings of the same melodic phrase is layered on top of each other. However, applying this effect poses two prominent problems, being either too time consuming, or the end product sounding too artificial. A solution to this, is applying ML to the process of creating music.

Chien-Yu Lu et. al. describes decomposing music into content and style [1]. The former referring to the structure imposed by a composer, e.g., the pitch, and the latter referring to the interpretation of the composer e.g., timbre [1]. Timbre is the attribute that gives the unique characteristics to instruments, enabling humans to distinguish between two instruments playing the same note. With this in mind, they trained a deep neural network (DNN) to transfer one instruments' style, onto another e.g., generating a piano with the timbre of a guitar. This proved successful, showing that generative DNNs are a means to achieve style transfer. In this paper, we will use our own definition of said style, referring to it as *timbre transfer*.

With this paper, we aim to explore the notion of timbre transfer using DNNs to make the process of applying effects faster, solving the said problems that come with audio effects. As a proof of concept, we chose an equalizer (EQ) effect, since it does not alter the phase domain. For this, we implemented three models, two variational autoencoders (VAE) trained with 20 and 200 epochs respectively, and a linear autoencoder with 200 epochs. Each model was trained on five seconds long audio tracks represented as magnitude spectrograms and the output spectrogram was split further using Musical Source Separation (MSS) to remove the generated noise. For more information about MSS, see Worksheet [Section 1.7.5]. The goal was to compare the three models against each other and evaluate their performance with human perception as a metric of success. To measure it, the MUltiple Stimuli with Hidden Reference and Anchor (MUSHRA) and A/B tests were conducted on 26 participants.

2. RELATED WORK

DNN is a term used for describing multiple machine learning methods, either being supervised, unsupervised or learned. Matthia Patterna investigated the possibility of timbre transfer between two wind instruments using unsupervised convolutional autoencoder (CAE), using a logarithmic spectrogram as input [2]. The task was split up into two smaller parts, being reconstruction and transformation. The former refers to the model's ability to recreate the input and the latter refers to model's ability to transfer a separate instruments' timbre onto another. He investigated multiple variations of CAEs, using the Adam optimizer in each, to conclude which model would be most sufficient. The results showed that a single hidden layer CAE is the best for reconstruction, and a CAE with dilation and residual connections is the best for transformation [2].

Similarly, Giovanni Pepe et al. also investigated DNNs, using two different models and one shallow neural network, as a means to shape the timbre, using impulse responses as input, as opposed to logarithmic spectrograms. This would produce finite impulse response filter coefficients used in a digital EQ filter, which would be iteratively optimized to minimize a loss function, which would compare the euclidean distance between the desired curve and the output curve. The models of interest were a convolutional neural network (CNN), a CAE and a Multilayer Perceptron (MLP). The results showed the CNN model performed the best [3].

Lastly, Jesse Engel et al. created a DNN called *GanSynth*, which is a network based on generative adversarial networks (GAN) and has been trained on the *NSynth dataset* [4], which consists of 1000 different instruments with 3000 music notes per instrument. This was then pre-processed using a Short Time Fourier Transform (STFT) to yield a spectrogram, which was further processed into a logarithmic spectrogram, similarly to Matthia Patterna [2]. For more information about STFT, see Worksheet [Section 1.7.2]. The output is a combination of two different instruments, producing a new timbre, showing that it is possible to use ML as a tool for musicians [5].

3. METHODS

Based on Section 2, it is evident that using DNNs for timbre transfer is a possibility, however, the majority used either CNN's or GANs. For this reason we sought out to test, whether it is possible to achieve the same or better results using a different DNN model, namely VAE. Thus we created three models, two VAEs and one linear autoencoder.

3.1. Models

The three models were created with *Python* [6], using the *TensorFlow* library for machine learning [7]. The two VAEs share the same architecture, which can be seen in Figure 1, using 512 latent dimensions, with a 4D tensor $(None, 513, 216, 1)$ as an input, but differ in amount of epochs, using 20 and 200. The linear model consists of two layers, making it a shallow neural network (SNN), which sizes are 1024 and 110.808, and takes a flattened version of said 4D tensor as a 2D tensor input $(None, 513 * 216 * 1 = 110.808)$.

The pipeline for the models can be seen in Figure 2, showing the three stages in the workflow which are essential for each model to work.

The pre-processing stage ensures that all data is identical and can be divided into multiple steps, which can be seen in Figure 2 in Pre-processing. First, all the training audio is sliced into an identical number of samples, creating 992 five second audio files. The audio files were then processed with the Efektor GQ3607 Graphic Equalizer [8] VST, the parameters can be seen in Worksheet [Section 2.1]. These are then passed into a STFT, which yields a power spectrogram. From this, the magnitude and phase are split into their own separate spectrograms. The former is used as an input to the model, as seen in Figure 2 in Processing, and the latter is used in post-processing.

Once the model is finished, a power spectrogram is created as a product of the magnitude and phase, as seen in Figure 2 in processing.

This is then decomposed using MSS, which separates the harmonic and percussive components into their own respective spectrogram, as mentioned in Section 2, where the former is recovered into an audio signal using inverse STFT (ISTFT), and the latter is unused. This is then the final output of the model.

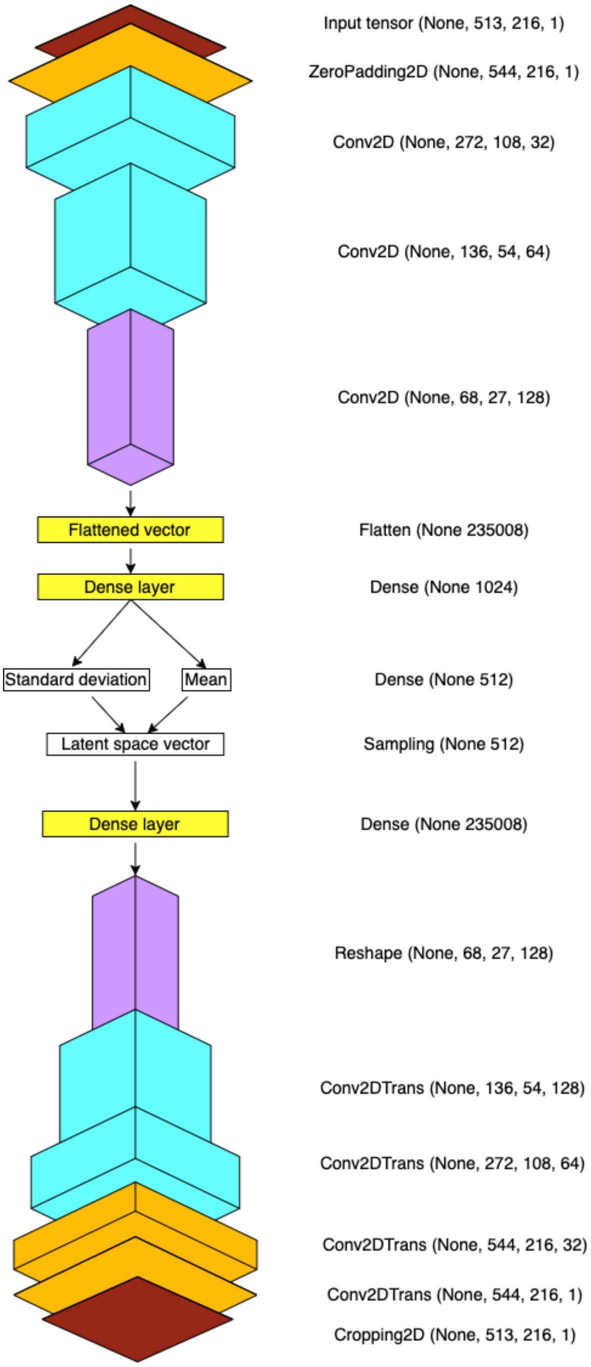
Input tensor (None, 513, 216, 1)
ZeroPadding2D (None, 544, 216, 1)
Conv2D (None, 272, 108, 32)
Conv2D (None, 136, 54, 64)
Conv2D (None, 68, 27, 128)
Flattened vector
Flatten (None 235008)
Dense layer
Dense (None 1024)
Standard deviation
Mean
Dense (None 512)
Latent space vector
Sampling (None 512)
Dense layer
Dense (None 235008)
Reshape (None, 68, 27, 128)
Conv2DTrans (None, 136, 54, 128)
Conv2DTrans (None, 272, 108, 64)
Conv2DTrans (None, 544, 216, 32)
Conv2DTrans (None, 544, 216, 1)
Cropping2D (None, 513, 216, 1)

Figure 1: *The architecture used for the VAE20 and VAE200.*

3.2. Experimental setup

The output-audio of the three models' perceptual quality was evaluated using the MUSHRA listening test in conjunction with the A/B test, which are explained in more detail in Worksheet [Section 3.1]. The test was conducted in a "quiet room" using a Lenovo IdeaPad L340-15IRH laptop running the webMUSHRA [9] and Sony WH-1000MX3 headphones. Two test conductors were present in the room during the test, to troubleshoot any possible techni-

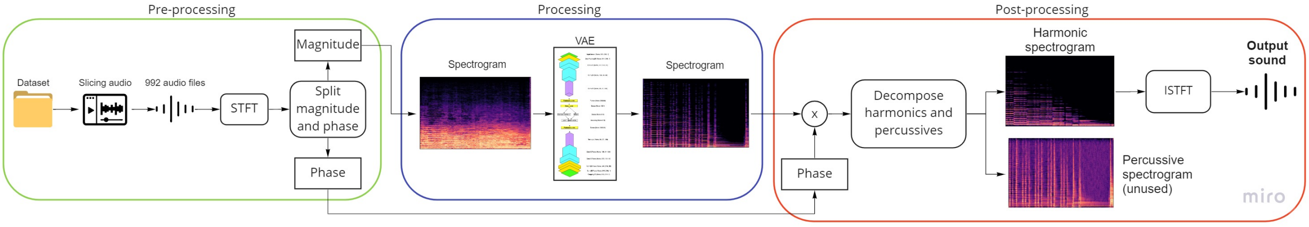

Figure 2: *A system overview showing the three distinct stages of the pipeline.*

cal difficulties, and note any other anomalies, as suggested by the MUSHRA specifications [10].

3.3. Procedure

The participants were inquired to fill out a General Data Protection Regulation consent form. Once signed, the test conductor would explain the outline of the experiment and start the test. Both the MUSHRA and the A/B test had instructions within the webMUSHRA, and if the participant had any questions regarding the test, they could ask the test conductor for clarification.

3.4. Participants

The test was carried out on 26 participants (7 female, 18 male and 1 other) from Aalborg University, with an average age of 23.56 ± 0.1 years. Based on the MUSHRA requirements [9], see Worksheet [Section 3.1.1], the participants were assessed based on their ratings of the mid-range anchor, which resulted in none of the participants being excluded. However, due to technical issues, one of the MUSHRA results was lost, therefore, the MUSHRA results are presented for 25 participants, while the A/B test has all 26.

4. RESULTS

The MUSHRA was tested for normality using the Shapiro-Wilk normality test, which proved the data was not normally distributed ($W = 0.83163$, p-value $< 2.2e - 16$), therefore, a non-parametric significance test was chosen, namely Kruskal-Wallis. The test yielded significant differences between the groups (Chi square = 1131.7, p-value $< 2.2e - 16$, df = 5), which was further investigated using the Pairwise Wilcoxon rank-sum test, whose results can be seen in Table 1.

	anchor35	anchor70	linear	reference	vae20
anchor70	0.0670	-	-	-	-
linear	<2e-16	<2e-16	-	-	-
reference	0.0995	0.7896	<2e-16	-	-
vae20	<2e-16	<2e-16	5.3e-16	<2e-16	-
vae200	<2e-16	<2e-16	0.0043	<2e-16	3.2e-08

Table 1: *Table of significance between models from the MUSHRA.*

The findings from MUSHRA can be seen in Figure 3, showing the interquartile ranges (IQR) of each model's rating. For additional information see Table 2.

The A/B was tested for normality also using the Shapiro-Wilk normality test, which also proved to be non-parametric (W=0.80846, p-value<$2.2e - 16$). To test for significance, Kruskal-Wallis was used, yielding non-significant results (Chi square = 0.85022, p-value < 0.6537, df = 2).

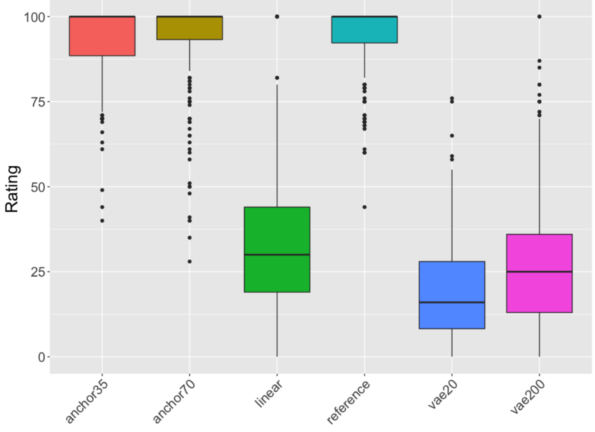

Figure 3: *Boxplot of scores per song.*

	anchor35	anchor70	reference	linear	vae20	vae200
IQR	11,5	7,75	7,25	25	19,75	23
Mean	93,14	93,52	94,74	33,88	18,95	28,5
Median	100	100	100	30	16	25

Table 2: *The IQR of each model from the MUSHRA.*

The findings from the A/B test can be seen in Figure 4, with additional information in Table 3.

	Vae20_vs_linear	Vae20_vs_vae200	Vae200_vs_linear
IQR	8.79	6.819	9.527
Mean	14.016	13.512	14.185
Median	11.594	11.720	12.204

Table 3: *The IQR between each model from the A/B Test.*

5. DISCUSSION

First of all, as the results show, there is no statistically significant difference in the scores between the hidden reference and the two anchors in the MUSHRA test, seen in Table 1. This might have been caused due to the large difference between the reference and the generated audio, resulting in the participants not noticing the smaller differences between the anchor and hidden reference.

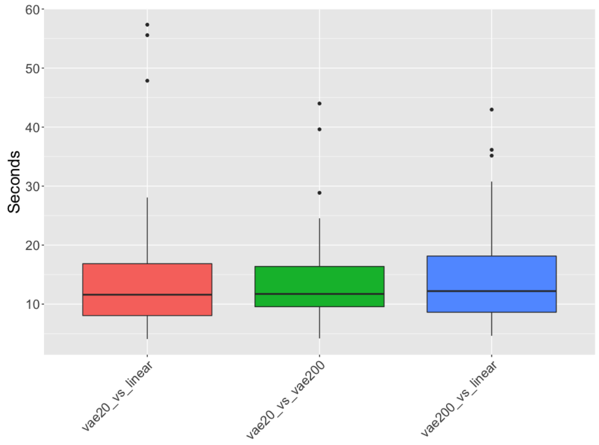

Figure 4: *A boxplot showing the time it took per participant to choose between stimuli in seconds for each comparison condition.*

The results show that the perceived quality of the generated sounds from all of the models is worse compared to the original sounds with an applied EQ. This comes as no surprise, since the generated sounds included a lot of noise, and did not nearly match the original EQ.

However, one positive result is that there is a significant difference between the generated audio from all of the models. The linear model scored higher than both of the VAEs, which might suggest that having a non-linear model trying to model a linear effect might not be the most suitable. Consequently, we investigated the VAEs further.

Firstly, we removed the model from the pipeline to validate whether it was the source of error. The pre- and post-processing stage proved to reliably reconstruct the sample, showing that the underlying issue is within the VAE.

Secondly, when comparing the two VAEs, it shows that training a model for more epochs does in fact yield better results, as one would expect. However, we believe the gap which lies in the perceived difference between the two models does not reflect on the amount of epochs per model. This leads us to believe the 200 epoch model, either is not improving much per epoch, the learning stagnates after a certain number of epochs, or the model might not be learning the correct attributes of the training set. This is also the case in the A/B test, though insignificant, it showed a tendency towards participants differentiating between the two models with an accuracy of 96%. However, taking into consideration the vast difference between the number of epochs, the difference of the perceived quality might seem unmatched.

Thirdly, the models' architecture could have been improved if we took into consideration the work from external sources, such as Matthia Patterna [2]. Specifically, using a single hidden layer CAE for reconstruction and a CAE with dilation and residual connections for transformation to create a hybrid VAE.

Another possibility is changing the model completely into a GAN instead. The paper from Jesse Engel et al. proved that using GANs for timbre transfer is valid, as it proved to successfully fuse two instruments' timbre together [4]. It appears that VAEs have a tendency to produce relatively blurred output compared to GANs, whereas GANs have a tendency to produce sharper but less accurate depictions of the samples from the original dataset e.g., some outputs appear to be fusions of entirely different samples. This would in theory make the output spectrograms less noisy, which could result in a circumvention of MSS.

Lastly, another important factor was the training-dataset. It originally comprised of 180 samples of different lengths (14s-45s), which were then split into five second-long segments, resulting in 992 samples. Though the quantity and length of the samples was in theory sufficient, the quality of each was not. Each sample varied in either tone, tempo or complexity, which could have resulted in the model not properly learning the applied EQ. This could in turn be the consequence of mainly percussive noise, as the tracks would have different general spikes in the amplitude envelope. Meaning the model could have interpreted this as percussive noise, as there would be no consistency in the dataset in regards to the amplitude envelope. This might suggest that the model could perform better if trained on simpler audio samples such as singular notes without overtones, instead of complex melodies with multiple overlapping frequencies of varying strength. Alternatively, we could also change the data type completely, using impulse responses instead of spectrograms, similarly to Giovanni Pepe et al. [3]. However, this would mean a restructure of the model's architecture, regardless of the model used. Though we are unsure how much the dataset contributed to the results, it is still important to consider for future iterations.

6. CONCLUSION

In this paper we have explored the notion of using VAEs and a linear model to apply an EQ effect as a means for timbre transfer, extending upon the research within the field of DNNs for audio, mentioned in Section 2.

Two VAEs, with 20 and 200 epochs, and one linear model were tested on 26 participants, using the MUSHRA and A/B test. The results showed that the generated audio tracks scored substantially worse than the respective anchors.

The models performed sub-optimally, with the linear model scoring the best of the three. From a further investigation it was discovered that the models were the major problem within the pipeline. This pointed to a multitude of possible solutions discussed in Section 5, being:

- Changing the dataset into single notes, or using impulse responses instead of spectrograms to produce filter coefficients.
- Implementing suggestions by Matthia Patterna, creating a hybrid between a VAE and CNN.
- Change the model into a GAN.

This could be implemented for future iterations since the models, as of now, are not suited for a real-world scenario. However, with this paper we have outlined the possible shortcomings of using a VAE for timbre transfer.

7. REFERENCES

[1] Chien-Yu Lu, Min-Xin Xue, Chia-Che Chang, Che-Rung Lee, and Li Su, "Play as you like: Timbre-enhanced multi-modal music style transfer," *arXiv preprint arXiv:1811.12214v1*, vol. abs/1811.12214, 2018.

[2] Mattia Paterna, "Timbre modification using deep learning," 2017.

[3] Giovanni Pepe, Leonardo Gabrielli, Stefano Squartini, and Luca Cattani, "Designing audio equalization filters by deep neural networks," *Deep Learning for Applications in Acoustics: Modeling, Synthesis, and Listening*, 2020.

[4] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Douglas Eck, Karen Simonyan, and Mohammad Norouzi, "Neural audio synthesis of musical notes with wavenet autoencoders," 2017.

[5] Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts, "Gansynth: Adversarial neural audio synthesis," 2019.

[6] Guido Van Rossum and Fred L Drake Jr, *Python reference manual*, Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[7] 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, Software available from tensorflow.org.

[8] Kuassa Teknika, "Efektor gq3607 graphic equalizer - guitar effects software: Kuassa," Aug 2021.

[9] M. et al. Schoeffler, "webMUSHRA — A Comprehensive Framework for Web-based Listening Tests. Journal of Open Research Software," accessed November 20, 2021, Available at *https://github.com/audiolabs/webMUSHRA/*.

[10] B Series, "Method for the subjective assessment of intermediate quality level of audio systems," *International Telecommunication Union Radiocommunication Assembly*, 2014.

[11] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu, "Wavenet: A generative model for raw audio," *arXiv preprint arXiv:1609.03499*, 2016.

[12] Shulei Ji, Jing Luo, and Xinyu Yang, "A comprehensive survey on deep music generation: Multi-level representations, algorithms, evaluations, and future directions," *arXiv preprint arXiv:2011.06801*, 2020.

[13] Executable Books Community, "Jupyter book," Feb. 2020.

[14] Simon Haykin, *Adaptive Filter Theory*, Prentice Hall, Englewood Cliffs, NJ, USA, second edition, 1991.

[15] McGarry G., Chamberlain A., Crabtree A., and Greenhalgh C., "Placing ai in the creative industries: The case for intelligent music production," *Communications in Computer and Information Science, vol 1419. Springer, Cham.*, 2021.