Interpolation Based Neural Audio Synthesis using Convolutional Autoencoders

Benedikt Langer, BSc

MASTERARBEIT

eingereicht am

Fachhochschul-Masterstudiengang

Mobile Computing

in Hagenberg

im Juni 2023

Advisor:

FH-Prof. DI Stephan Selinger
Alexander Palmanshofer, BSc MSc

© Copyright 2023 Benedikt Langer, BSc

This work is published under the conditions of the Creative Commons License *Attribution-NonCommercial-NoDerivatives 4.0 International* (CC BY-NC-ND 4.0)—see https://creativecommons.org/licenses/by-nc-nd/4.0/.

Declaration

I hereby declare and confirm that this thesis is entirely the result of my own original work. Where other sources of information have been used, they have been indicated as such and properly acknowledged. I further declare that this or similar work has not been submitted for credit elsewhere. This printed copy is identical to the submitted electronic version.

Hagenberg, June 27, 2023

Benedikt Langer, BSc

Contents

Preface

Abstract

This should be a 1-page (maximum) summary of your work in English.

Kurzfassung

An dieser Stelle steht eine Zusammenfassung der Arbeit, Umfang max. 1 Seite. ...

Chapter 1

Introduction

Chapter 2

Related Works

Despite this technology is not that well explored and popular as in the image domain, there exist a few proposed approaches that have developed a rather good solution. Some of these approaches have proven, that with neural networks it is possible to generate synthesized audio up to a certain quality. Those approaches can get categorized into different areas, as their workflow and principle differ in certain ways. As this field is related to the technique of image style transfer, a lot of works apply those methods to audio (spectrograms) and therefore call it explicitly "Audio Style Transfer". This is also because those solutions, are specifically defining a content and a style sound to combine, but more on that in section 2.2. Those methods who don't use this principle of content and style, can get categorized to the technique of "Neural Audio Synthesis" or simply just audio synthesis (see 2.1).

2.1 Neural Audio Synthesis

Neural Audio Synthesis is the field of creating/synthesizing novel sounds with the help of neural networks. The problem is similar and related to the field of Audio Style transfer. Like mentioned before, approaches in this domain differ in certain ways in those from style transfer. As a major difference, with Neural Audio Synthesis, no content or style sound is specified, which means, that for the creation of novel sounds, two sound sources are used equally. While Audio Style Transfer gets also a lot applied on whole audio samples or musical pieces, in synthesis the focus is more on the application for single notes. With a special look onto Autoencoder-Networks, Neural Audio Synthesis also includes the tasks of learning important sound features for compression and recreation of the input data. On how different approaches are designed, which (machine learning) techniques and which results could be obtained, will be shown in the following points.

Probably one of the most prominent solution, in the field of Neural Audio Synthesis, comes from Engel et al. [4]. With their work "Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders" they have proposed a system that is capable of synthesizing audio as well as interpolating/morphing encoded audio data of two instruments to create new audio. Not only they have proposed a system, but also a public available Dataset called "NSynth" that contains a large scale of high quality musical notes. The

latter has been used for training of this specific project. In their work regarding the synthesis, Engel et. al. developed and compared two different approaches with two different kind of networks. Nevertheless they have a similar structure, as they are both designed as Autoencoders but accept different kinds of data and thus have different components. While the one kind of network operates on time domain data the other one is trained on the spectral representation of audio samples. Throughout their work the second technology using spectrograms is referenced/used as Baseline Model as they focus on the use of so called “WaveNet Autoencoders” that are trained on continuous time signal. With using the Autoencoder-Structure they make use of its ability in learning efficient encodings of the music data. These encodings are representing essential features from the original audio. To create new sounds they take the encoded data from the embeddding space of two instruments and interpolate them linearly. In addition they used the decoder part to reconstruct it back to audio data. Using this mechanisms they were able to create some new sounds which contain the characteristics of two different audio signals. Comparing the performance of the two different networks used, they found the WaveNet-style autoencoder to advantageous. This not only got proven by the error scores for reconstructing the audios or auditory quality but also through quantitative comparison with a pitch and quality classifier model. Nevertheless it also can be said that the spectral baseline model has a strong performance too.
The result regarding the WaveNet Autoencoder can be explored via their online AI-Experiment called “Sound Maker”.[1] Furthermore can be mentioned at this point, that for their purpose they also created a huge dataset of audio samples (>306 000) that is open for public use.

In further publications and approaches, *Engel* continued the research on neural audio synthesis by using other network structures for this purpose. Especially mentioning here using generative adversarial networks (GANs) but also recurrent neural networks (RNNs), two more works have been published in the sake of neural audio synthesis. [3, 7] Similar to their work concerning WaveNet-style and convolutional autoencoder, they conducted experiments in (re)synthesizing audios but also to e.g. interpolate extracted features has been done. Mentioning the results briefly, it can be said, that with those further works, the suitability of those kinds of networks for audio synthesis got proven, with also highlighting a major speedup in the computation of synthesized audio samples.

The work by *Natsious et al.* does not explicitly mention the term Neural Audio Synthesis in its title, but deals with it throughout the article. [11] In their work they do a research on the reconstruction capacity of (stacked) convolutional audio encoders in terms of log-mel-spectrograms and carry out experiments on different configurations. In their experiments they evaluate the effectiveness of autoencoders in terms of neural audio synthesis whereas also possible improvements through additional techniques are measured. As they mention that with their work an exploration on musical timbre compression is made, the synthesis gets specifically refered to timbre synthesis. As audio spectrograms exist with different scales, this approach uses in contrast to others, the log-mel scale. They prove it beneficial, as it already captures the most significant

[1]“Sound Maker” https://magenta.tensorflow.org/nsynth-instrument

properties with the effect of consuming less memory and computational power. For the training they used the NSynth-Dataset proposed by *Engel et al.* [4], whereas just a sub-sample consisting of samples of different instruments of one single pitch was considered. The model(s) that where used throughout their experiments, followed the general structure of an (stacked) convolutional autoencoder network, which consists mainly of 2D convolutional layers. For experimental reasons, additional layers and techniques such as pooling layers, fully connected layers, dropout, kernel regularization got applied (added/removed). To measure the results of their experiments they were using error metrics such as root mean squared error (RMSE), structural similarity index (SSIM) but found out that those cannot accurately say something about the quality. Because of this reason, they also introduced a precision and recall score but also combined it in a F1_score. In order to generate from the spectrograms sounds, they were reusing the preserved phase information, unless there was no modification of the embedding. In the latter case the Griffin Lim phase estimation algorithm was applied, as no phase information is present.
Regarding the results that could be obtained by running these experiments by reconstructing spectrograms (without modification in latent space), some interesting findings could be extracted. To their surprise, by reducing the size of the latent space, they found out that the smaller it is, more accurate spectrograms with a smoother distribution could be generated. Also in some cases where kernel regularization got applied, the spectrograms were more accurate, while with dropout layers no improvement could be achieved. The use of (max) pooling also resulted in a more accurate time-frequency resolution with less noise, then with just convolution layers. Finally removing the fully connected / dense layer showed, that without it, the quality was significant better, as spatial information gets better preserved.

Regarding audio synthesis, *Colonel et al.* proposed over the year a few works, where they investigated the suitability of autoencoder networks regarding this task. [1, 2, 9] Starting in 2017 they proposed an autoencoder based audio synthesis through compression and reconstruction of audio spectrograms. In contrast to the before mentioned approaches, this one uses an autoencoder based on fully connected layers without convolutions. Also a different dataset was used, as they generated it themselves using an own synthesizer. A difference to e.G. the NSynth dataset used in other approaches, is that it also contains polyphonic notes and thus more complex harmonies. During the experiment they trained different parameterized networks, where they vary the depth and width of the network and its layers as well as the activation functions and used different optimizers. As error metric in this work the mean squared error (MSE) was used. Comparing these scores regarding networks of one or two hidden layers on each side show, that using the Adam optimizer worked out best in contrast to using Momentum as optimizer. These networks, just using sigmoid activation functions worked best when less compression is applied. Having 4 hidden layers, they found out, having a mix of ReLU and sigmoid activation functions worked out best. To mention here also, by applying regularization methods such as dropout and l2 penalty, that the latter was proven better as the results where of better auditory quality. Some more interesting results that could get obtained, where that having sigmoid activations led to fuller sound then with ReLU. Furthermore by using bias terms introduced noise in the results, whereas despite of the better convergence, they chose to let them out. In the end they came to the result, that using the network with 4 hidden layers and a composition of sigmoid and ReLU worked out best also in terms of auditory quality.

Another work by *Colonel et al.* was proposed in 2018, which actually states an improvement of the method, described in their previous work from 2017. [1] Those improvements contain the use of a phase reconstruction method not used before, which allows in this method to directly activate the latent space. Furthermore to improve the models convergence, the autoencoder was designed asymmetrically, via input augmentation. This means they padded the input magnitude data with different permutations (first/second order difference or mel-frequency cepstral coefficients (MFCC)). As in the previous work only MSE was contemplated as error metric, this one made use and comparison of several cost functions. To these cost functions the mean absolute error (MAE) as well the as spectral convergence cost function (SC) with L2 penalty were considered. An advantage in using the latter they found out that via the penalization of the total spectral power, the power in the output is more accurate then with the others. In comparison to their work from 2017, they also kept to leave out additional bias terms but decided to just use ReLU-activations instead of a mixture with sigmoid.
Coming to their results, overall can be said, that improvements to their previous previous could be achieved regarding the additional methods they applied. Concerning the augmentation of the input data, a significant improvement regarding score could be reached, whereas augmenting with first order difference outperformed all other. With a look onto the generated sound, it can be observed, that by padding with the MFCCs a different sound palette is present. In further comparison to their baseline, they introduced the possibility to omit the encoder part of the network. This enables to directly activate the innermost 8 neuron layer whereas the decoder can generate novel sounds. As no phase information is present, this one gets calculated/estimated by a method called "Real-time phase gradient heap integration" in order to be able to generate a playable sound. In addition to this work, they implemented a small program including a GUI, where it is possible to directly interact and activate the innermost neurons (eight control values in latent space) to generate new sounds.

In a more recent work, *Colonel et al.* implemented and compared autoencoder networks with different topologies regarding their performances for musical timbre generation. [2] This work already utilizes findings and methodologies from previous works (that have been mentioned before). Referencing the previous work from 2018, they implemented a mechanism to directly activate and control the latent space of a trained autoencoder with a graphical tool, to synthesize sounds. They found out that this technique is proven to be difficult in terms of controlling the latent space. To overcome this issue and improve the work, they added in this approach chroma-based input augmentation to improve the reconstruction performance. Besides of this input augmentation with chroma-values, they also implemented a so called skip connection, where the latent space gets conditioned with the chroma-value. In this work the chroma-values get represented via a one-hot encoded representation for each training sample, whereas the maximum value is set to one while all others to zero. The chroma-values are based on the 12 note (western) scale to represent the dominant note present in an audio sample.

For this work these one-hot encoded chroma representations tell e.g. the note played in a single-note audio. With this technique they can shape the timbre around a specific note class. For the networks topologies, they decided, to vary the size of the "bottleneck"-layer (8, 3 or 2 neurons) but also the activation functions, input augmentation, the use of the chroma skip connection as well as different datasets. To mention they trained and experimented with the self generated dataset from their previous works containing five octaves of notes, a one octave subset of it but also with a separate violin note dataset.

Concerning the results they found out, that with the network with the eight neuron bottleneck, the version with the chroma-based input augmentation worked out best. Thus, for the rest of the experiments *Colonel et al.* were using this technique. Concerning the two neuron bottleneck network, using sigmoid activation functions and no skip connection worked out best for the one octave dataset. Using the skip connection turned out to work best for the violin dataset (sigmoid and two neurons). Finally with three neurons also the variant with the skip connection worked out best for both datasets. By analyzing the latent spaces some interesting observations could be made, also for the sake of audio synthesis. They applied a clustering method to see the distribution of the values in the latent space concerning their note and timbre. Using sigmoid activations turned out to bound the values in the range of (0,1) as well as destributing the values in a more uniform manner. Also the skip connections lead to denser representation. By seeing this as advantage, and moving forward with just sigmoid activations, sampling of the latent space (with a mesh grid e.g. 350x350 for two neurons bottleneck) was done to generate a new timbre. In combination with setting the additional chroma conditioning vector to a given note class, the decoder generates the timbre that matches the chroma vector and thus the desired note to be present in the output sample.

A comparative work on autoencoders, in terms of music sound modeling, has been published by *Roche et al.* in 2019. [13] In this work they implemented four different kinds of autoencoder networks, that have been compared in terms of audio synthesis. Similar to the other techniques described earlier, this one also orientates itself on the principle of autoencoders, to project the input data to a low-dimensional space, from which input can be (re)synthesized. In their experiments the proposed autoencoder networks consist of (shallow) autoencoers (AEs), deep autoencoders (DAEs), recurrent autoencoders (LSTM-AEs) and variational autoencoders (VAEs) which all got compared to principal componen analysis (PCA). As sound data for training and experimenting, they used a subsample of 10,000 different random selected notes from the public availably NSynth dataset. The networks that where implemented got trained on the normalized log-magnitude spectra of those samples. Regarding the structure or the depth of the different networks they used for the DAE two and three layers on each side, for VAE just one version with two layers and one version of the LSTM-AE with one layer on each side. Concerning the size of the output from the encoder (latent space), they experimented with different values that reach from 4 to 100. The conducted experiments consist of an resynthesis-analysis where the reconstruction error (RMSE in dB) of the different methods got compared. Additionally to the RMSE so called PEMO-Q scores were introduced to calculate the objective measures of perceptual audio quality.
The results regarding the reconstruction error, showed to their surprise, that PCA outperformed the shallow autoencoder network. Continuing with DAEs, the reconstruction

performed almost 20% better than the shallow AE having an encoding size of 12 and 16. Also the error decreases faster when decreasing the dimension of the latent space. Even better results with over 23% improvement to PCA could be achieved by using LSTM-AEs which brought them to the concolusion, that its beneficial to use more complex architectures. Beneficial not at least as more compression and thus a small latent space can be generated which is important for sound-synthesis. Taking the results of the VAE into comparison, the reconstuction error lies between the one of the DAE and shallow AE/PCA. As the size of the latent space influences the reconstruction error, it can be said, that the bigger the size, the lower the error, with PCA outperforming all models (having an encoding size of 100). In addition to the RMSE score, the perceptual audio quality got measured with te PEMO-Q score. The results here are comparable to those with RMSE, with just the LSTM-AE having a slightly lower score (compared to RMSE). As in this work it was investigated how the latent space values can be used to be controlled by musicians, the correlation between those values has been calculated. Averaged over all samples per model, it showed that the values from LSTM have the most correlation while VAE has the least. Having less correlation makes VAE the better candidate in terms of using the latent values as control values for synthesis (less redundancy and clear perceptual meaning). In terms of audio synthesis, having the latent space variables, they also showed how to use it for sound interpolation like in the Work of *Engel et al.*. For this task they selected the latent space vectors of two sounds with different characteristics, to linear interpolate each value. By decoding and in addition applying the inverse STFT and Griffin Lim, new interesting sounds could be generated.

2.2 Audio Style Transfer

The works in this section have in common that they all entitle their work, as "Audio Style Transfer". In their methodology they all orientate themselves at technique of image style transfer. Applying the method of image style transfer to audio also means, as audio is a time-continuous signal, that it has to be brought into a similar shape, which will be done mostly by generating spectrograms out of signals. As for image style transfer, a content and a style picture is needed, this principle also gets applied to audio style transfer. In image style transfer, the style (e.g. brush strokes, colors) and content of an other image (e.g. contours, scenery) get combined, to form a new stylised image. [5] This means that in the output image, the content image looks like painted with a certain "style". Mapping this principle to the audio domain, this means, that there has to be a specific content sound (sample) that gets stylized with a certain style of a sound (e.g. style of a specific instrument). As in the image domain distinguishing content from style is already difficult, it is also a big or even bigger question that appears in the different approaches. Most of the time when defining the style of a certain audio, the authors define it as a musical instruments' timbre or even a musical genre. Alongside this a content might get defined as global music structure containing rhythmic constraints. [6] Those questions also might be influenced if whole audio samples/musical pieces might be taken to get stylised or just some single notes from an instruments. Furthermore if as audio data, speech is considered, style and content also is different defined. Here style could be e.g. the emotion of the voice or the speakers identity and content the spoken words in an sample. The following works show different solutions specific to the problem of Audio Style transfer in which they also get compared and assessed.

One approach that applies this principle, is the solution proposed by Ramani et al. in 2018. [12] In their approach they developed a Neural Network that is constructed as an (convolutional) Autoencoder Architecture (like in the work from Engel et al.). As also the title says, they speak officially about their system as "Audio Style Transfer Algorithm". The process of generating an audio containing characteristics of two audio signals is here slightly different as in the work of Engel et al. as they use in order two networks, namely a transformation network and a loss network. Both networks have the same structure and composition of layers. The loss network is trained to compress input spectrograms to lower dimensions which means that the encoder part learns to preserve the high level features of the input. In addition the decoder learns to reconstruct from the compressed data a spectrogram similar to the input of the network. For the training of the transformation network, the pre-trained weights of the loss network are used which speeds up training (just optimization towards low level features/style). Having the trained transformation network, it then is able to transform an input spectrogram into a stylised spectrogram. The loss network is subsequently used to calculate the style-loss but also content-loss between the respective spectrograms and the output from the transformation network. This loss gets minimized by back- propagation to the transformation network. By this procedure it is possible to pass a single spectrogram through the transformation network which in order outputs a new spectrogram containing the characteristics of itself (content) but also of one other style audio. To be also mentioned due to its architecture it also performs really fast and could be used for real-time use.

Verma et al. presented in their paper in 2018 a new machine learning technique for the purpose of generating novel sounds [14]. In this approach they tried to apply the method for artistic image style transfer to audio where they specifically mentioned the approach proposed by Gatys et al.[5] (see section 2.3). Unlike to Gatys, they adapted and trained an AlexNet architecture on the classification of audio-samples. This kind of network is a so called convolutional neural network, whereas the audio therefore gets converted into spectrograms, as those can be seen as grey-scale images. An important note here is that in this work they used the log-magnitude data of the STFT output. Also to mention, they adapted the network to use a smaller receptive size (kernel) of 3x3 instead of the larger ones in the original network, as they claim that it retains the resolution of the audio. As in the image domain the stylised output image gets initialized with random noise, they also use here an input spectrogram consisting of a gaussian noise signal. The random noise spectrogramm afterwards gets iteratively optimized by minimizing the content- but also style loss via back-propagation. In the end this process creates a spectrogram containing the content of one audio with the style of one other audio sample. They also found out that including additional loss terms for temporal and frequency energy envelopes, helped to improve the quality, as otherwise temporal dynamics would not get incorporated. For their experiments they imposed the style of a tuning fork onto a harp sound and also transferred the style of a violin sound onto a sample of a singing voice. In this way they developed a novel method for achieving cross-synthesis by using image style transfer methods.

More work in that field is coming from *Liu et al.* [10] which also explored the application of technologies given from the image domain for “mixing audio”. This also means, that this approches focuses on using audio as spectrograms. As the previous work solely investigated on the one technique by Gatys et al. this one explores two more approaches in addition to compare the results. While one of those two additional is inspired by Johnson et al. which is a convolutional autoencoder coupled with a VGG classification network the other one uses an approach with GAN (Generative Adversarial Network). In their work they called Gatys’ approach specifically slow transfer, as the iterative computation from gaussian noise was proven really slow. In contrast to the previous work by Verma et al., they used for the “slow transfer” method an adapted VGG network (1 input channel in first layer instead of 3) which has also been used in Gatys’ image style transfer. The transfer process is also similar to the previous work, as they use a spectrogram initialized as gaussian noise to iteratively minimize the content loss (in the higher layers) and the style loss (lower layers). Using this one as baseline model, they also adapted a faster style transfer method as coupling the VGG network with a convolutional autencoder network. The purpose of this network is to take as input the content spectrogram and outputting a spectrogram containing also the style features of a style spectrogram. Comparing it to other approaches this is very similar to the one of Ramani et al. having a transformation network. The only difference is the second network as here they are using a VGG classification network and no second autoencoder. Having the output of the autoencoder network (also called generative network) this one is the initial spectrogram on which the content and style loss gets computed in the VGG network (just like previously with gaussian noise). The gradient descent then will get applied to the autoencoder network, resulting after few iterations, in a stylised spectrogram. They have proven that this approach is way faster than the one with gaussian noise. As already mentioned before, for the third experiment they adapted a cycleGAN to accept audio spectrograms instead of images. In the image domain this kind of network is able to apply style transfer to only a portion of the input images. Also when using this method, two new sounds are calculated (in both directions). They also mentioned, that this approach generates the results in a shorter amount of time. For comparison, they listen to the outcome but also apply objective mechanisms like visual assessment of spectrograms, consistency tests with classification and examination of signal clusters. With the baseline approach e.g. the harmonic is not clear and high frequencies are discarded, also the faster transfer emphasizes on lower frequencies but is missing out on beginnings of the notes. With cycleGAN also the lower frequencies get emphasized while higher ones get discarded. The listenable results of each approach are provided online.[2]

As the already mentioned approaches are working on single notes/sounds, the work of *Grinstein et al.* has been implemented for whole audio samples [6]. Within their work they were adapting several other approaches with neural networks from the image domain for his idea. Besides of neural networks, they also implemented a handcrafted sound texture model which got compared to the neural approaches. The latter one is composed of three sound processing steps, that in combination emulates the human

[2] https://www.xuehaoliu.com/audio-show

auditory system. Taking a closer look on their approach, especially with the neural networks, it can be said that it differs in several ways. On the one hand they do not use a random noise spectrogram, moreover they already use the content spectrogram which then gets stylised through their methods. Most/many approaches that deal explicitly with Audio Style Transfer, are computing the result with a combined loss (function), that incorporates a style loss but also a content loss. *Grinstein et al.* do not make use of this concept, as they already initialize the future stylised spectrogam with the content spectrogram, like mentioned previously. On this one, just the style loss gets optimized, as the content is already present. To mention here, they proved this method to have compelling results, as the global structure of the content sound also is preserved.
With the neural network-based approach they investigated the use of three different network architectures for the purpose of audio style transfer. Concerning all three network types, they minimized the style loss on the content sound/spectrogram. The style loss is equally computed as in Gatys' image style transfer approach, to a "style sound/spectrogram", at specific layers in the network that extract the style. Via back-propagation the loss gets minimized again at each layer, which results after a few iterations in a stylised content sound/spectrogram. This workflow was applied to all three different network types and compared. As the first network they used a VGG-19 network like Gatys, whereas the input spectrogram was replicated three times in order to mach the input shape (RGB-like). By averaging all three channels in the end they obtained the final stylised spectrogram. The second network they used SoundNet which is Convolutional network learned on unlabeld videos including sounds. This type of network operates on the raw waveform wheras no generation of spectrograms has to be done in advance. As final network a wide-shallow-random network was used with audio spectrograms consisting of just one-layer CNN (like in the work of Ulyanov and Lebedev [15]). As the fourth and last method they used a handcrafted sound texture model that emulates the human auditory system. Even if its no neural network, it consists of three layers doing cochlear filtering, envelop extraction with compressive non-linearity and modulation filtering.
Having the results of their experiments using those approaches, a comparison could be made. While using the VGG network no meaningful results could be obtained (extremely noisy), the SoundNet yielded more relevant results despite also containing some noise. To their surprise the shallow random network performed best together with the sound texture model. for a better understanding, they provided their results online.[3]

When writing about audio style transfer, the work of Ulyanov and Lebedev has to be mentioned, as they are often referred to be one of the first, that explored transfer algorithms for audio. [15] In fact, they took the architecture used in image style transfer by Gatys et al. and adapted it to be used for audio spectrograms. Rather then seeing the spectrogram as a picture, with the dimensions of frequency x time, they took the freuquency values as channels for the CNN. The network itself is designed as a shallow network (1-layer) using 1D-Convolutions with random weights. To obtain a final spectrogram containing content and style, an optimization is made on random noise, to minimize the loss values to a style and a content spectrogram. Instead of applying it on

[3]https://egrinstein.github.io/2017/10/25/ast.html

single notes, this approach also uses longer samples or music snippets.

2.3 Image Style Transfer

In the previous sections it has been written about neural audio synthesis as well as neural audio style transfer. A lot of these works especially those proposing solutions for neural audio style transfer, took their inspirations from the image domain. For this reason this section makes a short excursion on the works of *Gatys et al.* as well as *Johnson et al.* as it has a relevance for this topic. [5, 8] *Gatys et al.* were the first to implement a system of neural style transfer which gets applied on visual data. By using a convolutional network trained on object recognition and localization (VGG-19) in images, they were able to extract on the one hand the texture (style) but also the content of an image. They found that especially from higher layers in the network, just high-level features of the images, such as objects and their arragments in the scene, without the exact pixel information, can get reconstructed, which will be used as content representation further on. Using a special feature space for texture synthesis, the style of a content image can get extracted by using the feature responses at certain layers in the network. By combining these two principles, respective style losses and content loss can get computed which will be used for the style transfer. The generation of the resulting starts by initialization of a random noise image, on which those losses get minimized by using gradient descent.
Johnson et al. developed on the basis of the former methodology an improved image style mechanisms, that especially shows improvements regarding computational speed. For the computation of content and style losses they use a VGG network with 16 layers that is pre-trained on image classification. To this network they add a special transformation network, that is designed as autoencoder. This one takes a target image as input (which is also the content image) and (re)produces an image on which the style and content loss gets calculated in the VGG network instead of a random noise. By performing back-propagation just in the transformation network, the VGG network stays fixed, and makes the transformation network to produce a stylised image (after training). By comparing this to the method by *Gatys et al.*, this shows a significant improvement regarding computational speed.
Having these methods in the image domain, they inspired significantly the development of audio style transfer algorithms, which are presented in section 2.2. Mentioning the works by [12] and [10] they adapted and applied the method of *Johnson et al.* while all other in section 2.2 mainly used the methodology proposed by *Gatys et al.*.

Chapter 3

Approach

Chapter 4

Experiment

Chapter 5

Results

Chapter 6

Discussion/Evaluation

Chapter 7

Conclusion

Chapter 8

Future Work

Appendix A

Technical Details

Appendix B

Supplementary Materials

List of supplementary data submitted to the degree-granting institution for archival storage (in ZIP format).

B.1 PDF Files

Path: /

thesis.pdf	Master/Bachelor thesis (complete document)

B.2 Media Files

Path: /media

*.ai, *.pdf	Adobe Illustrator files
*.jpg, *.png	raster images
*.mp3	audio files
*.mp4	video files

B.3 Online Sources (PDF Captures)

Path: /online-sources

Reliquienschrein-Wikipedia.pdf	**WikiReliquienschrein2022**

Appendix C

Questionnaire

Appendix D

LaTeX Source Code

References

Literature

[1] Joseph Colonel, Christopher Curro, and Sam Keene. *Autoencoding Neural Networks as Musical Audio Synthesizers.* 2018. eprint: 2004.13172 (eess.AS) (cit. on pp. 4, 5).

[2] Joseph T Colonel and Sam Keene. "Conditioning Autoencoder Latent Spaces for Real-Time Timbre Interpolation and Synthesis". In: *2020 International Joint Conference on Neural Networks (IJCNN).* 2020, pp. 1–7. DOI: 10.1109/IJCNN48605.2020.9207666 (cit. on pp. 4, 5).

[3] Jesse H. Engel et al. "GANSynth: Adversarial Neural Audio Synthesis". *CoRR* abs/1902.08710 (2019). arXiv: 1902.08710. URL: http://arxiv.org/abs/1902.08710 (cit. on p. 3).

[4] Jesse H. Engel et al. "Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders". *CoRR* abs/1704.01279 (2017). arXiv: 1704.01279. URL: http://arxiv.org/abs/1704.01279 (cit. on pp. 2, 4).

[5] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. "Image Style Transfer Using Convolutional Neural Networks". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2016, pp. 2414–2423. DOI: 10.1109/CVPR.2016.265 (cit. on pp. 7, 8, 11).

[6] Eric Grinstein et al. "Audio Style Transfer". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP).* 2018, pp. 586–590. DOI: 10.1109/ICASSP.2018.8461711 (cit. on pp. 7, 9).

[7] Lamtharn Hantrakul et al. "Fast and Flexible Neural Audio Synthesis." In: *ISMIR.* 2019, pp. 524–530 (cit. on p. 3).

[8] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. "Perceptual losses for real-time style transfer and super-resolution". In: *European conference on computer vision.* Springer. 2016, pp. 694–711 (cit. on p. 11).

[9] joseph colonel joseph, christopher curro christopher, and sam keene sam. "improving neural net auto encoders for music synthesis". *journal of the audio engineering society* (Oct. 2017) (cit. on p. 4).

[10] Xuehao Liu, Sarah Delany, and Susan Mckeever. "Sound Transformation: Applying Image Neural Style Transfer Networks to Audio Spectograms". In: Aug. 2019, pp. 330–341. DOI: 10.1007/978-3-030-29891-3_29 (cit. on pp. 9, 11).

[11] Anastasia Natsiou, Luca Longo, and Sean O'Leary. *An investigation of the reconstruction capacity of stacked convolutional autoencoders for log-mel-spectrograms.* 2023. DOI: 10.48550/ARXIV.2301.07665 (cit. on p. 3).

[12] Dhruv Ramani et al. “Autoencoder Based Architecture For Fast & Real Time Audio Style Transfer”. *CoRR* abs/1812.07159 (2018). arXiv: 1812.07159. URL: http://arxiv.org/abs/1812.07159 (cit. on pp. 8, 11).

[13] Fanny Roche et al. *Autoencoders for music sound modeling: a comparison of linear, shallow, deep, recurrent and variational models.* 2019. arXiv: 1806.04096 [eess.AS] (cit. on p. 6).

[14] Prateek Verma and Julius O Smith. “Neural style transfer for audio spectograms”. *arXiv preprint arXiv:1801.01589* (2018) (cit. on p. 8).

Online sources

[15] Dmitry Ulyanov and Vadim Lebedev. *Audio texture synthesis and style transfer.* 2016. URL: https://dmitryulyanov.github.io/audio-texture-synthesis-and-style-transfer (visited on 03/14/2023) (cit. on p. 10).

Check Final Print Size

— Check final print size! —

width = 100mm
height = 50mm

— Remove this page after printing! —