GAN ALBUM ART

An Exploration of the Potential of Generative Adversarial Networks as Design Tool for Decision-Making and Experiment Through Album Cover Artwork

Vera van de Seyp
Media Technology
Leiden University

GAN ALBUM ART

An Exploration of the Potential of Generative Adversarial Networks as Design Tool for Decision-Making and Experiment Through Album Cover Artwork

Vera van de Seyp

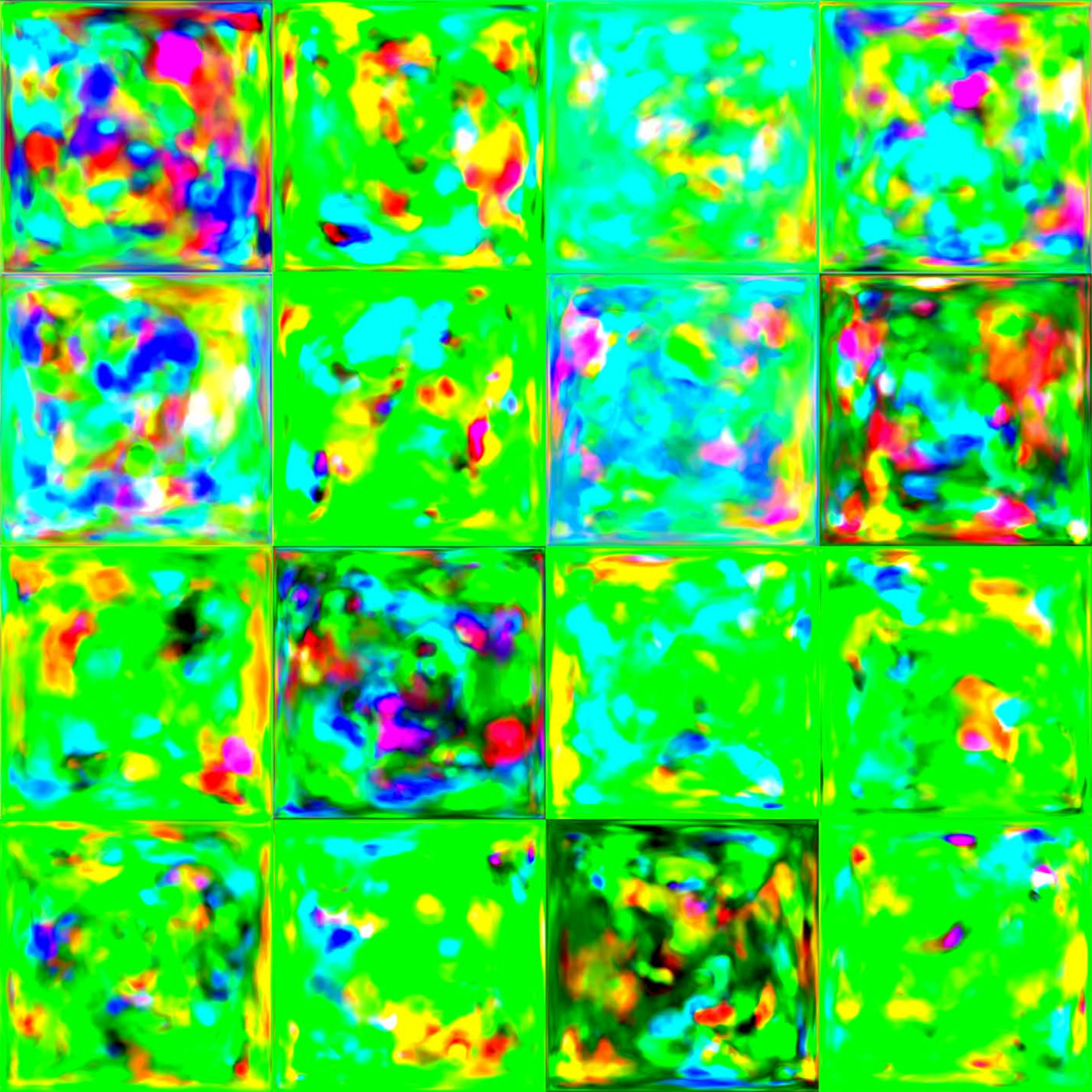
Graduation Thesis, 2020
Media Technology MSc program
Leiden University
Supervised by Tessa Verhoef & Melvin Wevers
thesis@veravandeseyp.com

ABSTRACT

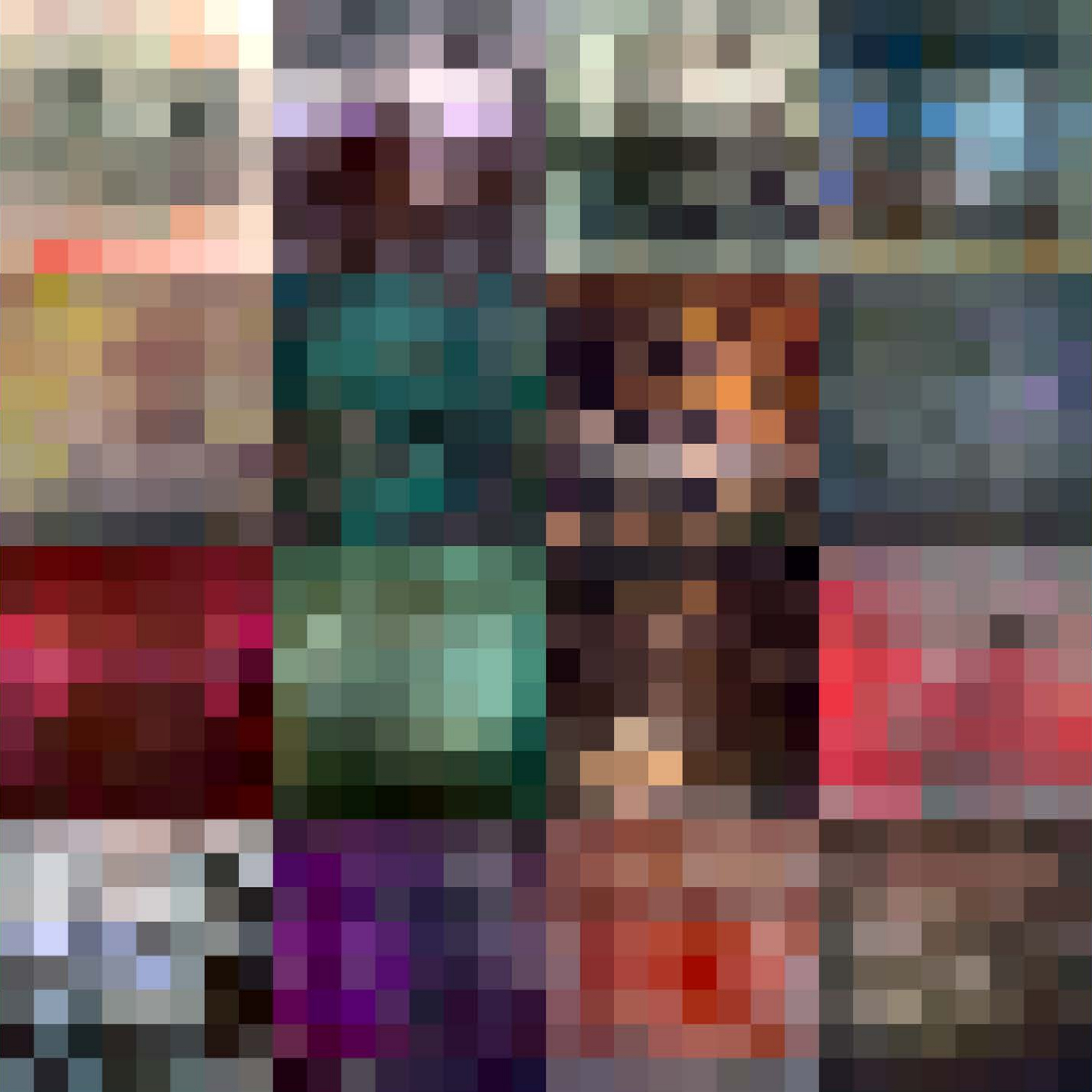
This study explores how a Style-based Generative Adversarial Network (StyleGAN) can be employed as a tool to design album cover artwork. The training dataset was created by scraping and curating around 150.000 album cover artworks from the open source music sharing community Discogs, including the accompanying metadata. These data are used to train an adaptation of NVIDIA's StyleGAN (2019) to generate high resolution images. Questions about design agency with the use of such a tool are explored and an interface to navigate the latent space of the network is introduced. Finally, a survey is done to review how the generated album cover artwork is appreciated / liked in relation to real album cover artwork: which aspects leave room for improvement, and which aspects can be useful tools for exploration and decision-making in the design field.

KEYWORDS

agency, album artwork, album covers, algorithmic design, artificial creativity, artificial intelligence, artistic research, authorship, automation, cover artwork, graphic design, design research, design tools, Discogs, experiment, experimental design, GAN, generative design, graphic design, machine learning, music, music genres, StyleGAN



(0) TABLE OF CONTENTS



(1) INTRODUCTION

To design artwork for an album cover is an applied yet creative process usually dedicated to a graphic designer (or simply designer). As design is a creative process with an output exposed to subjectivity, it is difficult to pinpoint what factors contribute to the effectiveness and consistency of the process. As the design of album cover artwork is applied, meaning it is normally designed for a specific artist or collection of music, it can at this point not be fully automated either. In contrast, computers normally follow automated processes in a rigid order to execute tasks that are assigned to them, with not much room for subjectivity or consistency. Hence, it would be impressive if a computer could somehow be taught how to generate such a subjectively reviewed, creative process, let alone album cover artwork in general.

Even though the precedent paragraph explains why it is practically impossible—at the current level of technology at the release of this paper—to generate album cover artwork from scratch, this research is attempting to discover to what extent it might be possible, using a novel technique within the field of machine learning, Generative Adversarial Network (GAN). This technique gives even less control over the output than most generative design systems presently existing. As the image quality of this technique recently became higher than ever, a curiosity arose for potential functional applications within the graphic design practice, for instance as a design tool that could make album cover artwork faster and more efficient and perhaps come up with novel things, beyond the constraints and biases of the human mind.

(1.1) RELATED WORK

Since the development of the first GAN — a concept further explained in section (2.1) — in 2014, a miscellaneous collection of experiments has stemmed from multiple versions of the network.[1] Some of those experiments involve the generation of album cover art as well. An adequate example of this at the time of writing this paper,

1 For instance, see Radford, A. et al. (2016), Bachl, M. et al. (2019) and Rani, H. et al. (2019).

is the research *Album Cover Generation from Genre Tags* (2017) by Hepburn et al. in which album covers are generated by genre using a GAN. Not only did they introduce the generation of album covers with a GAN, they also divided the covers per music genre. Hepburn et al. generated album covers using an Auxiliary Classifier Generative Adversarial Network (AC-GAN), that was pre-trained on the OMACIR[2] dataset and later trained on a Spotify dataset of 50.000 entries divided over 5 genres. An extension of the well-known structural similarity index, a multi-scale structural similarity index measurement (MS-SSIM) was used to compare structural similarities and the network was able to predict the genre with an accuracy of 35%. Differences were distinguishable between the 5 genres when using the same random and latent variables (Hepburn et al., 2017).

Lībeks and Turnbull researched the visual effect of album cover artwork from another perspective in their paper *You Can Judge an Artist by an Album Cover: Using Images for Music Annotation* (2011). Through a user study, they proved that the album cover artwork as well as promotional imagery carries information that helps place a musician into a musical genre. Their paper thus demonstrated that visual appearance differs recognizably between musical genres.

In terms of synthesis of state-of-the-art quality images with GANs, a recently published paper *A Style-Based Generator Architecture for Generative Adversarial Networks* (2019) Karras, Laine & Aila proposed a method borrowing from style transfer literature. Based on the Progressive GAN Karras et al. released in 2018, they proposed an alternative architecture that produces very realistic, high quality images by introducing progressive, scale-specific control of the image synthesis.

(1.2) RESEARCH QUESTION

The recent release of the StyleGAN by Karras et al. (2018) raises questions about possible practical implementations of such a technique. Their research allowed for a production of images with higher resolutions than any GAN had produced before, meaning that the technique has a bigger potential for functionality in visual domains. However, Karras et al. focused more on the improvement of the technique itself than discussing which potential applications would be conceivable. Album cover artwork is a resourceful object to study this potential for design because it is a popular, multi-faceted design canvas. Therefore, this research focuses on the question:

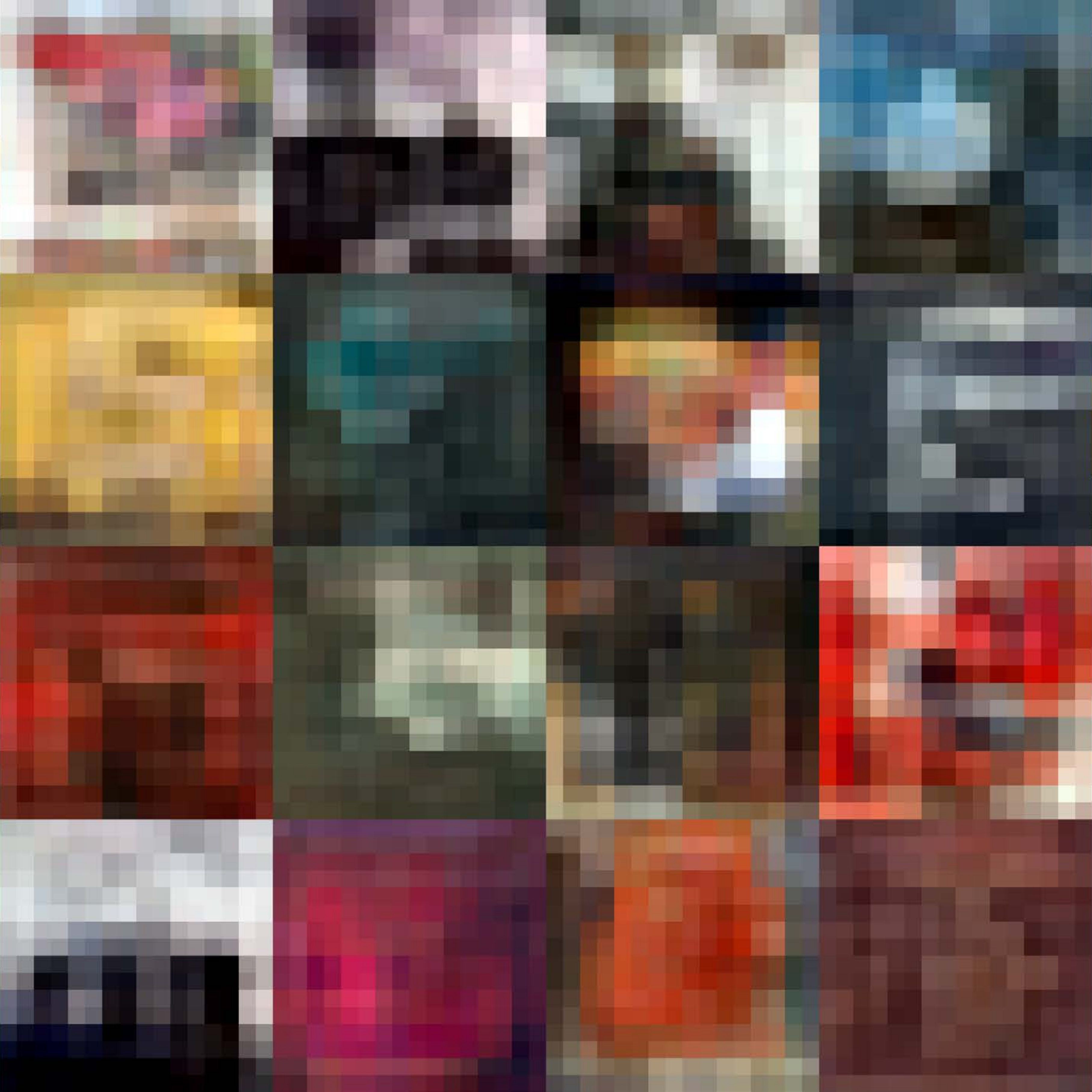
Can a StyleGAN be trained to design album cover art that looks as good as human-designed album covers?

Crucial to this research is to discover which design elements can already be outsourced to such a network, and which elements are still performed better by human designers at this point. This is not aimed at determining whether StyleGANs could entirely replace designers, but rather in which ways these two entities could reinforce each other. This collaborative aspect separates this research further from the Karras study. Designers were collaborating with machines and software far before[3] the invention of GANs, so the question is what this specific technique could offer and what a collaboration would look like. This incites the more general sub-question:

What could a human-machine design collaboration look like with a technique such as a StyleGAN?

2 The One Million Audio Cover Images for Research. Retrieved July 23, 2019 from archive.org/details/audio-covers.

3 For example, Macromedia Matisse in 1995. web.archive.org/web/20170201140928/pages.swcp.com/rtoads/vivisect/software/matisse.html.



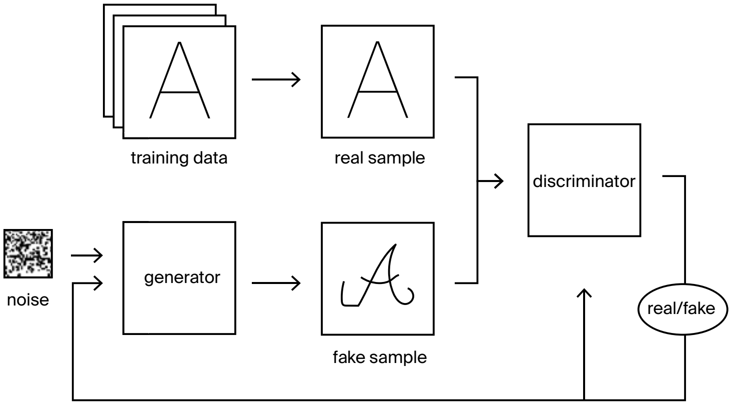
(2) BACKGROUND

Before expanding on the research, the background is discussed: the specific definition of terms used throughout the research are established and explained. Section (2.1) is focused on GANs, (2.2) on the difference between art and design, (2.3) on computational art and design, (2.4) on the notion of agency, (2.5) on the album artwork and section (2.6) on the different musical genres and subgenres.

(2.1) GENERATIVE ADVERSARIAL NETWORKS

GANs were initially introduced by Goodfellow et al. in 2014 as a novel concept within machine learning that can generate artificial samples of their given input, such as images. A GAN is composed of two adjacent neural networks, a generative model and discriminative model (Goodfellow et al., 2014). The objective of the generator is to generate data as realistic as possible in order to have a high probability rate that the discriminator cannot distinguish it from real data, initially receiving a random vector (noise) as input. The discriminator receives this generated data as input as well as real data and assesses in which category they belong, passing this feedback on to the generator.

The variables in the discriminator are gradually optimised to distinguish between real and generated input, whilst the variables in the generator are optimised to fool the discriminator into classifying the generated input as real. The generator thus learns how to create images that look as realistic as the actual images, while the discriminator learns to discriminate between generated images and images from the dataset. The two models tend to gradually enhance each other. The paper by Goodfellow et al. uses an analogy to describe the GAN process: if the generator would be counterfeiters, attempting to create realistic fake money, the discriminator would be the police, seeking to stop them. The competition between these two parties causes both to improve until the fake money would be indistinguishable from the real (Goodfellow et al., 2014).



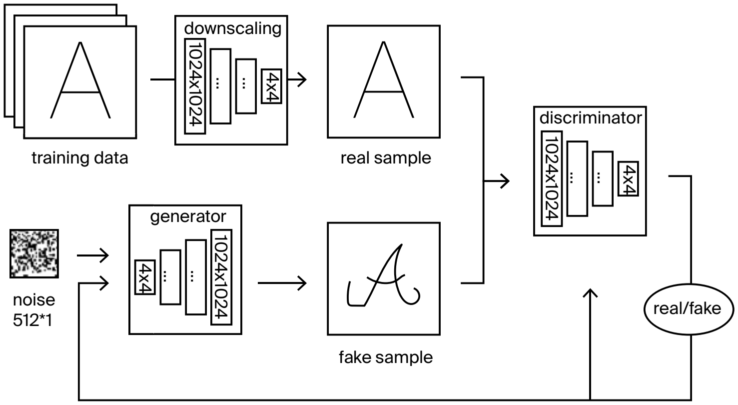
(fig. 1) An abstracted representation of the elements of a GAN. The figure is created by the author based on the paper by Goodfellow.

The adversarial framework presents some challenges, e.g. that training phases have to stay balanced as the two models have to improve at more or less the same pace—one cannot significantly outperform the other. This can be done by tweaking the parameters of the models, but the effect is unpredictable. Another challenge is generating large size, high-quality images, as images have two dimensions which is a matter of computing power. One of the positive results of GANs is the low chance of overfitting, because the generator cannot see the input data directly but only through the discriminator feedback, and is thus prevented from overfitting.

(2.1.1) PROGAN

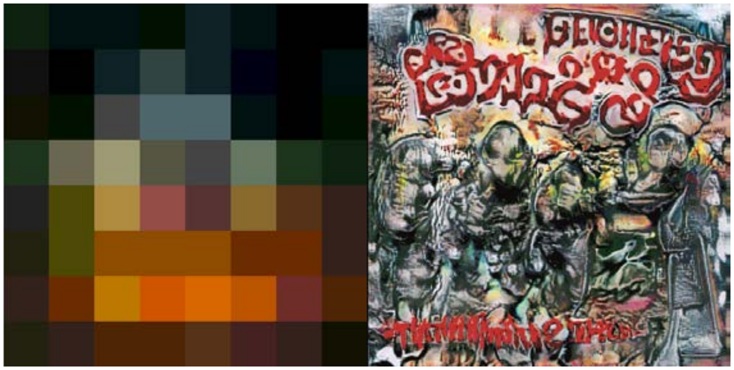
In 2018, the challenge of producing high quality image files was tackled by a new adaptation: the progressive GAN (ProGan) from the paper of Karras et al. (2018), researchers at NVIDIA. The new approach of this GAN was the progressive training of the data input, starting with a very low resolution of 4 x 4 pixels and gradually adding higher resolution layers over time.

The effectiveness of this approach lies in the creation of a foundation of an image input at the very beginning, as the core features of an image are even visible in a small resolution. As the resolution of the input images builds up, increasingly more details are learned over time.



(fig. 2) An abstracted representation of the architecture of a ProGAN.

Training on the low resolution images is faster and aids in training at higher levels of resolution as well, therefore the total training time is faster too. However, there is still a lack of control on the features.



(fig. 3) ProGAN progressive training and increasing resolution.

(2.1.2) STYLEGAN

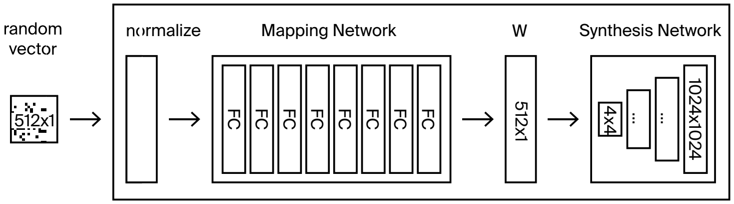
In December 2018, the website titled *This Person Does Not Exist* was launched, showing a new generated face on each page refresh.[4] This was a visual introduction to the paper of Karras et al. (2019) introducing yet again an improved version of GANs, this time an adaptation of their ProGAN. Here, they try to tackle the control of features by mapping them into the progressive size of the layers, rendering faces. The lower resolution layers have a coarser effect on the outcome than the larger layers, divided into three parts.

4 This person does not exist. Retrieved May 11, 2019, from thispersondoesnotexist.com.

(1) Coarse, images of 8 x 8 pixels or smaller: general shapes and composition, e.g. hair style and pose.

(2) Middle, images of 16 x 16 to 32 x 32 pixels: finer features, e.g. open or closed eyes.

(3) Detailed, 64 x 64 pixels and up: colour scheme and micro features, e.g. shape of nose and eyes



(fig. 4) The generator of the StyleGAN with the Mapping Network and the Synthesis Network of the ProGAN based on illustration from Karras et al. (2019).

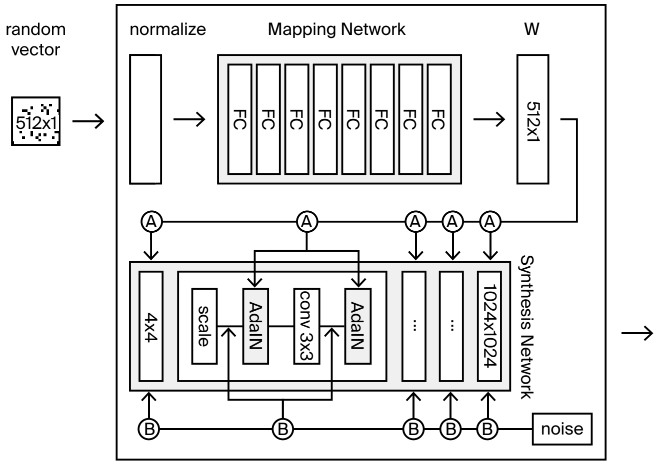
These style-based GANs (StyleGANs) differ from ProGANs in terms of generator architecture, specifically in one element: rather than repeating deconvolutions, the generator network is fully connected and two techniques are added to it. The first added part is called the Mapping Network (Karras et al., 2019) and consists of 8 fully connected layers. It maps a random code into parameters of the images of the dataset for adaptive instance normalisation layers, e.g. if there are predominantly faces with dark skin in the dataset, more input will be mapped to that trait. This results in feature entanglement: the incapability of mapping parts of the input noise to certain traits (Karras et al., 2019). To correct this, a second module was added, named Adaptive Instance Normalization (AdaIN), to transfer the output of the Mapping Network (W) for each resolution layer of the generated image. On each level of image resolution, this technique is added to determine the level of expression of features. This AdaIN process can be explained in a few steps:

(1) Each channel of the current convolution layer is normalized.

(2) The vector from the Mapping Network is transformed by another fully-connected layer (A) to a scale and bias channel.

(3) These vectors shift all normalized channels of the convolution layer, determining the importance of each filter in the convolution.

Whereas the previously described models use random noise as initial input for the generator, the StyleGAN uses a constant value as the feature improvements are controlled by this mapping network instead, rendering the noise input for the generator unnecessary and starting with a constant vector instead. However, random noise is used for stochastic variation of the images, which are small details that can make an image just different, i.e. wrinkles in the face of a person (Karras et al., 2019). To avoid this noise (B) intervening with the feature entanglement, it is added via the AdaIN technology and changes some details within its resolution layer. Primarily because of its improved architecture, which uses W and AdaIN to control the image features, the StyleGAN has a better performance than its predecessors. It thus allows for a higher quality when it comes to generating images (Rani, H., 2018).



(fig. 5) The generator of the StyleGAN with the Mapping Network, noise and the AdaIN in the Synthesis Network based on illustration from Karras et al. (2019).

(2.2) DESIGN AND ART

In the term "album cover artwork", the word "artwork" is slightly ambiguous, as it sounds similar to art-work—a work of art – but rather, it is an applied, graphic artefact that could be classified as graphic design. To understand which factors makes for successful album cover artwork, it is important to establish the difference between art and graphic design and attribute either or both categories to it.

(2.2.1) GRAPHIC DESIGN

Graphic designer Tibor Kalman proposes a generic definition of graphic design. He argues it is a medium or a means of communication using "words and images on more or less everything, more or less everywhere". With the term design, graphic design is implied: the practice that ranges from conceptual problem-solving to making compositions with text, image and graphics (Kalman, 1991).

In his renowned book *Design as Art* (1966), designer Bruno Munari describes designing as planning with aesthetic sense, giving an object "a form as appropriate as possible to its function". Munari argues that from those design decisions, certain industrial products depend heavily on designers for their commercial success. These design decisions can be for instance about choosing the most appropriate material, but can also address psychological and aesthetic factors. Important to note here is that he understands the term "aesthetic" as a formal coherence, rather than beauty in the abstract sense. An orange is the perfect example of coherence of form, function and consumption: plenty perfectly bite-sized, silky containers of sweet, juicy flesh, its seeds—the actual product to be transmitted!—delicately hidden, packed together ergonomically in a boldly coloured, protective shell, strong yet soft to the touch, that can be discarded without guilt (Munari, 1966). If it would only be the seeds, no animal would want to eat it. Thus, according to Munari's elaborate description of the orange and many more objects, the form should be coherent with the purpose, adhering to the famous adagio by Jean-Baptiste Lamarck: the form follows the function.[5] Not all designers agree whether visual decoration belongs to this function. For example, Austrian architect Adolf Loos criticised the use of ornament within a functional object (Loos, A., 1929), in contrast to Munari or Kalman who viewed the thoughtful use of decoration as an *addition* to an object's functionality.

Defining *good* design depends on one's definition of the design practice in general. For example, in his book *Design is Invisible* (1980), Swiss sociologist Lucius Burckhardt was one of the first to suggest that design extends beyond the visible. He explains that a beautiful tram is not a *good* tram when it doesn't run at night, therefore the concept of its design is extended to invisible aspects such as its timetable (Burckhardt, 1980). Additionally, designer Ellen Lupton and curator Andrea Lipps co-created a book that serves as a manifesto for a multisensory design practice. In *The Senses: Design Beyond Vision* (2018) they propose that design extends beyond vision.

Recently, the role of the design practice in the context of society has often been addressed. In *Design Futuring* (2008), philosopher Tony Fry argues that in the current day and age the design practice plays important ethical role in overcoming an unsustainable way of living. Strategic designer Dan Hill aggrandizes the concept of design even more. He asserts that traditional design principles can be used beyond the design practice, in order to provoke change on a large societal scale. He debates that effective design engages with politics and attempts to tackle systemic challenges such as the environment, healthcare and education (Hill, 2014).

However, the focus in this research lies on these basic visual principles and a *fundamental*, *visual* essence of what an album cover artwork should be as a design object. For pragmatic reasons design visions and political perspectives are set aside for a focus on the fundamental principles of graphic design.

5 Found in Munari, B., (1966).

(2.2.2) ART

Munari makes a clear distinction between art and design, distinguishing between design as *applied art* and art as *pure art*. He argues that design is a form of art as well, since in ancient times art was a trade, and artists were commissioned to make public works of communication. Nowadays, this role of creating visual public communication has shifted to the designer, not because the traditional artist is not valued, but because the designer is in control of the full process, from idea to every part of the production process, connecting art with the public as applied art. Munari argues that pure art focuses more on beauty rather than communication. In his book *The Boundaries of Art* (2001), David Novitz agrees that the difference between the terms art and design is a functional rather than a definitional one.

According to the German philosopher and social critic Theodor Adorno, there are many different views on the definition of art. In his book *Ästhetische Theorie* (1970), he pleads that there is nothing self-evident about the notion of art, as many fields of practice define it from a different, operational perspective. Furthermore, beyond fields of practice, each era has its own aesthetic paradigm that continually evolves over time. In fact, there even exists a movement of scepticism about whether art should be defined at all. For example, Ludwig Wittgenstein suggests there should not be a definition of art as it is too diverse to capture; to limit the phenomenon of art with a definition would exert a suffocating influence on artistic creativity. This view has been backed for different reasons by theories of feminism, Marxism and postcolonialism (Adajian, 2018). However, as Adorno suggests, different trains of thought regarding the definition of art and its essence continue to come in and out of fashion. In the next two paragraphs, two views on the definition of art are discussed.

On the one hand, there are definitions stemming from the practice of philosophy. According to Immanuel Kant, art belongs to the broader topic of aesthetic judgment, which, amongst others, covers the judgement of beauty and the sublime. In the *Critique of Judgment* (1790), Kant proposes that the judgement of beauty has four key elements: devoid of self-interest, universal, necessary and “purposive without purpose” (Kant, 1790). Like Kant, philosopher Edmund Burke made a distinction between the Beautiful and the Sublime in art in his publication *A Philosophical Enquiry into the Origin of Our Ideas of the Sublime and Beautiful* (1757). For Burke, the Beautiful in art is aesthetically pleasing, whereas the Sublime is more destructive, it has the power to change us forever, to destroy us. The fictional novel *The Hitchhiker’s Guide to the Galaxy* (2002) by Douglas Adams goes as far to suggest that art is the value that defines humanity.

These romantic, fundamental definitions of the perception of art are in sharp contrast with the views of conventionalists. George Dickie, an institutionalist philosopher, argues that art is an artefact of a kind created by an artist, to be presented to an artworld public (Dickie, 1984). From a more personal perspective, according to Graham McFee, an artwork can obtain an art-status by the judgement of an art expert (McFee, 2011).

(2.2.3) CONTEXTUALIZING THE ALBUM COVER ARTWORK

Whichever definition of the above would be followed, it can be established that the album cover artwork is not a piece of art within any of them. From Kant's definition, judging an album cover artwork is not devoid of self-interest, or purposive without purpose, as the underlying purpose is to find out if an album would interest the spectator. As album cover artwork can be aesthetically pleasing, it doesn't normally have the destructive power of the sublime as mentioned by Burke. It also isn't an artefact that is presented to an artworld product as Dickie describes. If McFee's definition would hold, an art-expert had to be asked to categorize each single album cover artwork. For every single artwork, this would depend on the context and time, according to Adorno's views. Additionally, *The Hitchhiker's Guide to the Galaxy* does not mention album covers when defining art as the value of humanity (Adams, D., 1979).

(2.2.4) THE ALBUM COVER ARTWORK AS DESIGN OBJECT

In the following section addresses why the album artwork can be seen as a design object according to Kalman's as well as Munari's definition and what would make a single instance *succeed very well* in terms of design, i.e. establishing it as a *good* design object.

In Kalman's definition, graphic design is a medium and a means of communication. In the case of the album cover artwork, the musician and the type of music has to be communicated. Therefore, the first factor is:

(1) The cover artwork has to communicate.

Then, following Munari's definition, another factor can be determined: the artwork has to have a form coherent to its function. Munari describes that the form of posters has to surprise you, capture your attention directly—yet without screaming at you—as the function is to get information across and invite a person to look into the product in a short amount of time (Munari, 1966). A similar statement can be made about the album cover: the visual surface should allure and charm, while maintaining enough mystery to keep an individual's attention long enough to make them listen to the music it represents. How to spark one's curiosity differs per person and depends on their interests (Hirsch et al., 2012), ergo different music genres may lead to different visual styles of attractions. Therefore, the second goal of the album cover artwork as design object would be:

(2) The cover artwork has to be appealing to its target audience and incite curiosity.

Yet Munari argues that posters selling similar products need to differ from each other. He says: "But the designer's experiments have taught us that it would be enough to employ an unusual colour, a different form, and to give the passer-by exact and immediate information instead of assaulting him time and time again until he is battered senseless." (Munari, 1966). So repetition should be avoided, diversity is a way to stand out without screaming. Therefore, a third factor can be included:

(3) The cover artwork has to be recognisable.

Many more aspects can be added to the list, and the way this list is composed also heavily depends on who is composing, as demonstrated in (2.2.1). As this research is focused on the bare essentials of graphic design, conclusively three principles are appointed to the album artwork as a design object: it has to communicate, it has to be appealing and exciting to its target audience and it has to be recognisable.

(2.3) COMPUTATIONAL DESIGN

Within the field of graphic design, there are a few design philosophies that borrow from computer science, that observe design as a program and create their designs algorithmically. Since the increasing availability of the desktop computers, graphic design has been inseparable from digital software (Girard, D., 2014). Whereas the algorithms that ran early design software such as QuarkXPress[6] were hidden behind an intuitive, WYSIWYG user interface, there were initiatives that sought to control the algorithm and therefore the design possibilities. There was a need to go beyond the user interface and create tools closer to the core of the computer.[7]

In the early 1990s, designer and technologist John Maeda taught a course called "Design By Numbers", with accompanying software program DBN, at the MIT Media Lab.[8] The objective of the course was to create software that designers or artists could use to start programming and use in their practice.

One of the results of this course was the software Processing, developed by Maeda'a students Casey Reas and Ben Fry. Presently, different versions of the software

6 For information on QuarkXPress, visit quark.com/products/quarkxpress/. Retrieved September 9, 2019.
7 See an example of a collection of tools on *Generative Design, Creative Coding on the Web*, generative-gestaltung.de/2/. Retrieved July 28, 2019
8 To see the full course description, visit dbn.media.mit.edu/. Retrieved July 28, 2019.

are used in design curricula throughout the world.[9] Whereas computational design and art existed before the appearance of these types of accessible, well documented programs, they gave way to a new wave of computational design (Madsen, R., 2016). This is further discussed in the following two paragraphs.

(2.3.1) GENERATIVE DESIGN

According to the book *Generative Design* (2012) written by Hartmut Bohnacker et al., generative design is "a revolutionary new method of creating artwork, models, and animations from sets of rules, or algorithms. By using accessible programming languages such as Processing, artists and designers are producing extravagant, crystalline structures that can form the basis of anything from patterned textiles and typography to lighting, scientific diagrams, sculptures, films, and even fantastical buildings" (Bohnacker et al., 2012). A program like Processing runs a piece of code that holds a canvas on which shapes, text and images can be drawn, that can then be exported to printing files for books, moving image, and 3d models. As the code variables can quickly be changed or randomised, a large set of different versions of the same design can quickly be generated. This approach of working has thus become a powerful programmatic design tool (Bohnacker et al., 2012). Generative design is also described as a paradigm that can be used by designers to explore processes such as self-organisation, swarm systems, and evolutionary systems (McCormack et al., 2004).

9 For example, see KABK, The Hague, Course Description Graphic Design, 1. Retrieved October 12, 2019 from kabk.nl/storage/documents/course-descriptions/Course-descriptions-Graphic-Design-year1.pdf, Communications Media, Fitchburg State University. Retrieved October 12, 2019 from codingpedagogy.net/chapter08/index.html, or University of Arts, London. Retrieved October 12, 2019 from arts.ac.uk/subjects/fine-art/short-courses/creative-process/creative-coding-introduction-to-processing-online-short-course-csm#.

Celestino Soddu uses an analogy to explain the way that generative design systems function. In the field of generative design, design concepts are represented as code. In his 2002 paper, Soddu argues that the code of a generative art or design project functions as its DNA that can generate a multiplicity of products. The artwork or product itself is merely an Idea-Product, a phenotypic expression of the underlying code genes. Each of these phenotypic individuals form a species that is identifiable by one design Idea (Soddu, 2002).

(2.3.2) PARAMETRIC DESIGN

According to Robert Woodbury, parametric design is a process based on algorithmic thinking, in which the expression of parameters and rules are enabled, that in turn define the relationship between the initial design intent and the resulting design response. It is a paradigm where the relationship between elements is used to manipulate the design of structures.

An exemplary breakdown of parametric design is shown by Reas et al. in the book *FORM + CODE* (2010). First, one has to decompose a form into variables. A chair, the given example, has variables for the seat, the legs and the back. A unique chair exists for each combination of variable values, that are all in the design space: a range of possible chair designs within the design system.[10] Randomizing these variable values is a way to explore the design space (Reas et al., 2010).

(2.4) DESIGN AGENCY AND ALGORITHMS

This section explores the agency that algorithms have over a creative work. Within agency ownership, authorship, authenticity and autonomy are described.

In October 2018, an "AI artwork generated by an

10 For the chair illustration, see formandcode.com/code-examples/parameterize-chair. Retrieved September 17, 2019.

algorithm" was sold at an auction at Christie's at $432.500 for over 40 times the estimated value (Cohn, G., 2018 and Christie's, 2018). The sold work, *Portrait of Edmond Belamy*, was officially created and owned by French art collective *Obvious*, even though they replaced the artist's signature with the algebraic notation of the GAN algorithm that trained their work.

$$\min_G \max_D \mathbb{E}_x[\log(D(x))] + \mathbb{E}_z[\log(1 - D(G(z)))]$$

(fig. 6) Signature on the bottom of *Portrait of Edmond Belamy*. *minG* describes the generator and *minD* the discriminator.

However, this exact GAN was written by the 19 years old open source developer Robbie Barrat, who had not been credited or compensated for his work and who in turn based his work on code and concepts by Soumith Chintala, Ian Goodfellow and frameworks such as Tensorflow owned by Google (Radford, A., 2015). Next to that, the database on which the algorithm trained contained 15.000 portraits painted between the 14th and 20th century. The input was thus originally crafted by a large number of different painters, who in the end affected the aesthetics of *Portrait of Edmond Belamy*.

This event gave way to the questions (1) who owns the work? and (2) who is the author of a work of a machine learning algorithm? The first question is one that refers to legal ownership, and the second to creative authorship. This is relevant for this research because the same questions of agency and ownership apply, as it is aiming for a creative symbiosis between an ML framework and artists or designers.

(2.4.1) LEGAL OWNERSHIP

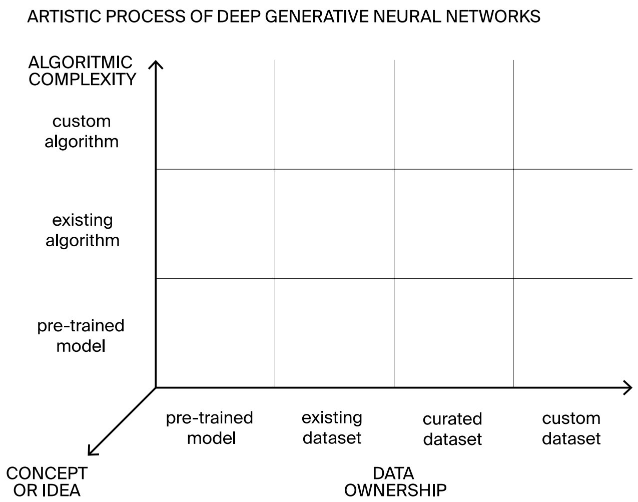
The question of ownership can be answered by law as code and repositories are released with a license. In the case of open-source software, the copyright holder grants anyone the right to study, change, and distribute the software to anyone and for any purpose (St. Laurent, 2018). The code written by Barrat had an open-source license as well as the code of Chintala.[11] *Obvious* was thus legally allowed to use and even sell products from the software as long as the license stayed attached to the code and could therefore call themselves the owner of the product that the code had produced. After the turbulence around the auction, Barrat has updated his license.[12]

(2.4.2) AUTHORSHIP

The second question regarding authorship is more fundamental than legal. In a late 2018 interview by Artnome's Renée Zachariou, machine learning artist and pioneer Memo Akten suggests a spectrum to approach the creative process of working with deep generative neural networks (fig. 5). In his description, three factors are at play: firstly, the algorithm, then the dataset, and thirdly the idea or concept. The complexity of the algorithm at hand can range between pre-trained models that can run at a click, an existing, an off-the-shelf or lightly modified algorithm and a custom or heavily modified one. The data ownership can range from a pre-trained model, to an existing dataset available online, to a curated dataset or a dataset of which the content is custom made. A combination of any of the above is possible, and Akten argues that with any of the combinations an original work can be made. However, the lower the agency in the former two categories, the more original the idea has to be to create a unique work. Thus, the role of creative authorship remains—next to control over algorithm and data—one of originality of the artistic concept (Zachariou, R., 2018).

11 The MIT License of the DCGAN by Chintala can be found on github.com/Newmu/dcgan_code/blob/master/LICENSE. Retrieved September 17, 2019.

12 The now updated BSD License of the art-DCGAN by Robbie Barrat can be found on github.com/robbiebarrat/art-DCGAN/blob/master/LICENSE.md. Retrieved September 17, 2019 from.



(fig. 7) Chart of the definition of the artistic process when using deep generative neural networks. Idea suggested by Memo Akten, chart inspired by an image of a Keynote presentation by Helena Sarin based on the same text.

The agency of the artwork has been a matter of debate in the art scene long before machine learning techniques were accessible. But now more factors are at play: Who is the author of an artwork generated by a computer program that can adapt and learn in response to external stimuli? Of course the developer remains the author of the code, but in the case of GANs a computer can learn how to make its own decisions, choices that the developer initially never anticipated or designed. Therefore, the program itself might also be the author, as well as being the artwork, at least to a certain extent.

However, as the paper by McCormack, Gifford & Hutchings *Autonomy, Authenticity, Authorship and Intention in computer generated art* (2019) argues, how authorship is defined might rely on the current paradigm of what an individual is. The contemporary notion of authorship was only present from the 17th century onwards, when Thomas Hobbes described a person as "He whose words or actions are considered either as his own or as representing the words or actions of any other thing to whom they are attributed, whether truly or by fiction".[13] Hobbes' definition of authorship was one of power, excluding women and indigenous people.

13 Found in McCormack et al., (2019).

Perhaps a self-learning computer program can only become an author once it will be perceived as an individual.

(2.4.3) AUTHENTICITY AND INTENTION

There has been strong debates about the authenticity of computational design and art. The lesser control a designer has over software, the lesser agency he or she can have over the visual outcome. McCormack et al. signify three elements of criticism:

(1) Works made with similar algorithms look alike. François Chollet has coined the term *GANism*, art made by GANs, referring to the visual 'sauce' that these tools make.[14]

(2) Artists working with algorithms do not fully comprehend the process of creation, therefore they often misinterpret characteristics of the output.

(3) Works cannot have an artistic take due to their heavy dependence on a generic technical process, originated from another, unrelated context.

Generally, as the accessibility and modifiability of an algorithm that produces a certain result decreases, creative agency shifts to the algorithm itself. The opportunity for artistic authenticity shifts to other fields: the curation of the dataset, the parameters that define how the algorithm is executed, as well as the creative process (McCormack et al., 2019).

Authenticity, intention and the level of intervention by the artist has been a question long before machine learning algorithms and computational art. In the early 20th century the dadaist movement, known for their so-called "ready-mades", for instance Marcel Duchamp's Fountain, questioned the long-standing tradition of craftsmanship, as seen in, for example, Salvi's Trevi Fountain (1762). Both are excellent artworks, yet one took years to create and the other could have been completed by a trip to the hardware store. McCormack et al. (2019) argue that it is the intention of sincerity that is important for authenticity, as they define authenticity as a "trait inherent in the work of the artist [is] the necessity of sincerity; the necessity that he shall not fake or compromise". However, a computer program's only intention is—to present day—to complete its dedicated task.

(2.4.4) AUTONOMY

Margaret Boden distinguishes two different types of autonomy, in an analogue context as well as in computational art and design: physical autonomy as exhibited in homeostatic biological systems, and mental/intentional autonomy based on human free will. In generative art or design, both these types of autonomy are at play, may it be differently so than in real life. The first type of autonomy is self-organisation of the behavior of the system itself, that is expressed through concepts such as agency, adaptation and emergence. Self-organisation is also described by McCormack et al. as inherent to generative systems. The second type is inherently connected to human freedom and intention, something that cannot be reproduced by a machine just yet. Only when a computer program would be freely able to decide to make art, would it fall under the second type of autonomy.

GANs have freedom of choice within the given dataset they train on: the one that the generator attempts to mimic as they map the dataset to a latent space. Although compared to a more direct type of computational design, such as generative design, a bit more autonomy can be attributed to GANs. The latter still has little autonomy outside of the dataset and there is no evidence that they possess any free will or intentional decision-making capacity, nor ever will.

14 See Chollet's Twitter post on July 13, 2017: twitter.com/fchollet/status/885378870848901120 Retrieved August 23, 2019.

(2.4.5) DEEP LEARNING AND GENERATIVE DESIGN

McCormack et al. argue that there are currently no significant new elements introduced in the algorithms of GAN that would systematically show an augmentation of agency compared to other machine learning systems. The term “intelligence” within “artificial intelligence” is connotated with intent more than ‘generative’ would, and perhaps this connotation is overrepresented in the media. Whereas there is a fundamental difference in parametric control within a machine learning system and an average generative design system, the GAN remains an algorithm that has no intent of its own. The level of autonomy, authenticity and authorship remains attributable to the artist or designer working with the technique and their intent, idea and concept, how much of their code was self-written, the way they deal with the database and the creative process.

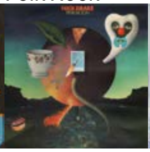
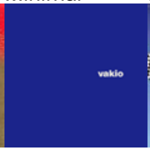
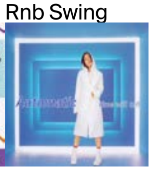
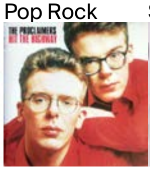
(2.5) ALBUM COVERS

Album covers and album art(work) describe the visual front of a musical release. The term can refer to the physical front of LP or CDs as well as the accompanying image of a digital music file. Album artwork is a resourceful format for this research because of three main reasons. First of all, cover artwork is used as an expressive canvas for designers more so than many other design formats. Secondly, the covers essentially have the same, square format: an essential aspect for training GANs is the (a)similarity of the input data. Thirdly, a large enough database can be very likely composed from existing online databases with album cover imagery. Apart from high availability, album covers pose challenges as a medium as well. As compared to frequently used image datasets for training GANs such as the CelebA or MNIST datasets, album covers tend to have an enormous variety of elements in them, as well as aspects that tend to be challenging to reproduce by a GAN such as different design styles and typography. Furthermore the lack of a high resolution and labelled training dataset are a challenge as well.

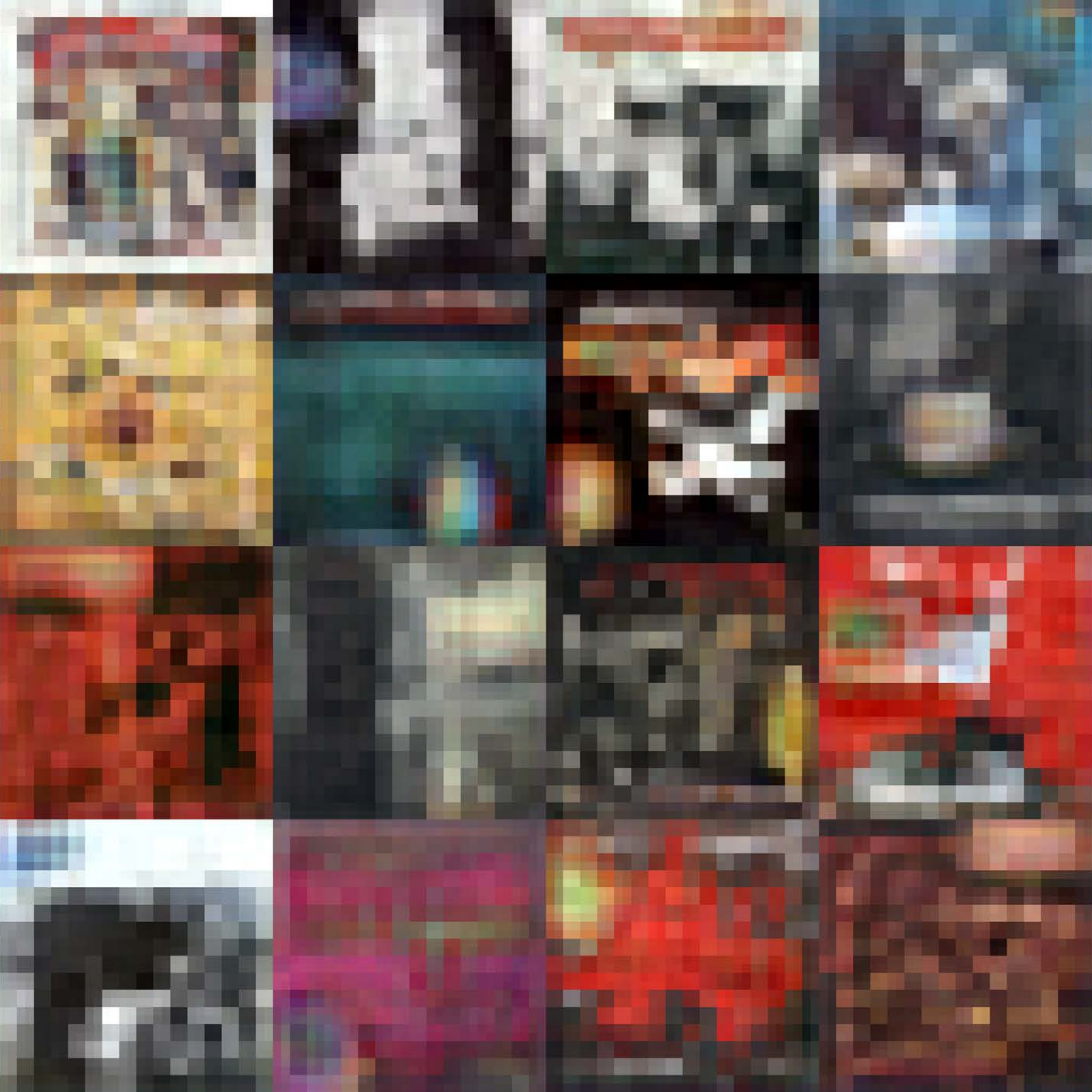
(fig. 8) Album covers left to right: Rock > Psychedelic, Electronic > Trance, Hip-Hop > Gangsta

(2.6) (SUB)GENRES

Throughout this paper, music and its album covers are categorised into genres, that in turn are divided into subgenres. The genres are not attributed by the author of this paper, instead the structure of the curated dataset follows the division as made by users on the Discogs platform. As musical genres have a certain discernible style, so does their artwork: an algorithm can distinguish the visual style that is inherent to it (Hepburn et al., 2017) Therefore, multiple genres and subgenres are part of this research to shed a more faceted light on the possibilities of this technique: perhaps certain visual styles are reproduced with more success than other ones.

Classical	Baroque	Classical	Modern	Romantic
				
Country Folk	Country	Folk	Folk Rock	Vocal
				
Electronic	Ambient	Minimal	New Age	Trance
				
Hip Hop	Conscious	Gangsta	Pop Rap	Rnb Swing
				
Jazz	Big Band	Fusion	Smooth	Swing
				
Pop	Europop	J-Pop	Pop Rock	Schlager
				
Rock	Experimental	Heavy Metal	Psychedelic	Punk
				

(tab. 1) An overview of the genres, subgenres and a random sample album cover artwork.



(3) METHOD

This chapter is divided into two main sections: section (3.1) concentrates on the creation process of the generated covers and their context and section (3.2) on the reflective part of the method achieved through an empirical study.

(3.1) PART I

The first section of the method is the practical part of setting the GAN and make it render proper album covers. The process accommodated the following steps:

(1) Selection dataset and metadata from Discogs using a scraper.

(2) Curation dataset and adapt metadata.

- (2.1) Remove empty images (where cover is missing in the folder)
- (2.2) Remove album covers that are just a scan of an LP (by sorting)
- (2.3) Resize all images to 512 x 512 (a power of 2)
- (2.4) Run through the metadata files and remove the entries of “empty images”, image files that were not found on Discogs yet stored as a file by the scraper.

(3) Preparation GAN and find correct hardware.

- (3.1) Use a GPU with at least 11 GB of temporary RAM, with a driver. A NVIDIA GeForce RTX 2080 Titan was used for this project.
- (3.2) On a Windows or Linux computer, install Tensorflow, CUDA, cudnn.

(4) Train process GAN on the entire database.

(5) Train process GAN on specific (sub)genres.

(6) Navigation in the latent space for samples.

(3.1.1) SELECTION DATASET

The selection of the dataset has been approached with a few aspects in mind. First, the dataset needed to come from an open source community, as there would not be any problems with authorship or with conflict of interest. Next to that, the image files needed to be from a reasonable quality and have a high resolution. On top of that, the images should have somehow been linked to their metadata, making it possible to filter them by music genre or by other aspects. It was preferable to curate a dataset for this project specifically instead of using an existing dataset, as the image cover datasets that were publicly available had a low resolution or were lacking metadata on each image.

Discogs is an online open source music sharing platform with an active user community that has turned it into a vast crowdsourced database for music recordings, that includes commercial, promotional and labelless releases. It currently contains more than 11 million releases by over 6 million musicians, contributed by about half a million user accounts.[15] The content shared on the website is uploaded per release, each one normally containing the following information: album title, name of artist, release year, genre, style, album cover image, type of release and a unique ID number. As content is generated by users, the content on the platform that initially started with solely electronic releases has now slowly evolved into a diverse spectrum of different genres and styles. The most frequently covered genre is rock with over 4.5 million releases.[16] The releases are divided into normal and master ones, where the masters are releases that have been uploaded several times, which is generally the case with albums from popular artists.

15 More information on Discogs can be found on discogs.com/about, discogs.com/search and discogs.com/stats/contributors. Retrieved September 15, 2019.

16 Exploring Rock. Retrieved September 15, 2019, from discogs.com/search/?genre_exact=Rock.

An application programming interface (API) is included in the platform, making the content of the platform accessible to be requested by script. Using the API to send requests to the server is protected by a key that is sent upon subscription, and has a rate limit of one call per second for the necessary content.

A Node.js scraper was written as shown in (9.1), in order to collect the images of the album covers at the one second interval required by Discogs, folderized per genre and subgenre (named style on Discogs). The scraper named the images after the unique ID, allowing to link each image back to their metadata in the JSON files that were downloaded simultaneously.

The scraper was used to compose a database of seven genres, divided in four sub-genres each, resulting in a total of 28 subgenres. The genres were chosen pragmatically by the largest amount of releases available on Discogs per genre. The JSON files were saved per page on the Discogs API, using the dataset's maximum of 200 entries per page. For reasons not described in the developer documentation, the API always ended the scraping session after 10.000 queries, but that amount is sufficient to create a database large enough for the purpose.

Classical	Baroque	Classical	Modern	Romantic
Country	Country	Folk	Folk Rock	Vocal
Electronic	Ambient	Minimal	New Age	Trance
Hip Hop	Conscious	Gangsta	Pop Rap	RnB
Jazz	Big Band	Fusion	Smooth	Swing
Pop	Europop	J-Pop	Pop Rock	Schlager
Rock	Experimental	Heavy Metal	Psychedelic	Punk

(tab. 2) Genres and respective subgenres of the album covers.

(3.1.2) CURATION DATASET

In order to generate a high-quality database, the scraper had to be set to specific parameters. As the content on Discogs is uploaded by different users, the image quality of the uploaded album artwork was very diverse in terms of image accuracy, repetition and image size and image ratio. The lower quality images had to be filtered out, this counted for each of these factors.

(1) *Image accuracy*: images that are scans of the LP or CD rather than an image artwork were removed by setting a more specific search query, adding `format_exact=CD` to the URL.

(2) *Repetition*: By setting the query to `type=master` and adding this to the URL, there was no repetition per genre, meaning there were not multiple entries of different versions of the same album.
This would put more weight on the repeated images. Repetition across genres was allowed, as an album can simultaneously belong to multiple genres and/or subgenres. This occurrence is quite rare, maximum 15% of a sub-genre appeared in another sub-genre within that genre.

(3) *Image size*: The ideal width and height for the images to be used as dataset for the GAN would be a factor of 2 with a ratio of 1:1, ideally 512 x 512 pixels or 1024 x 1024 pixels. Most album covers uploaded on Discogs averaged more around the former, but the query also included exceptionally small image formats, for instance 100 x 100 pixels. Whereas images can be upscaled slightly, a threshold had to be set for images that were too small. The threshold was set to 256, half the size of the desired image size, and only images above the threshold were imported.

(4) *Image ratio*: Not all scraped images of album covers had a 1:1 width to height ratio. Images with a ratio between 1:25 and 1:1 were still included and given a black background to compensate for the difference and to be transformed to a square. Images with a larger difference in proportion were left out of the dataset.

After having scraped the entire database, there were still two elements that had to be adapted: removing all empty images and resizing the entire set of images to the

same size. Firstly, in case a Discogs release did not include an album cover artwork and a site-specific placeholder was used, the scraper had downloaded an empty image file. These were all removed by simply deleting all image files of 0 KB within the dataset, as well as deleting the corresponding metadata in the JSON files. Subsequently, the images were all automatically resized to 512 x 512 pixels, with the script in section (9.2). The final amount per genre and subgenre can be found in (fig. 3).

Classical 24.754	Baroque 5.767	Classical 5.903	Modern 3.860	Romantic 9.224
Country 28.095	Country 9.970	Folk 9.642	Folk Rock 4.473	Vocal 4.010
Electronic 1.4472	Ambient 2.182	Minimal 2.619	New Age 2.238	Trance 7.433
Hip Hop 16.669	Conscious 2.461	Gangsta 4.130	Pop Rap 4.153	RnB 5.925
Jazz 8.359	Big Band 1.761	Fusion 2.260	Smooth 1.982	Swing 2.356
Pop 24.158	Europop 9.317	J-Pop 3.096	Pop Rock 7.907	Schlager 3.838
Rock 28.700	Experimental 5.830	Heavy Metal 8.196	Psychedelic 5.002	Punk 9.672

(tab. 3) In total, there are 145.207 album cover artwork images in the dataset, of which 124.485 (85.7%) authentic covers.

(3.1.3) PREPARATION SOFT- AND HARDWARE

Built on NVIDIA's StyleGAN (Karras et al., 2019), the framework had some requirements. First, the dataset had to be converted from images to TFRecord files, and secondly, the right configuration of soft- and hardware had to be in place.

To prepare the database, the StyleGAN framework provides a python script for image conversion.
This could be simply executed by adapting the source and destination directory in the following command: `$ python dataset_tool.py create_from_images datasets/custom-dataset ~/custom-images`.

The images were converted to TFRecords, a simple format for storing a sequence of binary records. In the case of the StyleGAN, this is a set of TFRecords files that store the dataset in the progressive image sizes needed for the code to train.[17]

For the hardware, a Windows or Linux computer had to be used with a GPU of at least 11 GB DRAM, with a fitting driver. For this research, an NVIDIA GeForce RTX 2080 Titan was used, not donated by NVIDIA.
Apart from that, the computer required ample storage for saving the models and process. For the software, an Anaconda environment was created, including Python 3.6 with modules Pillow, tensorflow-gpu, scipy, and requests. CUDA toolkit 9.0 and cuDNN version 7.3.1. A script was used to test the setup before starting to train.

(3.1.4) TRAINING PROCESS

During the entire training process, solar power was used for the training device as well as for the one on which weekly logs were made. Designing a system that aims for sustainability was essential and should by any means be considered a design issue inherent to machine learning experiments. Within the code of the StyleGAN, a few settings were adapted to match the dataset and the system. For example, the compatibility for the graphics card in use was set, as well as the location of the dataset. Since this first part of the training was executed during a heat wave, the starting point of the training had to be set to the last back up after the training process crashed due to overheating and had to be restarted. One other change was setting the `mirror-augment` variable to `false`. As the dataset contains element such as type that do not function when mirrored, this setting, normally used for doubling the dataset by mirroring it, was unnecessary.

17 More on TFRecord and tf.Example on tensorflow.org/tutorials/load_data/tfrecord. Retrieved October 21, 2019.

(3.1.5) TRAINING PROCESS (SUB)GENRES

Each genre was trained on separately, and subsequently each subgenre. The duration per genre depended on the frechet score and was also decided based on the visual progress. Genres showed adaptation much slower than subgenres, with subgenres showing a significant change in visual style in less than 12 hours.

(3.1.6) NAVIGATION LATENT SPACE

A custom script, displayed in section (9.3), was built to navigate the latent space. A part of the function was based on the tutorial *How to Explore the GAN Latent Space When Generating Faces* (2019) by Jason Brownlee. Written in python, it chooses a random point or can take an input image that it tries to approximate in its space. The latent space has 512 dimensions with a value of [-1, 1], and the script can be set to interpolate between a set of random points in the latent space in a number *n* of given steps.

(3.2) PART II

The second section of the method describes the practical part of setting the GAN in such a way that it renders acceptable album covers. A survey had to be made and the following steps had to be undertaken:

(1) Define the target group for the survey.

(2) Define the requirements and restrictions for the survey.

(3) Create survey using real and existing album covers.

(4) Share survey:

- (1.1) During lectures.
- (1.2) Mailing list from first experiment.
- (1.3) Friends and peers.
- (1.4) Facebook group: Generative Design Research Network.

(5) Receive results (n > 25).

(6) List results.

(7) Based on results, compare real and generated album cover on scores.

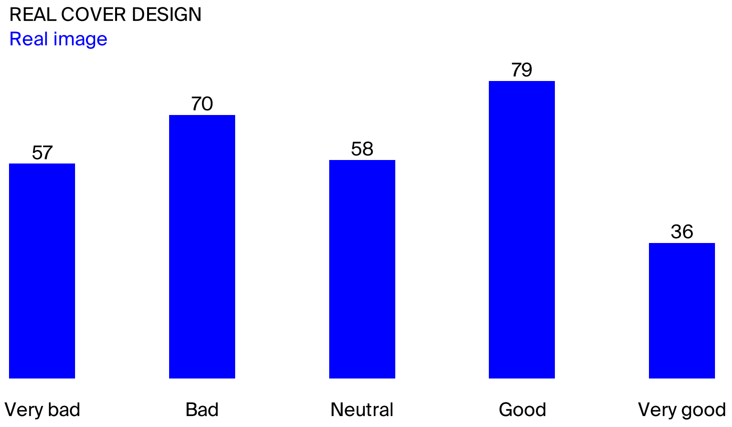
(3.2.1) TEST GROUP

The target group of the survey is people with a general interest in graphic design, preferably with a design-related profession. The age span for the target group is 18 to 65 years. The age target is set this way because the focus lies on working adults, but if participants of any age outside this target group are willing to participate they were included. While the research is held in The Netherlands, the target group is not limited geographically or exclusively focused on Dutch citizens. Current city and country of residence are asked instead of birthplace and nationality, because the tendencies of current surroundings might affect the preference for certain design choices of an individual more than origins as design styles are trend-sensitive and change rather quickly. The focus lies on Western design as the music covers mostly are for Western music genres, so Western countries of residence are the main focus of the target group. Any gender is accepted and included, and one is able to add their gender if it is not in the list of options or not enter it at all. To make sure all participants are people and not bots or scripts, entering a mail address is required to submit the survey.

(3.2.2) REQUIREMENTS

Before the final survey, a test survey was performed on June 27, 2019, at a Work In Progress event in Amsterdam.[18] This test was utilized to research what people's general opinion is on real images of album cover artwork from the dataset. Most of the attendees were designers, artists or creatives, thus it was fitting the intended target group. Twenty people assessed fifteen covers each from a subsection of the entire dataset in which each music category was equally represented.

18 See more of the event on facebook.com/events/404819966777873/. Retrieved June 28, 2019.



(fig. 9) The result of the first user test, feedback on the covers in general.

The final survey needed to address several aspects within the design practice in order to cover the elements of “good” design. The design has been divided into the categories colour, typography and composition, based on the feedback on the test survey. Next to that, the design as a whole needed to be addressed as well. The scale of these elements of the survey was set as Very bad, Bad, Neutral, Good, Very good.

A second feedback moment was organised to further improve the final survey with a group of five people. The group consisted of a Media Technology student, two designers, a data science student and a psychology student who had previously followed a course on survey-making. One recurring suggestion was the ability to select which specific elements were considered favorite or least favorite alongside the scales for the predeclared aspects. Another suggestion was the ability to select a favorite cover from a selection of an even amount of real and generated covers. Finally, there was a suggestion to distinguish between the general style – more focused on the feeling or general impression – and the design as a whole, more focused on the design aspects combined on the entire cover. As it happens, the style of an image can be good when the design is bad and the other way around. These three suggestions were all implemented in the final survey. The final survey can be found in section (9.4).

(3.2.3) STRUCTURE

When loading the survey website[19] a generated cover is shown for two seconds after which the participant is redirected to the survey automatically. Before initiating the survey, a short disclaimer appears stating what data will be collected, by whom, the purpose of collection, and guaranteeing secure storage of the data. Users are directed to the test if they give consent. Metadata such as browser type and IP address are saved separately from the entries to keep track of possible hack attempts or bots and to get an impression of which medium was most effective for sharing the survey.

(fig. 10) Introduction and explanation of the survey.

The survey itself is divided into four sections: (1) *General Information*, (2) *Real Album Cove*r and (3) *Generated Album Cover*, (4) *Favorite Cover*.

The purpose of section (1) *General Information* is to get an overall impression of the person submitting the survey and see if they approximate the target group as described in section (3.1.2). Perhaps personal aspects like professional background influence certain decisions. Time and date of each submission is saved automatically, just as general metadata,alongside the questions to be filled out by test users.

This is followed by two similar sections, one for a “real

19 The URL to share the survey was ganalbum.art.

image” and one for a “generated image” with exactly the same questions in the same order. In the final section, the user is asked to select a favorite cover from a set of three real and generated covers. In the entire survey, it is not mentioned which cover is real and which one is generated, nor is this asked to the participants as not to prime them about this distinction – although it is likely that they realise. On top of that, the research is shortly introduced so the use of generated imagery is not entirely hidden. The intention is to discover whether the generated covers are as good as human-designed covers, not whether it is possible to distinguish them.

(fig. 11) Select one favorite artwork, part four of the survey.

(3.2.4) IMAGE INPUT

For each music category, hundred images were randomly selected as input images for the survey, from both the real album artwork as well as the album artwork generated by the StyleGAN. For one subcategory per genre, hundred images were selected randomly as well for both the real and the generated album artwork. Only one (sub) category was shown per submission for the entire survey, as not to let the feedback for separate images be influenced differently by a person’s music genre preference. The feedback should focus on the aesthetic aspects rather than the musical reference. There was a 50% chance a survey would show a category or a subcategory. The participants are not shown the difference between categories

and subcategories. Images could be repeated different iterations of reloading of the survey, but not within one instance of the survey, meaning that there was no repetition of images in the two covers studied in detail and the series of potential favorite covers.

(3.2.5) METHODS

A qualitative version of the Likert Scale is used for the assessment of the design aspects.[20] The Likert Scale is used to measure respondents’ attitudes to a particular question. To analyse the data for qualitative purposes it is usually encoded as follows, after which the mean or most frequent response or the median is valid.[21]

(1) Very Bad
(2) Bad
(3) Neutral
(4) Good
(5) Very Good

For both subcategory and category entries, the user can attribute one of the seven main genres: Classical, Country/Folk/World, Electronic, Hip Hop, Jazz, Pop, Rock. Each of the three sections had a space for textual remarks of up to 160 characters, and there is also a space for general remarks. The favorite and least favorite aspects are multiple selection questions with the options Color(s), Composition, Typography, Image/Illustration, General Style and Other. One can type their own aspects in a text field by selecting “Other”.

20 Likert Scale Examples for Surveys, Iowa State University Extension. extension.iastate.edu/Documents/ANR/LikertScaleExamplesforSurveys.pdf. Retrieved November 1, 2019.
21 More on analysing Likert Scale data, st-andrews.ac.uk/media/capod/students/mathssupport/Likert.pdf. Retrieved November 1, 2019.

(3.2.6) RELATION RESEARCH

The survey reflects the factors as discussed in section (2.2.4). However, these factors cannot be directly translated to questions as the generated images in the survey are not an applied product yet, but rather a meta-product. Therefore, visual characteristics within basic formal theory are addressed to focus on the potential rather than the current functionality. Here are the questions in the survey mapped to the factors of a successful design product:

(1) The cover artwork has to communicate.
- (1.1) How is the functionality of the typography on the cover?
- (1.2) Which genre does the cover belong to?
- (1.3) What do you think of the design of the cover?

(2) The cover artwork has to be appealing to its target audience/incite curiosity.
- (2.1) From these six images, which one do you like the most?
- (2.2) What do you think of the hierarchy in the composition on this cover?
- (2.3) What do you think of the aesthetics/visual appearance of the typography on the cover?
- (2.4) What do you think of the combination of colors in this cover?

(3) The cover artwork has to be recognisable.
- (3.1) What do you find of the general style of this cover?
- (3.2) What do you think of the originality of the composition on this cover?
- (3.3) What do you think of the originality of the colors on this cover?

The first factor can be answered with the dichotomous (yes/no) question: is it understandable what is communicated? The second question is answered through differences participants' personal preference of aspects such as colors, type, composition and style. The third question depends on participants' opinion of originality and style.

(4) RESULTS

This chapter is divided into two main sections, that refer to two parts of the process just as in the previous chapter (3). The first part exposes the results of the training and the second the results of the survey.

(4.1) PART I

This section shows the results of the training of the model. After months of training, each genre has had a separate trained model that has a reasonable sample quality. All back-ups of all models have been mapped to the same coarse level. The plan was to first train the covers on the entire dataset, then on the seven genres and finally on at least seven subgenres. Before the initiation of this training plan, the training process itself was tested on a subset of the dataset. One of the largest subgenres, punk, was used to confirm that the settings of the StyleGAN were fitting and the training was improving. This test period lasted seven days and was successful.

(4.1.1) ENTIRE DATASET

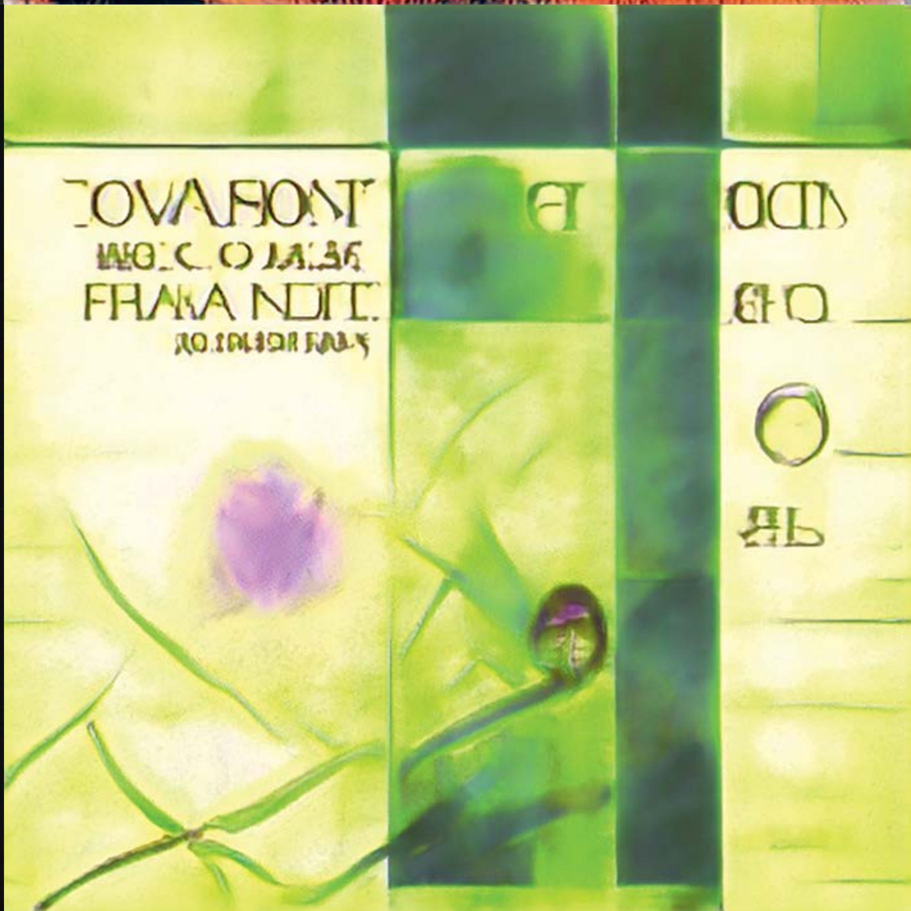
The StyleGAN algorithm was first trained on the entire dataset from the ground up. This part of the training process lasted for 21 days. The training on the entire dataset was completed when the total length of the training, measured in thousands of real images, was 15.000, so (15.000*1.000 =) 15 million images. This count was stored by variable `kimg` and checked at each interval where an image was saved.

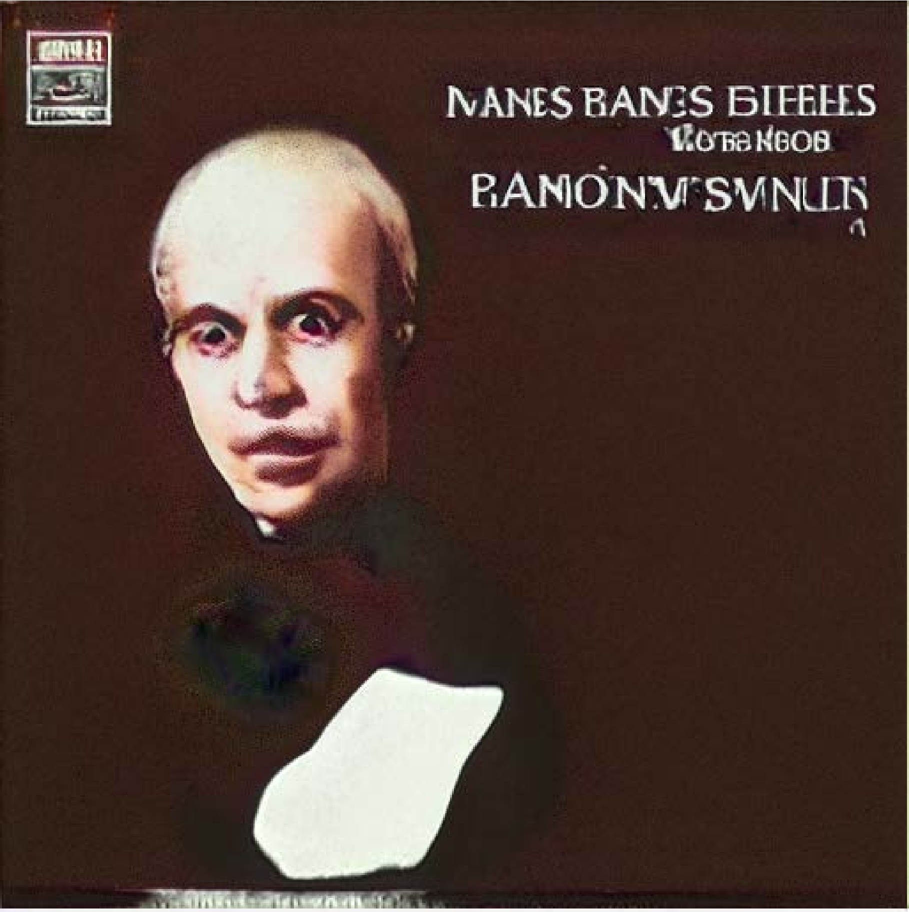
(fig. 12) Samples from the model after training on the entire dataset.

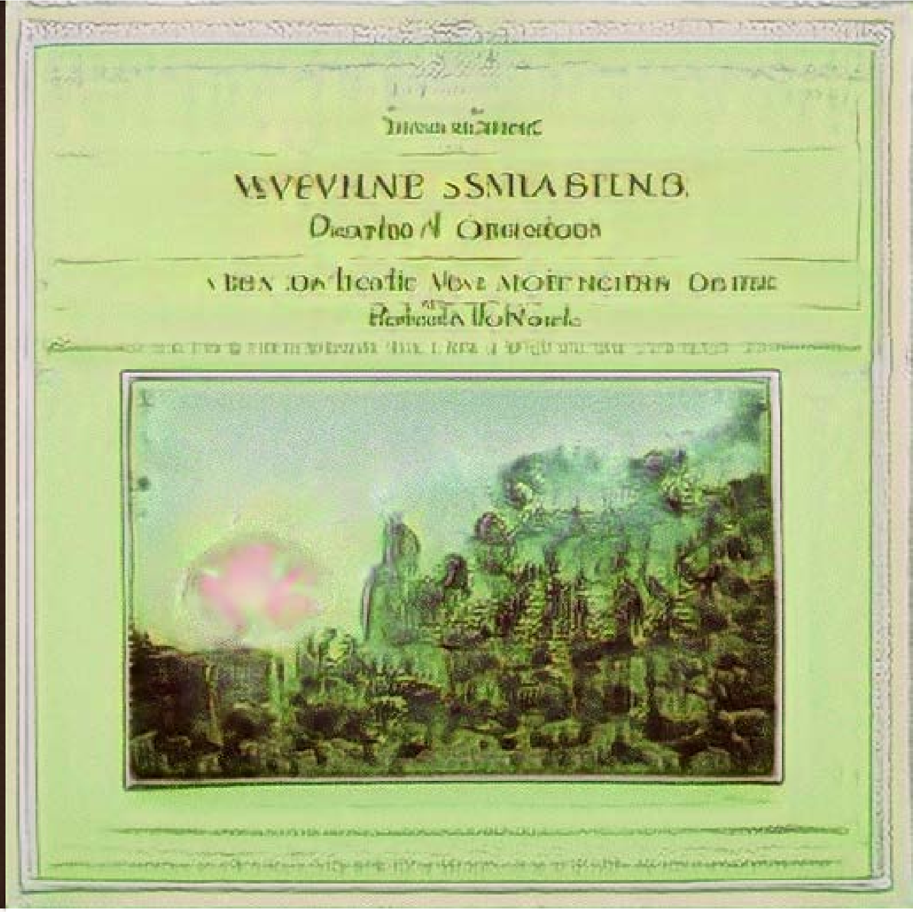










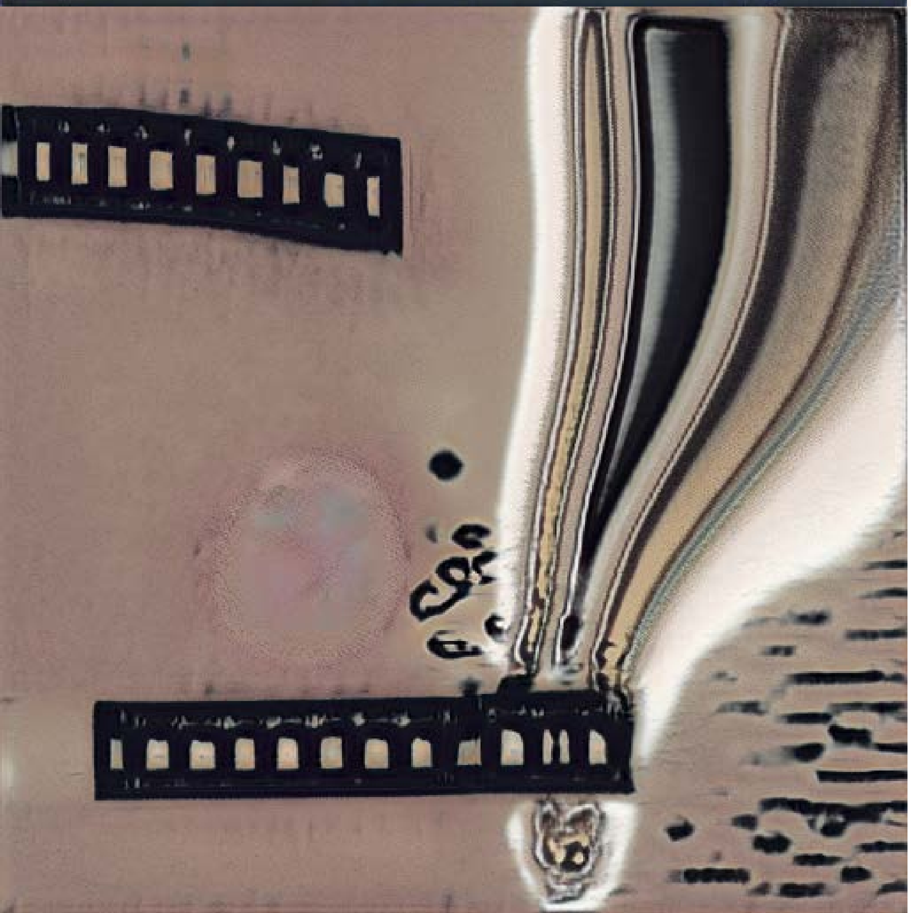


(4.1.2) GENRES

After training on the entire dataset, the StyleGAN was trained on the seven different genres. This time, the training was stopped manually instead of metrically. The last model back-up from the entire dataset training step from the previous section was used as a starting point for each genre. The training continued for five to seven days, until the differences between the last two snapshots were hardly distinguishable.

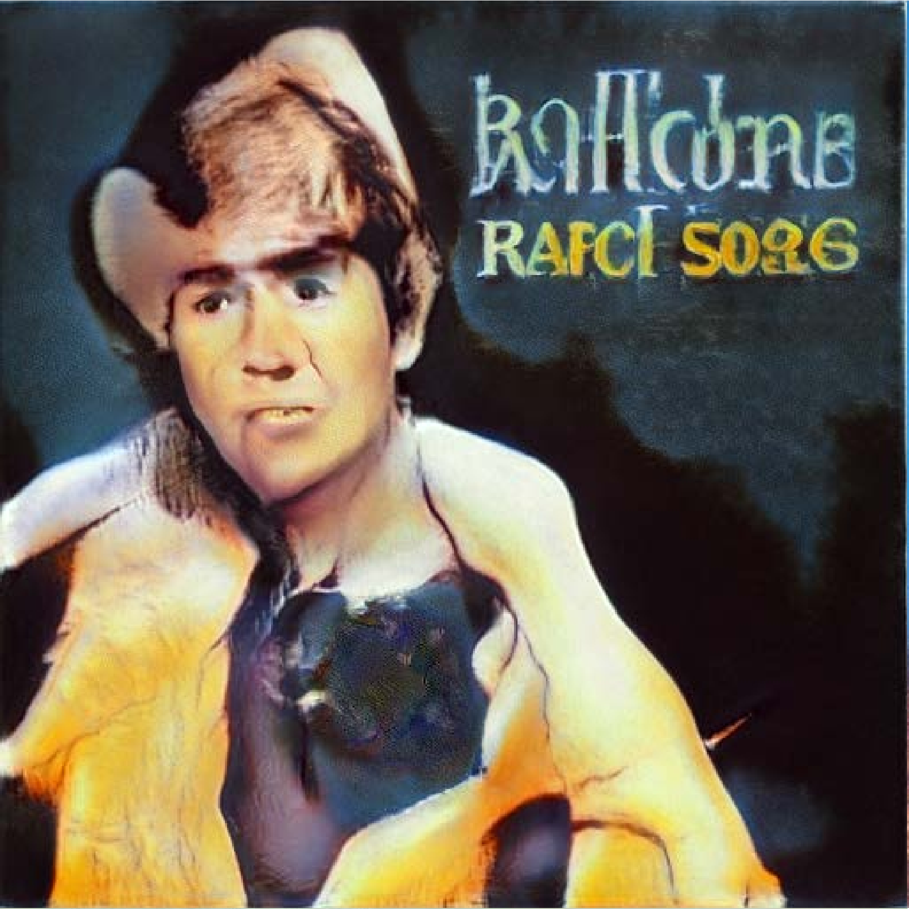
In the Classical generated album covers, many aspects become instantly evident. First of all, the colors are often dark shades or fully white. There is a distinct graphical style with centered images, serif type and emblems. On many points in the model, the face of a single white-haired, white man is depicted.

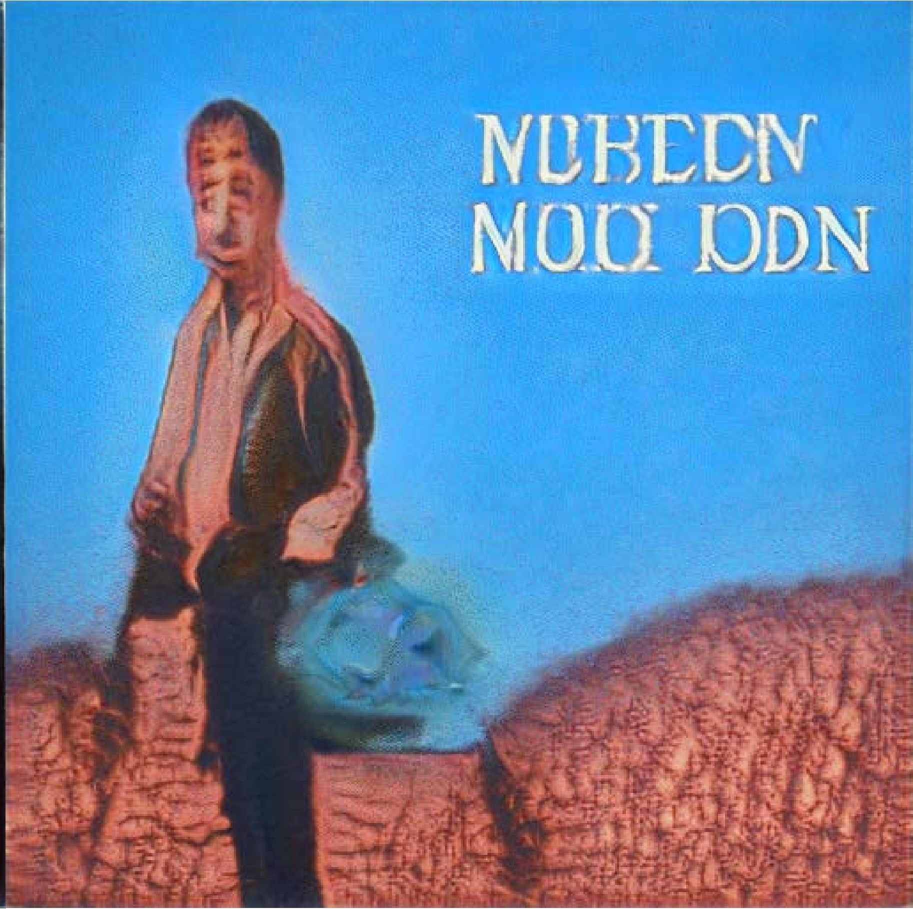
(fig. 13) Sample from Genre: Classical



The Electronic generated album covers are often abstract and alternating compositions in contrasting, saturated tones. The typography is either not present or quite minimal, kept to a single line of text.

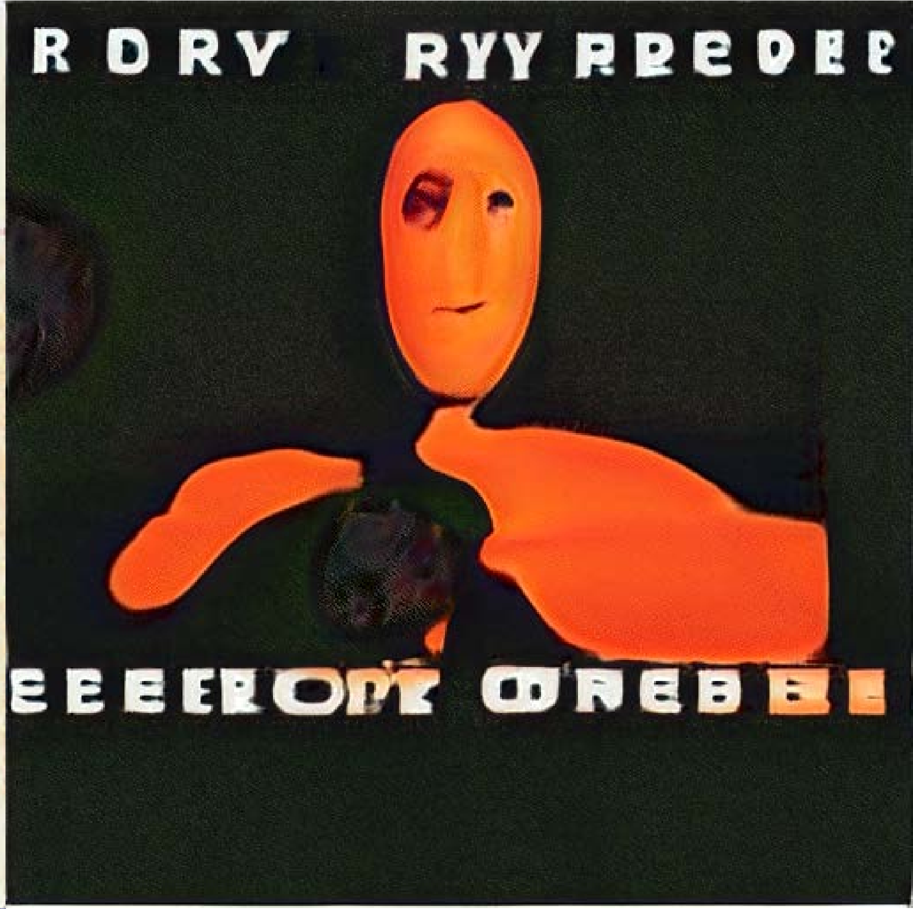
(fig. 14) Sample from Genre: Electronic





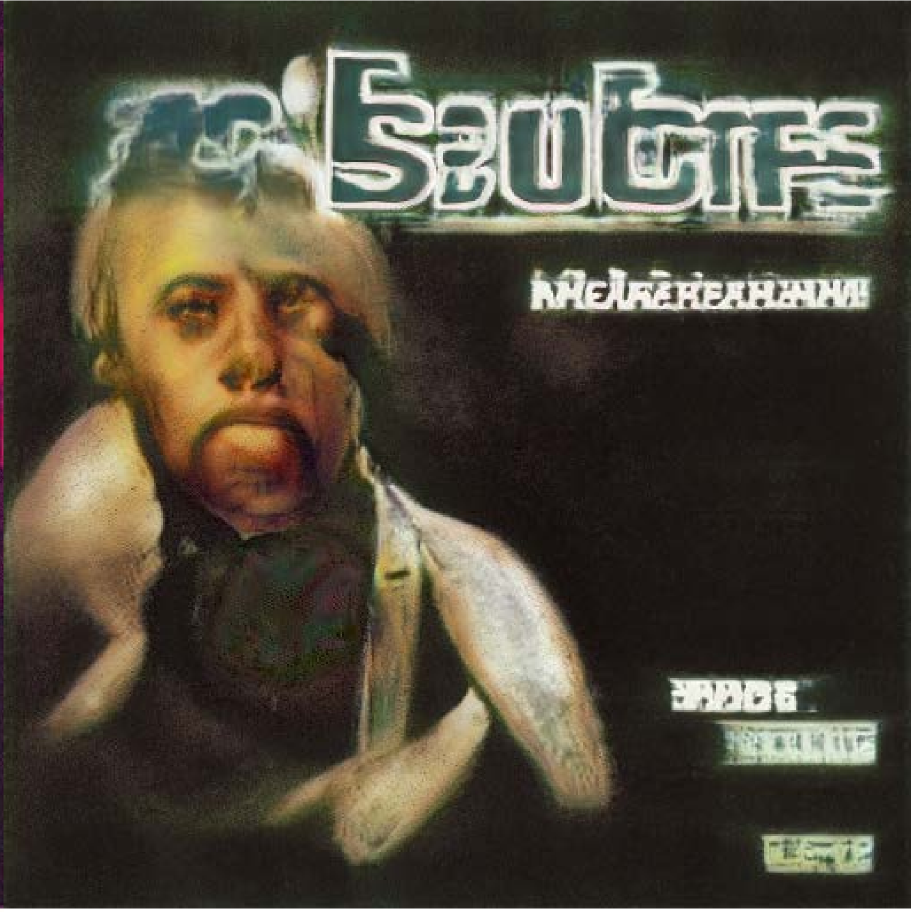
The generated Folk/World/Country album covers have unsaturated, beige tones, many different sorts of typography, and often depict one man or woman—sometimes with a cowboy hat—on a neutral background.

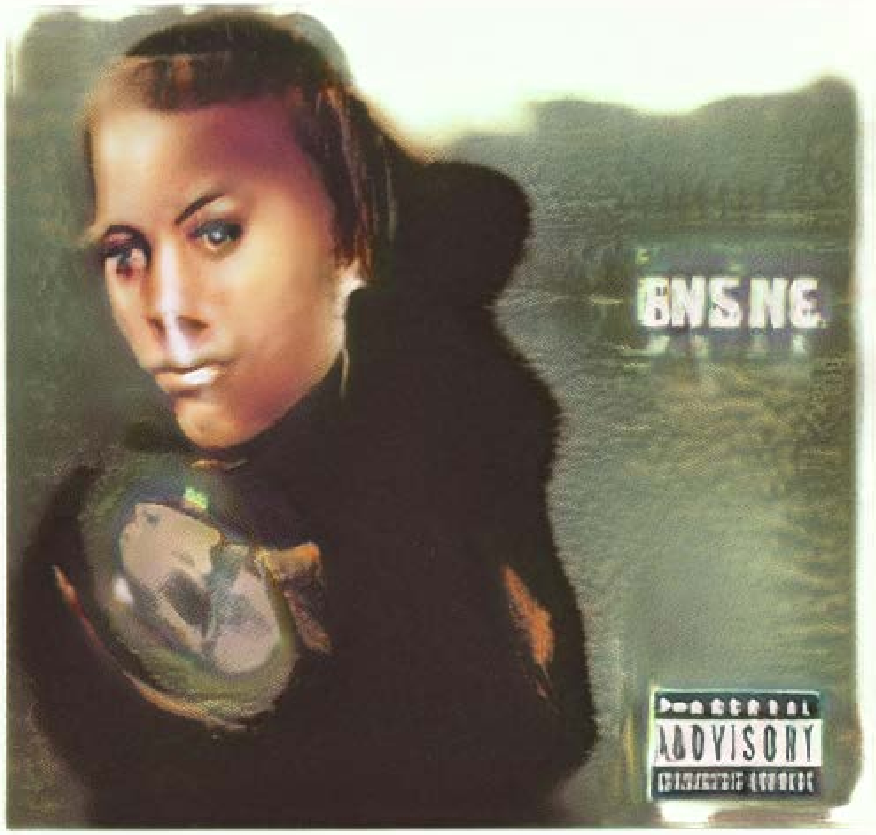
(fig. 15) Sample from Genre: Folk/World/Country



The generated Jazz covers use a lot of earthy tones, and have many different compositions and typography. Some depict groups of people, others a single person, and again others more abstract compositions. Remarkably, this dataset was the smallest from the seven genres, which is immediately visible in the lack of details and some sort of grainy texture covering the images.

(fig. 16) Sample from Genre: Jazz





The Hip Hop genre has a lot of generated images with different layouts as well, but here the colors are more limited to darker tones. Sometimes a single person or a group is depicted, sometimes the image is abstract. The faces on the samples often have a darker shade as well. The typography is often a bit bolder than in images from other genres. Remarkable here is the near-exact reproduction of the “Parental Advisory” mark, that is instantly legible.

(fig. 17) Sample from Genre: Hip Hop

Rock produced the most alternative samples from the seven genres. Some were rather abstract, some more figurative, Samples with bright colors, dark or pale colors were all present. Some samples had bold compositions, others more minimal.

(fig. 18) Sample from Genre: Rock

Generated images from a network trained on Pop use saturated colors, and almost exclusively depict at least one person. The appearance of this person varies, but is often a young man or woman. The typography is divergent per cover too.

(fig. 19) Sample from Genre: Pop

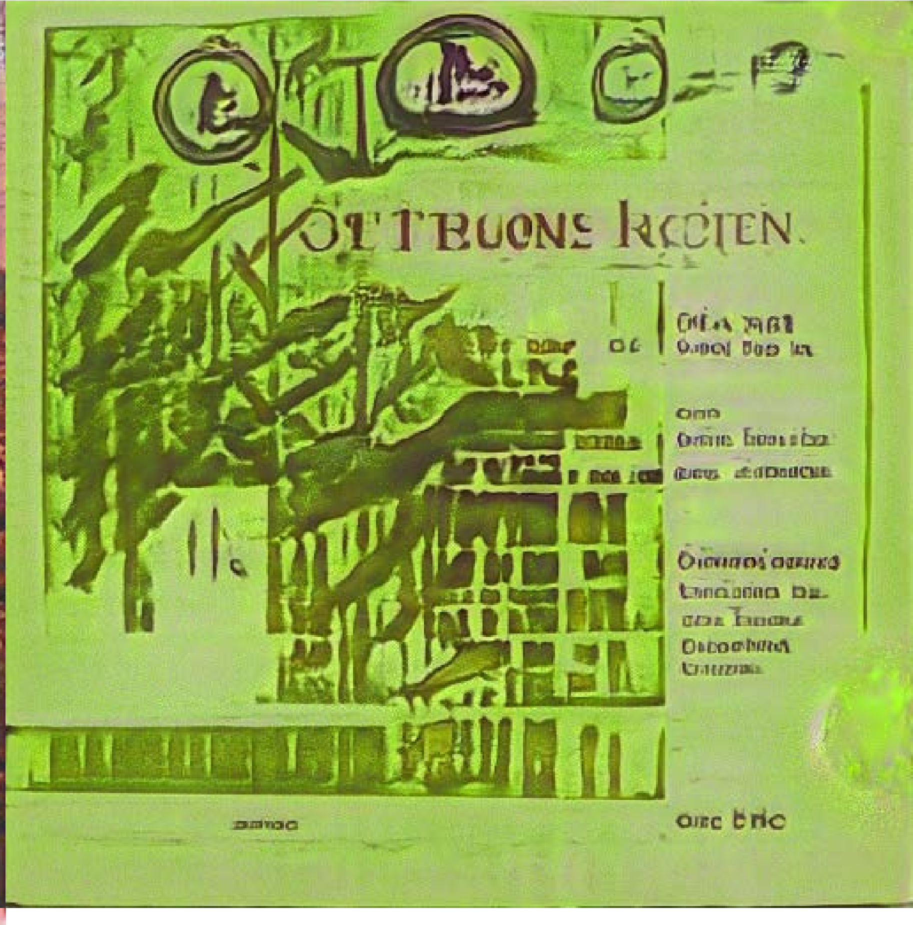
(4.1.3) SUBGENRES

For some subgenres the reproduction of characteristic aspects was even more evident. From each of the respective genres, a single subgenre with a substantial dataset and a distinct style was chosen to continue the training with. This resulted in the following subgenres:

Classical	Electronic	F/W/C	Hip Hop	Jazz	Pop	Rock
Romantic	Trance	Country	Gangsta	Swing	Europop	Heavy Metal

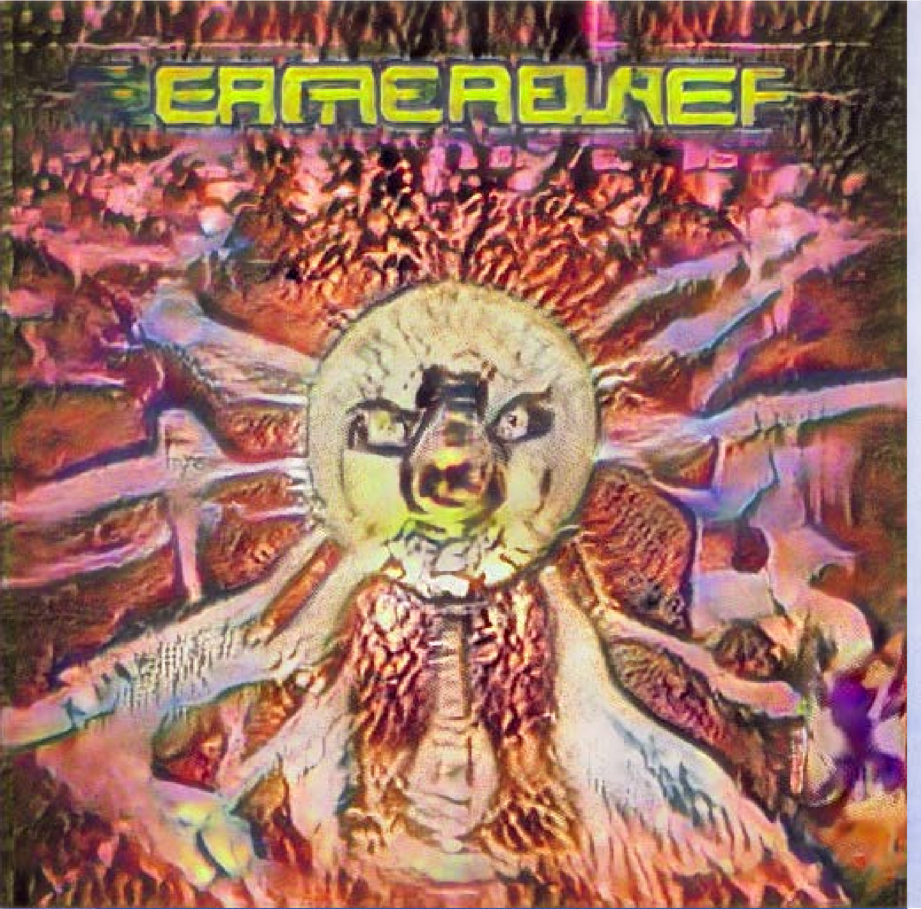
(tab. 4) Overview from chosen images.

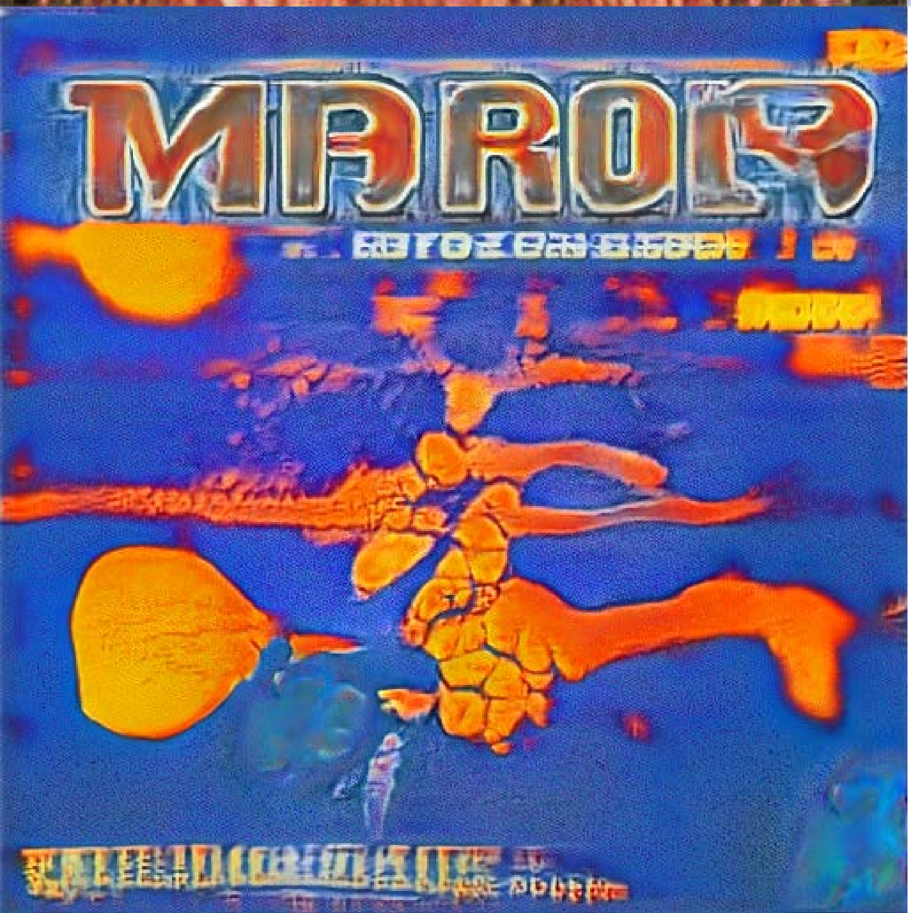
The training for each subgenre was continued from the last snapshot from its respective genre and continued for three to five days until the differences between the last two snapshots were hardly distinguishable.



Samples from the model trained on Romantic covers often depicted a birds-eye view of landscapes, or elements that look like water or clouds. Sometimes a person is still depicted, but these depictions became much less distinct, compared to the main category of Classical music album covers. Typography was almost exclusively serif letters and some logos or emblems became legible.

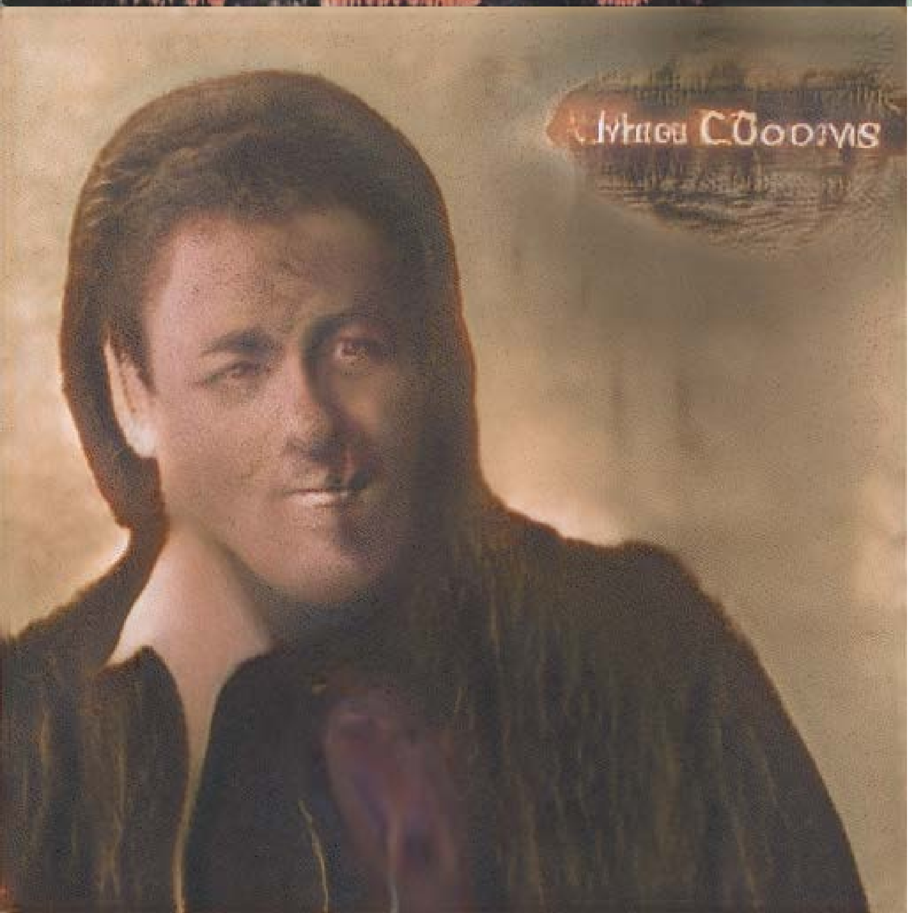
(fig. 20) Sample from Subgenre: Classical > Romantic





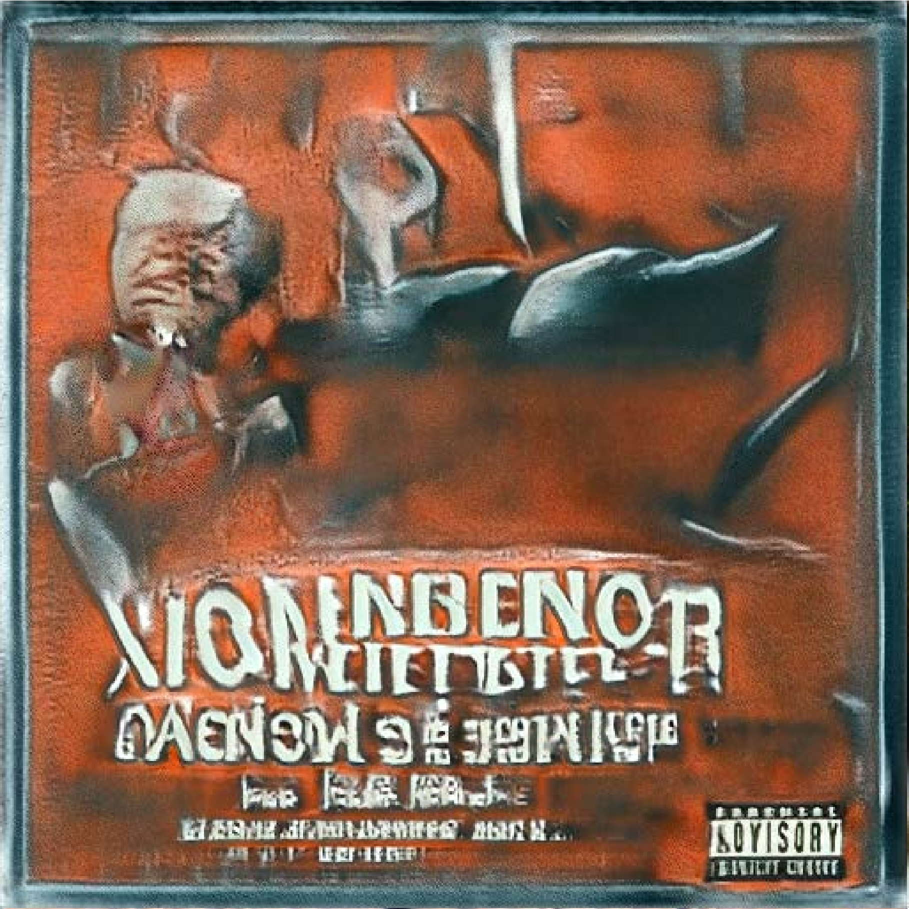
Samples from the model trained on Trance covers were mainly abstract compositions, and frequently combined many different saturated colors. The samples had a analogous style of typography with large text in multiple colors.

(fig. 21) Sample from Subgenre: Electronic > Trance



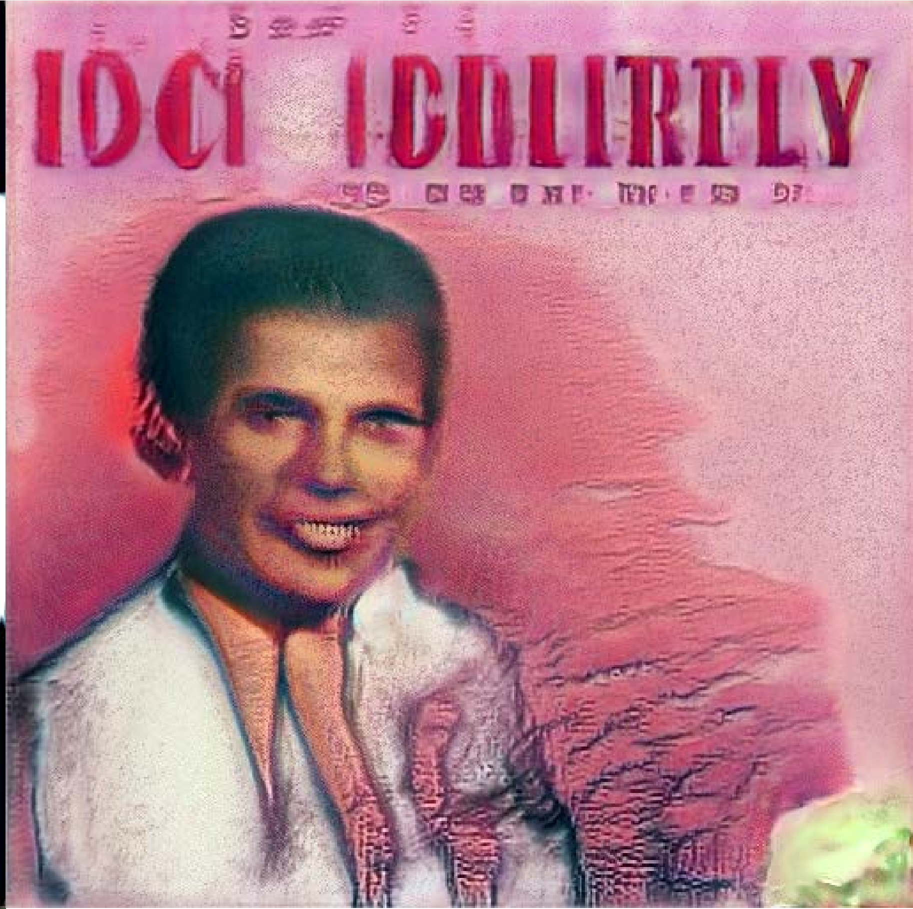
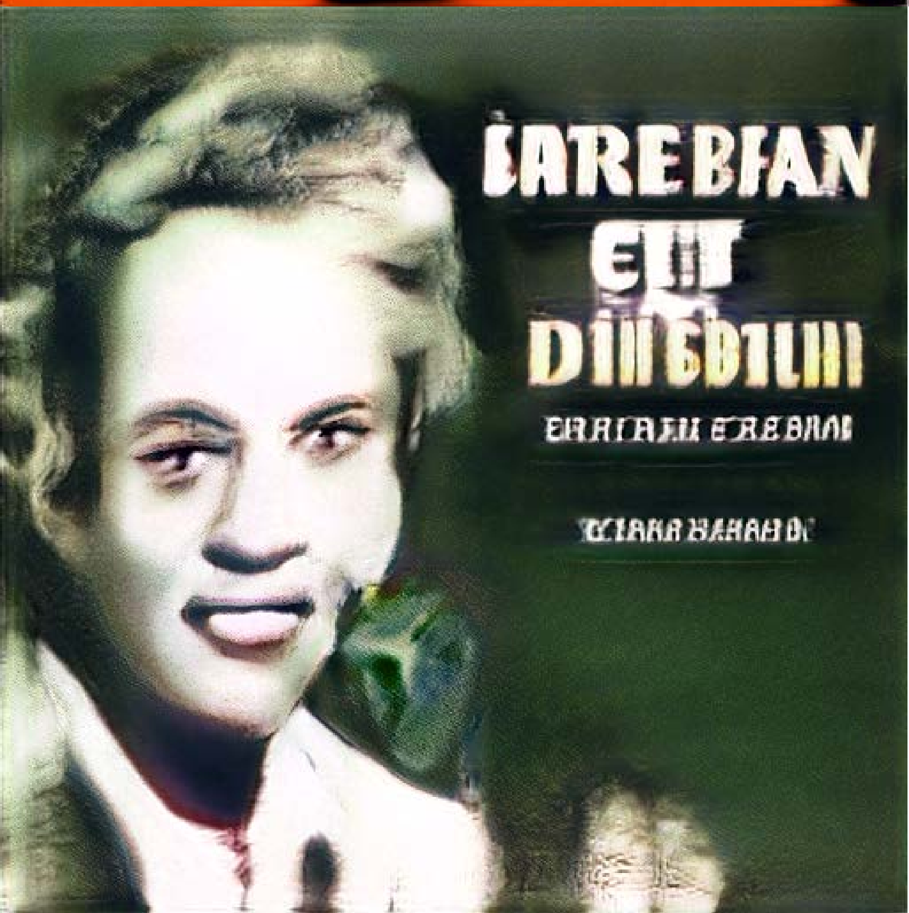
In the images generated from the model trained on Country covers, some cowboy hats are still depicted. The images are extremely similar to the ones from their main genre, maybe with even more beige colors. The typefaces are a bit more specific.

(fig. 22) Sample from Subgenre: Folk/World/Country > Country



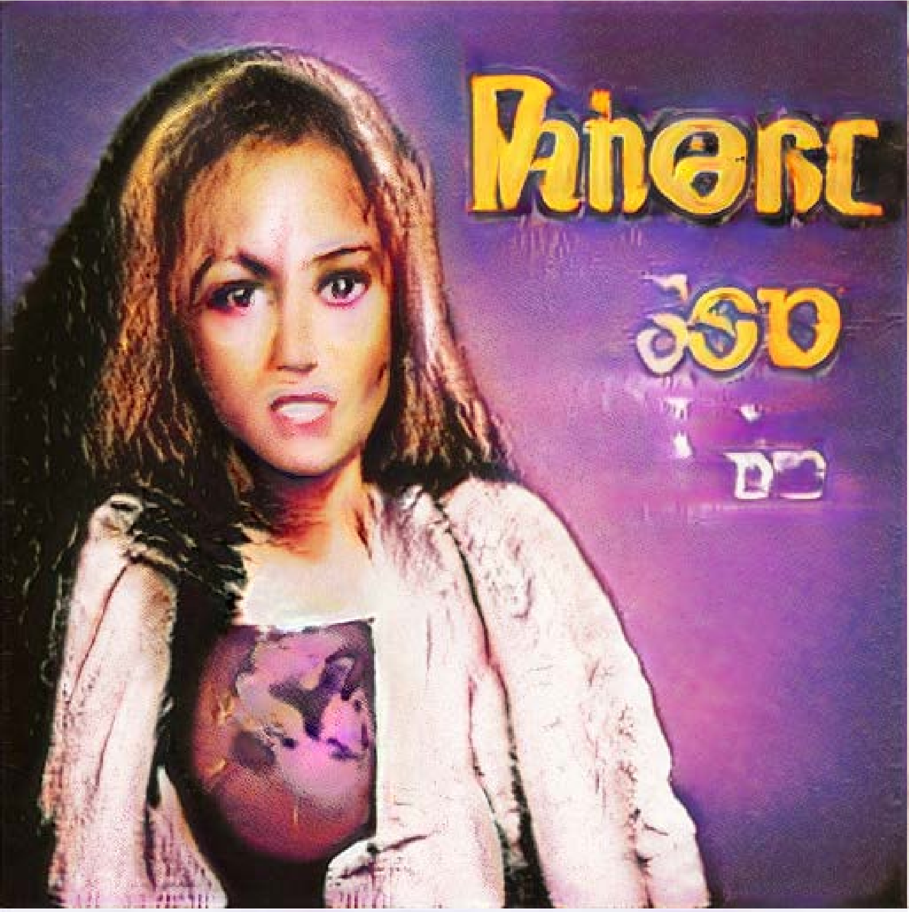
The model trained on Gangsta Rap has even more legible parental advisory stickers. The covers became much less sharp, but with differing typographic forms.

(fig. 23) Sample from Subgenre: Hip Hop > Gangsta Rap

The Swing subgenre has more specific faces and some distinct saturated, abstract covers.

(fig. 24) Sample from Subgenre: Jazz > Swing



Europop looks like its main genre as well, but with less clearly depicted faces. The typography is often bright and covers a big part of the text as well.

(fig. 25) Sample from Subgenre: Pop > Europop

Heavy metal samples have a distinct style, often with gothic typography, fire and skull-like imagery and dark colors. This subgenre might be most fitting to the uncanny face-like shapes as the input images use this as well.

(fig. 26) Sample from Subgenre: Rock > Heavy Metal

(4.1.4) FINDINGS

Certain findings became instantly clear during the training process. In this paragraph the most remarkable ones will be discussed.

The models managed to recreate letter characters perfectly legible, but the combination of letters had no semantic meaning. The style of the letter characters was very outspoken, especially after the `mirror-augment` training setting was switched off during the training process. This setting multiplies the dataset by mirroring each image, which is suboptimal for letter characters that are not symmetric.

(fig. 27) Samples of distinguishable renderings of the letter “A”.

The titles are often not even pronounceable, even though they might appear like that when glancing at them indirectly. The titles often consist of uppercase, but sometimes a combination of upper- and lowercase letters.

(fig. 28) Samples of titles, composed of legible letters, but without semantic meaning.

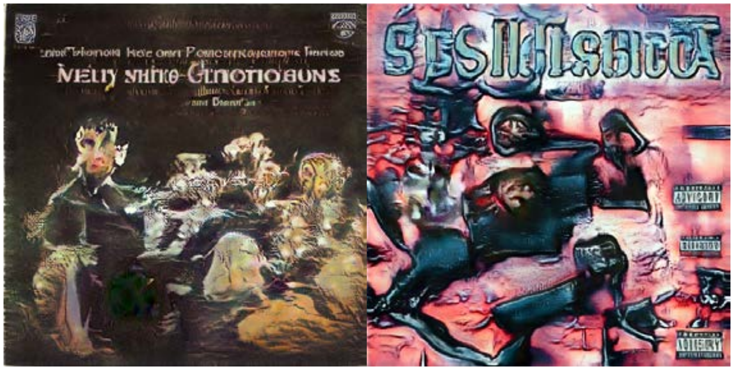
Another curious discovery was the perfect reproduction of certain graphical elements that were repeated over multiple album covers. This reproduction was separable in two different types:

(1) Graphical elements;
(2) Physical elements, e.g. CD edges.

The first type was more desirable within this research: these labels are typical for a certain genre, thus tend to appear on the cover of that genre. As they have a specific style and are repeated more frequently on album covers than other aspects, they became more clearly visible than other aspects. These marks appeared most in the Hip Hop and Classical genre.

(fig. 29) Samples of repetitive elements.

Striking was that the smaller graphical elements tended to appear up to three times on one cover, something that did not occur on the real album covers.

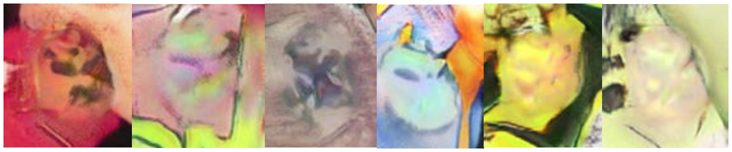


(fig. 30) Examples of covers with repetitive similar graphical elements.

The second type of reproduction were physical aspects, elements that were not part of the graphic composition of the cover but rather of the physical album cover object. Even though the scraper of the database was set to parameters that avoided scans of LPs or CDs, a part of the dataset seemed to have slipped through these restrictions. Among these were barcodes, others CD cases, and sometimes even entire CDs. This type of reproduction is normally not useful for the research, as it is not necessarily part of the graphic design of the cover. However, when special materials are used intentionally in the design, they could be attributed to the graphic design of the cover.

(fig. 31) Examples of covers with physical elements.

A third observation that became apparent was the existence of a glitchy "spot" on each generated cover. This spot is round or organically shaped and covers about a 10th in width and height of the surface. Normally the spot is less detailed than the rest of the depiction, and sometimes differs in colour from the rest of the image. The spot was already present from the beginning of the training, and did not disappear during the process. When navigating through the latent space, the spot moves location quite gradually instead of randomly, suggesting it might be part of the depiction of the images.



(fig. 32) Examples of the glitch spot in different album covers.

This spot seems to appear in the original research by NVIDIA as well, and no explanation could be found for it. Perhaps this glitch is a blind spot of the GAN, or even a focus point. It is a given that it usually appears on a background surface and not on or over type or depictions of faces or objects. In some cases, the depiction even seems to take shape around it, as if the two avoid each other.

(fig. 33) Examples of the glitch spot with depictions formed around it.

Another remarkable observation was the remaining elements of a composition on a specific point in the latent space after training on different genres. As all models are mapped to the same coarse level, the same points remain relatable. However, as shown in the example below, the composition on each latent point still deviates significantly per subgenre.

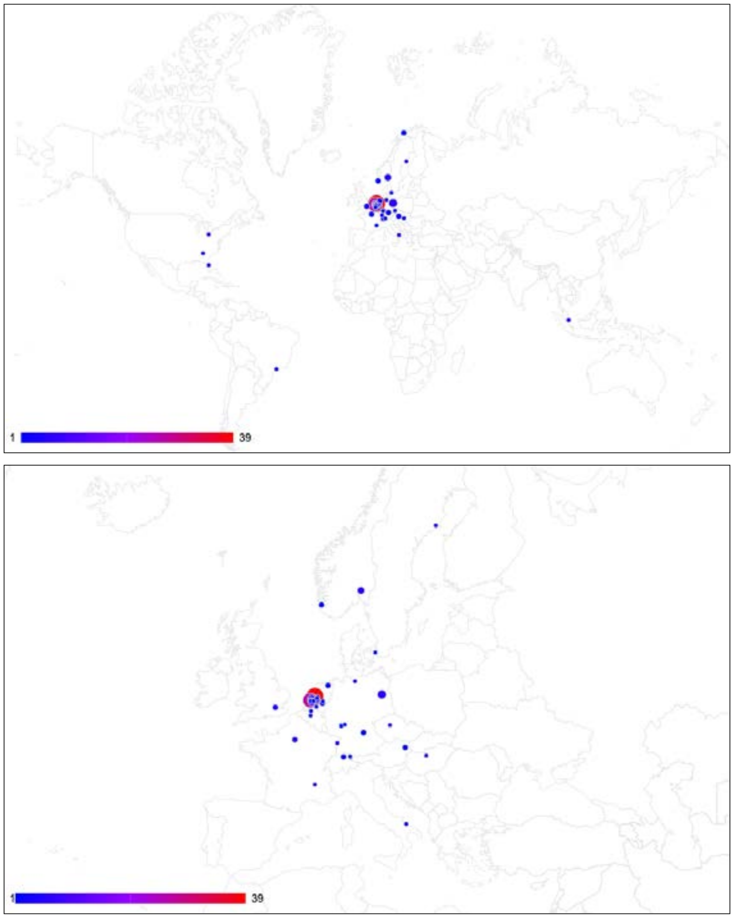
(fig. 34) Overview of the same two points in the latent space of different subgenres networks. From left to right column: Country, Heavy Metal, Europop, Hip Hop.

(4.2) PART II

The purpose of this section is to frame the test results, rather than analysing them. For the evaluation of the results see chapter (5).

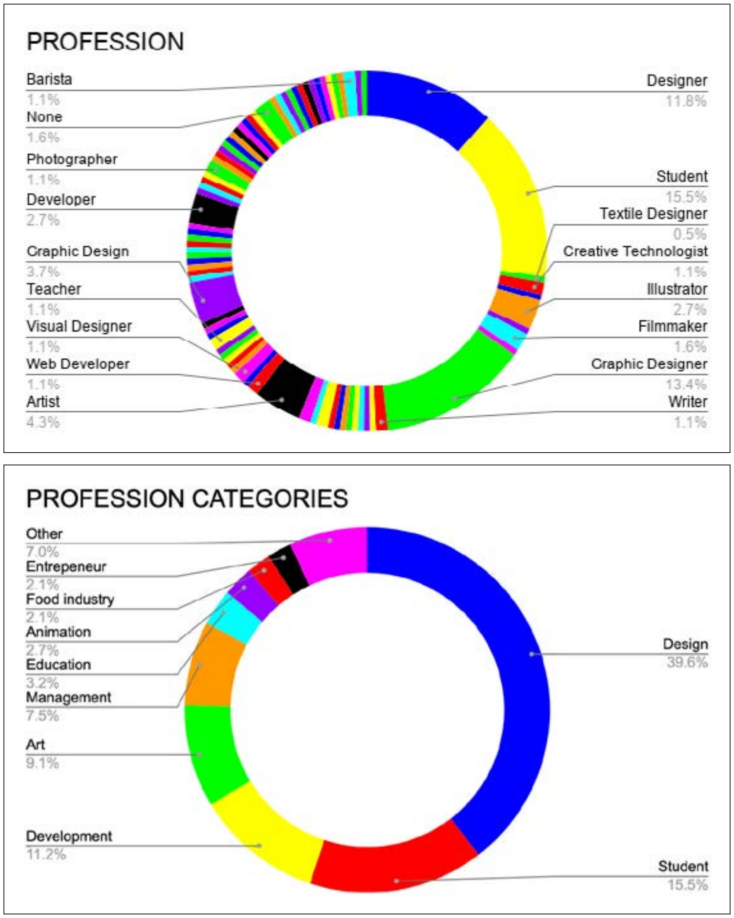
(4.2.1) DEMOGRAPHICS

In total 154 people participated in the survey. The cities and countries of residence were mostly located in the West, and generally in West Europe. The participants' age ranged from 18 to 61 years, averaging at thirty years with a median at 27 years. Roughly 60% of the participants identified as male, 33% as female and 2% as nonbinary; the rest preferred not to answer.



(fig. 35) The current city of residence of the participants. The larger the ellipse, the more participants were from that location.

The profession of the participants varied from 74 counts of "~Designer" to a range of singular entries such as "Leader Of Vera's Fanclub", "Part Of The Proletariat", "Escaped Tiger Catcher" and "Yes". It was possible to enter multiple professions and many participants did. To have a clearer insight in the division of professions, the mentioned professions were divided in categories as demonstrated in the following images.



(fig. 36) The profession and profession categories of the participants.

About 40% of the participants had at least one profession in the design field, ranging from graphic design to front-end design. 16% of the participants were students of some sort. 11% mentioned a profession in the field of development, 9% was active in the art industry, 8% in management fields and 3% in education fields as teacher or lecturer. The "other" category included some unemployment entries and a broad assortment of other professions.

(4.2.2) METADATA

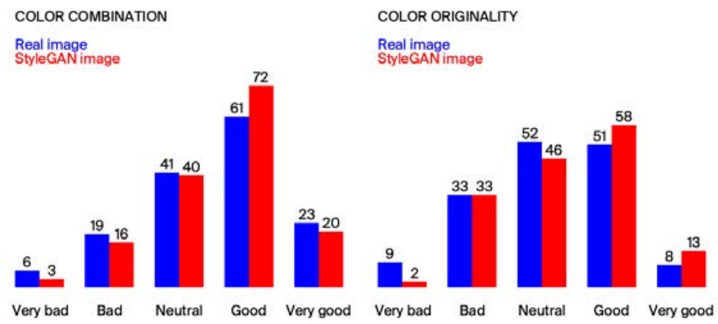
In total, the survey website had 463 unique visitors, of which 162 submitted their entries. Some of this data was not complete and thus removed, which lead to 154 valid entries. Most of the participants landed on the survey via social media: 67 via posts on Instagram, 61 via posts on Facebook, others via links in direct messaging on said platforms or mails. The average visiting duration was 340 seconds, about 5 and a half minutes. Most of the operating systems that were used to visit the site were Apple systems: 84%, followed by 11% Linux and 5% Windows systems. About 18% performed the survey on mobile devices, the rest on other devices. Irrelevant for the research, but a characteristic finding was that up to 206 bots or people downloaded the typeface used on the website illegally from the source folder.

(5) EVALUATION

In this chapter, the test results will be explained and framed. The real album cover artwork samples will be referred to as "real images" and the album cover artwork generated by the StyleGAN will be referred to as "StyleGAN images" or "generated images".

Likert Scale data do not allow using the mean as a measurement of central tendency, but rather the median.[22] As there are only five options on this particular scale, using it for statistics would not fully reveal relevant tendencies – especially not with a user group of this size. The data require a more insightful analysis than only a mean to be framed properly, as it would be helpful to gain insight on *why* participants had certain preferences. Therefore, the results are analysed in a qualitative manner: this includes general observations on the Likert Scale data as well as an elaborative analysis of written responses from the survey.

(5.1) COLOR

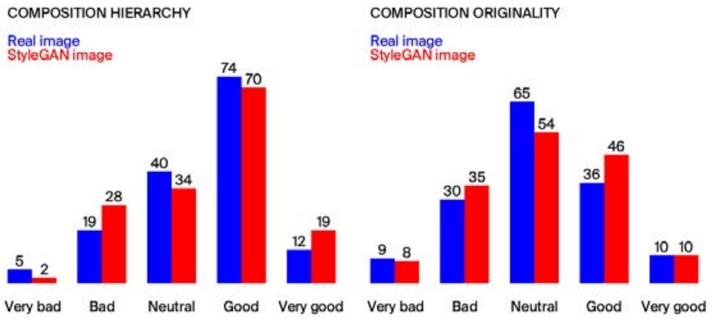


(fig. 37) The division of entries on colour combination and colour originality.

The survey entries of the generated images are distributed slightly more towards the positive side of the scale on both colour combination and originality than the real images. However, both real and generated images have received the same median value of 3/3 – "Neutral" – on colour combination and colour originality.

(5.2) COMPOSITION

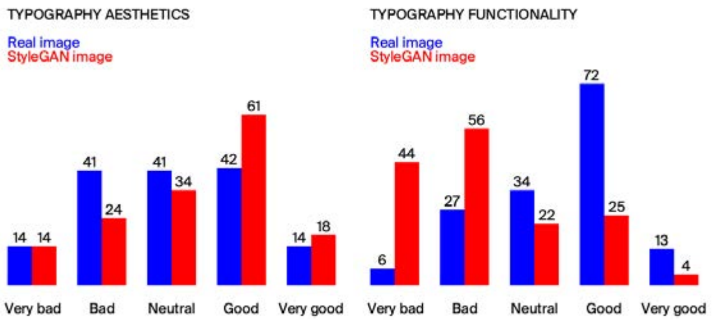
22 See st-andrews.ac.uk/media/capod/students/maths-support/Likert.pdf.



(fig. 38) The division of entries on composition hierarchy and composition originality.

The real and generated images have comparable results in terms of the hierarchy of the composition. The feedback of the real images is a bit more distributed around the fourth option of the scale, "Good", but both images received the same median (3/3). The feedback on originality of the composition is nearly similar for both images as well, but the real images have a bit more "Neutral" reviews and the feedback on the generated images has been distributed more evenly, with more "Bad" and "Good" reviews, but with the same median (3/3).

(5.3) TYPOGRAPHY



(fig. 39) The division of entries on typography aesthetics and typography functionality.

The generated images have received more positive feedback in terms of aesthetics of the typography than the real image, again with the same median (3/3), but worse in terms of functionality (3/2) This is the only Likert Scale question on which the median of the feedback is different.

(5.4) GENERAL STYLE AND DESIGN

(fig. 40) The division of entries on general style and design conclusion.

The generated covers tend to score higher on general style (3/3) and slightly higher on design than the real ones (3/3), with the same median.

(5.5) (SUB)GENRES

Overall, the real images were classified successfully 1,5 times more often than the generated images. In total, the real images were attributed to the right genre 91/153 times, out of which 51 belonged to their respective subgenre. The generated images were attributed to the right genre 37/153 times, of which 23 belonged to their respective subgenre. It appears that the images from a subgenre are easier to attribute to their respective genre correctly than an image from a genre; for the real images as well as the generated ones.

Correct attribution images to genre	Genre	Subgenre	Total	Percentage
Real images	40	51	91	54% (n = 153)
StyleGAN images	14	23	37	24% (n = 153)
All images	54	74	128	42% (n = 153 * 2)

(tab. 5) The correct attributions of images per (sub)genre of real and generated images.

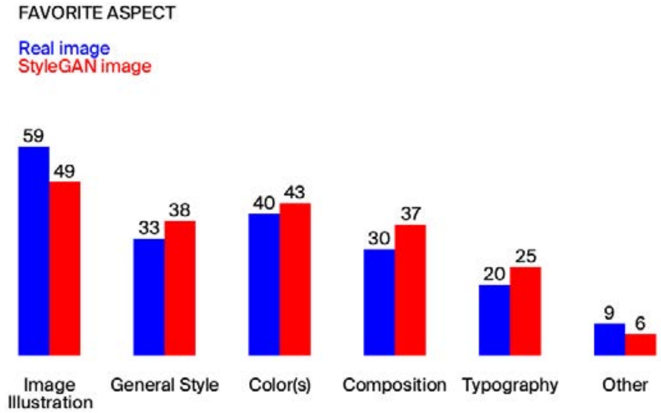
Correct attribution images to genre		Classical Romantic	Electronic Trance	F/W/C Country	Hip Hop Gangsta	Jazz Swing	Pop Europop	Rock H. Metal
Genre	*Real*	6	3	6	8	7	4	6
	StyleGAN	5	3	4	0	0	0	2
Subgenre	*Real*	8	6	7	5	3	9	13
	StyleGAN	5	3	2	1	1	0	11
Total successful assessments (n=128)		24	15	19	14	11	13	32
Total assessments (n=153 * 2)		42	38	48	42	36	52	48
Percentage success		57	39	40	33	31	25	66

(tab. 6) The correct attributions of images per (sub)genre of real and generated images, divided per genre.

From the different music genres and subgenres, especially Pop and Jazz were generally difficult to classify. Classical music and Rock music were easier to identify successfully, for both images. Remarkable was that the generated image has not been correctly classified for the genres Hip Hop, Jazz and Pop, and barely for their respective subgenres. The subgenre of Rock, Heavy Metal, was most successfully classified for both real and generated image.

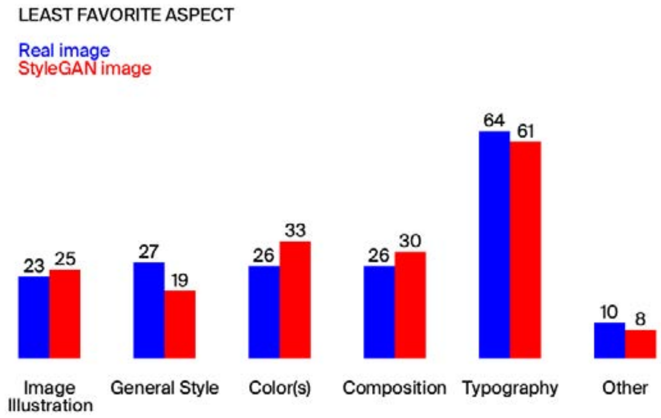
(5.6) (LEAST) FAVORITE ASPECTS

To the question what favorite aspect there was, a slight preference was given to the colors, general style and typography for the generated image. However, the illustration or depicting elements were more popular in the real images and had the largest count of favorite aspects for both the generated and real image. For this question in the survey it was possible to select multiple aspects at once. Other favorite aspects for the real images were often terms that related to the image, illustration or a design technique, such as “stars”, “red bird logo”, “overlay” and “materiality”. For the generated image, other favorite aspects were “contrasts”, “trashy”, “weird-ness”. Both favorite aspects for the real and the generated images were also answered with “none” or “nothing”, as to emphasise a negative opinion on the design.



(fig. 41) The division of favorite aspects of images.

The least favorite aspect of both real and generated images was by far the typography, almost twice as much as the second least favorite aspect. The real images occurred more frequently in terms of typography and general style, and the generated images slightly more frequently in terms of images, colors and composition. Other least favorite aspects for the real images were visual aspects such as “the edges”, “title font”, “crackling effect”. For the generated images, other least favorite aspects were for instance elements like “smudge” and “face of the person”.



(fig. 42) The division of least favorite aspects of images.

(5.7) FAVORITE COVERS

Out of the favorite cover selection, 66 (43%) real and 88 (57%) generated images were selected. This is a slight preference for the generated images, but the comments given as a reason for the selection by the participants is perhaps more insightful than this figure.

A recurring reason to select the real images was feedback on the fact that it was designed with an intention that was understandable to humans, for example:

> "I *instantly know what music genre it portrays*, yet the cover seems more original than a copy-paste fest/averaging different album covers."
> "Nice composition, *made by a human*."
> "... This just looks like a *real record* ..."
> "The composition really *looks "handmade"* and *not just randomly positioned elements* together."
> "... the fact that *you understand what is happening*!"

Secondly, the readability of the text was a recurring reason as well:

> "The cover looks right for its genre, *the typography is readable* and the illustration interesting to look at. Well composed."
> "*Readability of the text*."

Next to that, there were also some comments on the complexity or level of detail from participants that selected a real image:

> "*Nice, little details* all over like the faded-texture..."
> "*Complexity*."
> "I like *the mess* on this one. Feels like one of these big images for kids with many things to find/ discover."
> "Really enjoying the *sharp details* and colour choices..."
> "While having a *complex composition* it still manages to give off very soothing and peaceful vibes..."
> "Composition, *sharpness* and bling bling."
> "Love the *many details* in the picture!"

Another recurring reason to select the real images were aesthetic preference connected to a style that was popular or common in a certain time period:

> "... It radiates this *nice, ugly 80s vibe*."
> "It's got that *medieval vibe*."
> "It has that *nice, retro, genuine rap 90s* taste."
> "I like the *kitschy retro-quality*."

To summarize, the four most recurring arguments to select a real image over a generated one were the presence of intentional meaning, understandable typography, complexity and a clear connotation to a certain time period with a recognisable aesthetic. It is likely that these types of arguments resulted from the contrast between the real and generated images. The fact that the real images had an intentional purpose and meaning was mentioned often throughout the survey. The typography was considered more readable on real images too. Thirdly, the sharpness and details of the real images were favorable. As a GAN essentially attempts to imitate an instance of the dataset, perhaps it averaged the images and would need a different subset of the dataset to create more complex, sharp or detailed images. Finally, the aesthetic and treatment of real images may reflect more on a certain time period that is associated with the music genre than the aesthetic of the generated images.

For the generated images, a frequently recurring comment was that the image was intriguing in some way:

> "... *intriguing*, deformed face."
> "The image is very *intriguing*, looks like a microscope image and that fascinates me."
> "Bold composition and colors, *intriguing* shape of image."
> "Distorted image pattern is *interesting*."
> "*Curious* picture."

Additionally, some of the comments referred to how the image drew attention:

“*Invites to think and reflect* on the image.”
“Cover *gets my imagination going*.”
“… Portrait simply *draws my attention*.”

Another frequent comment in favor of the generated image was the surreal or mysterious quality:

“I like the *surreal* composition and typography.”
“Love the colors, atmosphere and *level of surrealism*.”
“*Ghost-like*.”
“It’s *mysterious*.”
“Looks nice but *mysterious*.”
“Interesting because it has some kind of *natural tension*, edgy..”
“… *Eerie*, in a good kind of way.”

Other participants described the weird, obscure quality of the generated images as a reason to pick them as favorite:

“*Weirdly* distorted …”
“It’s *weird* in a good way.”
“It’s *weird*.”
“*Obscure*, which is very important for a cover in my opinion!”
“This thing looks properly *obscure* and I’m into that…”
“*Strange, rotted image*. Pretty abstract but still looks like a (snake from hell) face staring at you.”

Some comments showed excitement about the general aesthetic as well as more specific aspects such as image, colors and typography.

“AI-Beyoncé looks really *cool*.”
“Nice use of colour, *cool* aesthetic.”
“*Coolest* general style.”
“I‘m *vibing* (nice colours and shapes).”

To conclude, participants usually chose generated images as favourite because of their intriguing quality. Whereas several types of descriptions were used, the eerie aesthetic appeared to be the primary reason to select the generated images as favorite.

(5.8) GENERAL REMARKS

Another insightful part of the survey was the general remarks section, where people could leave comments. The feedback in this section was rather polarised, although mostly positive.

The negative feedback was based on an incomprehension of the survey structure or the project:

“The questionnaire *seems to deal with one genre only*.”
“If this was a comparison, *it felt really random*.”

In these two cases, it was perhaps not clear enough for the participants how and why the survey was structured this way, although this was done intentionally to avoid priming them too much. The next case is the only clearly negative piece of feedback on the entire project in the survey, in which the participant indicates that they find the research superficial and that the generated covers are not functional. The first argument primarily demonstrates the participant’s excitement, but the remark about the absence of function is a critical one. This statement will be elaborated on in (6) as well as in (7).

“I don’t really understand why I am asked to compare real album covers and generated ones, feels superficial. *The generated covers do not seem made for any actual music*, just ghosts. It would probably fit some lo-fi dystopian ambient.”

The only other slightly negative remarks were the expression of a feeling of either incomprehension or curiosity:

“Not sure what just happened.”
“What is all this?”
“What just happened?”

Often, even when they had not realised that some images were generated, participants commented that functional typography was still the missing element to have a similar quality to real images:

> “I think that *mostly the text still distinguishes a GAN image from a real one*. The rest is very cool!”
> “Interesting that so many of these have *unreadable typography*. Is that a thing?”
> “Really cool project and looks promising… *Fonts training is a need*.”

Next to everything mentioned above, the greater part of the remarks were appreciative and enthusiastic:

> “Quite impressed (knowing you are working with generated images) that *a lot of them make sense and could be from a real album with no doubt*.”
> “*Nice faces*! Those are really the best.”
> “Love this project! Definitely want to check its developments!”

Next to the negative and positive remarks, there were also some suggestions on how to continue the research from people’s professional background:

> “Not sure if you are planning to retrain/evaluate your models with this data, if yes you might be interested in showing the same album covers to multiple users to measure annotator agreement.”
> (from a Machine Learning Engineer)
>
> “I could say a word from my animator perspective: I really like this project, but I think it would be even greater without the “pumping” and/or back-and-forth effects the motion has, like it was stuttering on more obvious pictures. (I’m only talking about the video [as mentioned in a lecture, ed.] not isolated images) And a global and fluid motion with some kinds of directions throughout the different compositions would be amazing!”
> (from an Animation filmmaker)

These suggestions on how to continue will be elaborated in the last section of chapter (7).

(6) CONCLUSION

In this research, album cover art generated by a StyleGAN was studied from a graphic design perspective, focusing on which graphic elements would function well as a part of a cover design and whether there was a difference in design quality between the generated and real cover art. Central to this study was the question "*can a StyleGAN be trained to design album cover art that looks as good as human-designed album covers?*"

In conclusion, we can state that the answer to this question is no, but perhaps almost, for the following reasons.

Firstly, if this question would be answered only by comparing metrics, the StyleGAN lost from human designers mainly on one aspect addressed in this research. Even though the images generated by the network had the same median as the real images on nearly all aspects—hierarchy, colors, composition, design, typography aesthetics—the samples had a lower median for the functionality of the typography than the real samples. If the question would be approached this way, the StyleGAN lost by one point.

However, the question demands a more complex answer. A more thorough approach is to compare the results of the generated and real samples by the three notions of a successful design product.

The cover artwork has to communicate: The generated samples do not belong to Kalman's definition of a good design product, namely a medium or a means of communication using "… words and images on more or less everything, more or less everywhere". Most importantly, the samples do not display functional typography yet; as one of the participants mentions: "I think that mostly the text still distinguishes a GAN image from a real one." To many participants, the functionality of the typography is really the most prominent feature that leaves room for improvement. The combination of letters on the generated samples are at this point not controllable and have no semantic meaning, as shown in the last section of the chapter Results: (4.1.4).

A comment by one of the participants was that "the generated covers do not seem made for any actual music". At this point, the generated samples are still devoid of any function or meaning, and there is nothing to be communicated in that case. Additionally, two recurring comments to elaborate on one's preference of a real sample was the legibility of the text as well as "... the fact that you understand what is happening".

Next to the lack of typographic functionality, it was harder to assign the generated samples to the correct genre, as shown in (5.5). So the imagery was lacking insight—more often than the real samples—to which music genre the cover belonged. Following Kalman's definition, the samples would be a good design product if it would be manageable to derive the music genre from the cover.

Thirdly, there was no big difference in the evaluation of the design of the generated samples and the real samples, but there was a tendency to give a more negative feedback to the generated images, as shown in (5.4). Conclusively, based on these three factors, the generated album artwork generally communicates less well than the real samples.

The cover artwork has to be appealing to its target audience and incite curiosity: As Munari argues that the artwork has to have a form coherent to its function, the samples should be appealing and spark curiosity.
The samples managed to appeal to the test group, since they were favored more often by the participants than the real samples as shown in (5.7).

According to Hirsch et al. (2012), the reason to be curious about something is different for each person and depends on their interests. Many participants mentioned the generated samples looked intriguing, mysterious, "weird in a good way", or simply "cool", therefore it seemed like they were often appealing to the test group. Other general elements that might add to the appeal of a cover were the aesthetics of the typography and the colour combination, both for which the generated samples tended to be rated higher.

On top of that, other participants mentioned the "... portrait simply draws my attention", the generated sample got their "imagination going", or that it was inviting to "think and reflect on the image", showing that curiosity may have been triggered in some cases.
There were no similar remarks of participants being intrigued by the real samples. The closest notion to intrigue might be a recurring interest in the cover's complexity as multiple participants enjoyed "... the sharp details", but nothing considerable was mentioned about the real samples on the participants' curiosity.

To conclude, the generated album artwork succeeded at appealing and provoking curiosity, and seemed to have superseded the real images in terms of the latter.

The cover artwork has to be recognisable: As Munari suggests, the simple use of an unusual colour, a different form or composition are enough to make a design product stand out and thus fulfill its function. Both the originality of the colors and the composition gravitated towards the more positive assessment in the case of the generated samples. On top of that, participants considered the covers to be unusual, eerie or weird. The response to the general style was also inclined to be more positive for the generated samples than the real ones, indicating that the generated album cover is more recognisable. The real samples were also mentioned to be part of a popular or common aesthetic linked to a certain time period, e.g. "... it radiates this nice, ugly 80s vibe".
The generated samples received similar comments cross-genre about a certain eerie, mysterious aesthetic. From this we can conclude that sometimes the real samples belonged to an aesthetic genre linked to time that is already recognisable, when the generated samples created a new aesthetic genre altogether.

In conclusion, the album cover artwork is not yet as good a design product as the real cover artwork, but has a potential to add in the fields of appeal, drawing attention and recognisability.

(7) DISCUSSION

In this chapter, the research is contextualised, multiple possible improvements are discussed and possible future explorations are suggested. The feedback on the research itself is sorted by subject, and the last paragraph suggests possible ways to continue.

The conclusion of this research shows that even a current state-of-the-art technique such as StyleGAN is not yet able to perform applied design tasks as well as a human designer, especially in terms of typography. Perhaps if the StyleGAN would somehow allow for more control by for instance textual input to retrieve a certain instance from the latent space of the model, it could greatly improve its potential for applied design tools. Whereas the scope of the model is largely defined by the dataset now, more handles in navigating the latent space would be beneficial.

The rather automated GAN training process and the subjective, human design process seem to be still quite far apart. However, it is clear from the method that the GAN algorithm follows along the lines of the generative design. With even less control over the specific outcome, the role of the designer gradually shifts even more from creation towards curation. The interpolation script offered a handle to curate from a big selection of options, rather than to spend that same time creating something from scratch.

However, the GAN also allows the generation of novel imagery, meaning that it could be an addition to the design practice instead of a replacement. As GANs can create imagery that morphs between multiple “logical” points in latent space – points with a recognisable depiction –, a rather unique technique so far that creates a novel aesthetic. As the aesthetic of this technique might be used more often in the near future, it will become more recognisable and understood. Therefore, the novelty will likely eventually wear out and only time can tell whether the technique is interesting enough to keep being used besides its novelty.

(7.1) DATASET

Perhaps the results would have been more comparable if each sub-genre and each genre had exactly the same amount of images as an input. Now the input per subgenre varied from 1.761 to 9.672 images and the genre between 8.359 and 28.700 images.

The dataset was scraped from open source platform Discogs. Possibly another source that was created in a more restrictive way would have been beneficial for the training results. For instance, the Spotify album cover artwork dataset has a fixed size for the larger images of the covers, where Discogs accepts different sizes.

Another possible improvement could have been the resizing script. Now, images were either resized if they did not match the target size of 512 x 512 pixels but were larger or slightly smaller, or deleted if the images were too small or the ratio was too contrasting from the target 1:1 ratio. Another solution could have been to place the images central on a black background if they were too small or a different ratio, instead of deleting them from the dataset. This could possibly have improved the results for some of the genres as their training dataset would have been bigger including the removed images.

(7.2) TRAINING PROCESS

If more computing power had been available, it would have been more feasible to correct mistakes. It also would have been interesting to see if there was a difference between metric control over the training duration instead of intuitive control.

The warm weather affected the training, sometimes causing the training process to be interrupted during peak temperatures. The training process would have been more effective during a colder weather or in a temperature-controlled surrounding.

Perhaps the training method could have been approached differently. In the current research the StyleGAN was trained on the entire database first, and subsequently on different genres and subgenres. Starting the training with the genres might have affected the final results in a positive manner. It is possible that continuing training from a pre-training model is not the ideal to recreate the most efficient model per genre and subgenre as the coarse level of the images is defined by a different dataset and the input is mapped differently from the beginning of the training.

(7.3) SURVEY

The Likert scale may have been a subject to distortion because of the following reasons:[23]

(1) *Central tendency bias*: avoidance of extreme responses. Most questions were answered with "Neutral" as median. However, as the user group consisted largely of designers whose everyday work includes judging and selecting, this bias type might not have had a large effect.

(2) *Acquiescence bias*: agreeing with statements the way they are presented, being inclined to give positive confirmation rather than negative feedback.

(3) *Social desirability bias*: attempt by the participants to be favorable, i.e. participants that knew about the research process might have been inclined to favor the generated covers to be portrayed in a more favorable light.

Perhaps the Likert scale could have been replaced by a grading system using a numeric value, which would have revealed differences between the real and generated images more easily.

The most challenging aspect of the survey was the difference in significance between the real and generated images. The real images have at some point been published and used by musicians, whereas the generated images have not. It would have been good to find a way to simulate the significance of the generated covers.

23 For more, see: st-andrews.ac.uk/media/capod/students/mathssupport/Likert.pdf.

Finally, even though the user group was large enough to distill preferences and tendency, a larger user group would have given more reliable results.

(7.4) RESULTS

It appears that it is easier to attribute the images from a subgenre to their respective genre correctly than an image from a genre; for both real images as well as the generated ones. A possible cause for this difference could be that a subgenre is more defining for the aesthetics than a genre, as a subgenre belongs to a genre but not the other way around.

As commented by one of the participants, "The generated covers do not seem made for any actual music ...". Intent is an important aspect of a design product, as presented by Munari and Kalman as "something that needs to communicate". During this research there was no significance attached to the generated samples, and perhaps it would have been more fair to either find a method to create meaning for the generated samples or compare them with real ones that are somehow stripped from their meaning. An approach to the latter could for instance have been the use of fragments of the covers that contain similar design elements, e.g. a fragment for each sample with only the typography or a background image.

It could have been an option as well to computationally scramble the typography on the real images in order to make them more similar to the generated ones. In this case, the participants could have been told this was done to disguise the title and band name, as to prime them to treat all the images more equally.

(7.5) FUTURE RESEARCH

The typography that is rendered by the StyleGAN is not legible, yet has a positively rated aesthetic. Further research could be done on which methods could allow control of the words displayed on the cover, and consequently how to generate images with legible, significant words. One method that could potentially be used

for this is a division of the training on different letters per (sub)genre using the StyleGAN, or maybe using an auxiliary classifier GAN (AC-GAN) to train on the covers tagged with metadata.

It could be interesting to test the generated samples on incitement of curiosity with a more specific test group. The current research was primarily targeted on participants with a creative or critical profession, that allowed them to judge aesthetics, but it could be interesting to divide the test group even further based on their music taste, e.g. whether a generated Rock cover works more appealing for a Rock fan. This would appeal to the f*orm follows function* statement, e.g. whether a Rock cover appeals to its target audience.

One participant mentioned that it might be interesting to show the same covers to multiple participants to measure annotator agreement. An experiment similar to this was performed with the real covers during the first user test, but not in a large enough quantity. It might be interesting to continue this to be able to see how the feedback varies for one sample. Retraining the models based on the survey data might be a way to continue, although more data might be needed for the models to improve properly.

The script written to navigate the latent space of a model that can be used to render animations might be improved and explored as well. Navigating through the latent space of a generative adversarial network is rather unique in terms of animation and might have potential to develop into a successful animation technique. The type of motion could be improved, as one participant who is an animator mentioned, to a less "pumping" effect. It is unclear which animation is referred to, as work-in-progress animations made by earlier versions of the script with lower motive quality have been demonstrated to audiences as well. However, more fluid motion than the current version of the script could be realised either way. From a development perspective it might be interesting to work with a curved interpolation instead of a linear one. On top of that, it might be interesting to select points with the same distance from each other instead of random ones in the latent space.

If the generated covers can never be used functionally, the network could work as generator for placeholder or template covers on music sharing sites, or as a means to discover stereotypes in certain music genres.

The use of GANs establishes a visual vocabulary that can be employed for graphic design as well as many other visual domains, introducing an eerie and mysterious aesthetic that has several potential applications. For human-machine design collaboration in general, this research shows that the StyleGAN is a powerful helper for image generation that presents a unique realm of experimentation and exploration. Building upon generative image-making practices, this human-network collaboration process introduces a form of collaboration where the machine has even slightly more agency in the creation process than before.

(8) BIBLIOGRAPHY

Adajian, T. (2018). The Definition of Art. *Stanford Encyclopedia of Philosophy* (Fall 2018 Edition). Zalta, E. (Ed.). Retrieved October 3, 2019, from plato.stanford.edu/entries/art-definition/.

Adams, D. (1979). *The Hitchhiker's Guide to the Galaxy*. London, UK: Pan Books.

Adorno, T. (1970). *Aesthetic Theory*. Frankfurt Am Main, Germany: Suhrkamp Verlag.

Arida, S. (2004). *Contextualizing Generative Design*. (Doctoral dissertation, Massachusetts Institute of Technology).

Bachl, M., & Ferreira, D. (2019). City-GAN: Learning architectural styles using a custom Conditional GAN architecture. arXiv preprint arXiv: 1907.05280. Retrieved July 28, 2019.

Ballardini, R., He, K., Roos, T. (2018). AI-Generated Content: Authorship and Inventorship in the Age of Artificial Intelligence. Retrieved October 12, 2019, from cs.helsinki.fi/u/ttonteri/pub/aicontent2018.pdf.

Barnard, M. (2013). *Graphic Design as Communication*. Abingdon, UK: Routledge.

Boden, M. (2010). *Creativity and Art: Three Roads to Surprise*. Oxford, UK: Oxford University Press.

Bohnacker, H., Groß, B., Laub, J. (2012). *Generative Design: Visualize, Program, and Create with Processing*. New York: Princeton Architectural Press.

Bourriaud, N. (1998). *Relational Aesthetics*. Pleasance, S., Woods, F, Copeland, M. (trans.). Dijon, France: Les Presses Du Réel.

Brownlee, J. (2019). How to Explore the GAN Latent Space When Generating Faces. *Machine Learning Mastery*. Retrieved from machinelearningmastery.com/how-to-interpolate-and-perform-vector-arithmetic-with-faces-using-a-generative-adversarial-network/ on August 22, 2019.

Burke, E. (2015). *A Philosophical Enquiry into the Origin of Our Ideas of the Sublime and Beautiful*. Oxford, UK: Oxford University Press. (Original work published in 1757).

Burckhardt, L. (2017). *Design is Invisible*. Berlin: De Gruyter. Blumenthal, S. (Ed.). (Original work published in 1980).

Cohn, G. (2018, October 25). AI Art at Christie's Sells for $432,500. *NY Times*. Retrieved from nytimes.com/2018/10/25/arts/design/ai-art-sold-christies.html on September 17, 2019.

Christie's. (November 2018). Is artificial intelligence set to become art's next medium? *Christie's*. Retrieved from christies.com/features/A-collaboration-between-two-artists-one-human-one-a-machine-9332-1.aspx on September 17, 2019.

Davis, D. (2013). A History of Parametric. Retrieved October 12, 2019, from danieldavis.com/a-history-of-parametric/.

Deltorn, J., Macrez. F. (2018). Authorship in the Age of Machine Learning and Artificial Intelligence. SSRN: 3261329. Retrieved October 12, 2019.

Fry, T. (2008). *Design Futuring*. Oxford, UK: Berg Publishers.

Galanter, P. (2016). Generative art theory. In C. Paul (Ed.), *A Companion to Digital Art*. pp. 146 – 180. Hoboken: Wiley.

Girard, D., (2014). "How QuarkXPress became a mere afterthought in publishing". *Ars Technica*. Retrieved September 9, 2019 from arstechnica.com/information-technology/2014/01/quarkxpress-the-demise-of-a-design-desk-darling/.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., et al. (2014). Generative adversarial nets. In *Advances in neural information processing systems*, 28. pp. 2672 – 2680.

Goodfellow, I. (2016). NIPS 2016. Tutorial: Generative Adversarial Networks. arXiv: 1701.00160v4. Retrieved June 24, 2019.

Hepburn, A., McConville, R., Santos-Rodriguez, R. (2017). Album Cover Generation from Genre Tags. Researchgate: 318987126. Retrieved on July 23, 2019.

Hill, D. (2014). *Dark Matter and Trojan Horses*. Moscow, Russia: Strelka Press.

Hirsch, J., Kang, S., Bodenhausen G. (2012). Personalized persuasion: tailoring persuasive appeals to recipients' personality traits. DOI:10.1177 / 0956797611436349. Retrieved September 29, 2019.

Jin, Y. et al. (2017). Towards the Automatic Anime Characters Creation with Generative Adversarial Networks. arXiv: 1708.05509. Retrieved August 12, 2019.

Kant, I. (1987). *Critique of Judgment*. Pluhar, W. (trans.). Cambridge, USA: Hackett Publishing Company. (Original work published 1790).

Karras, T., Laine, S., Aila, T. (2019). A Style-Based Generator Architecture for Generative Adversarial Networks. arXiv:1812.04948. Retrieved on September 16, 2019.

Karras, T., Aila, T., Laine, S., Lehtinen, J. (2018). Progressive Growing of GANs for Improved Quality, Stability, and Variation. arXiv:1710.10196. Retrieved September 16, 2019.

Lībeks, J., Turnbull, D. (2011). You Can Judge an Artist by an Album Cover: Using Images for Music Annotation. DOI: 10.1109/MMUL.2011.1. Retrieved June 12, 2019.

Loos, A. (1908). *Ornament and Crime*. Mitchell, M. (trans.). London, UK: Ariadne Press.

Lupton, E., Lipps, A. (2018). *The Senses: Design Beyond Vision*. New York, USA: Princeton Architectural Press.

Madsen, R. (2016). *A History Of Design Systems*. Retrieved October 12, 2019 from printingcode.runemadsen.com/lecture-intro/.

Maeda, J. (2000). *Design By Numbers*. Boston, USA: The MIT Press.

Manovich, L. (2001). *The Language of New Media*. Cambridge, USA: The MIT Press.

McCormack, J., Gifford, T., & Hutchings, P. (2019). Autonomy, Authenticity, Authorship and Intention in Computer Generated Art. In *International Conference on Computational Intelligence in Music, Sound, Art and Design* (Part of EvoStar). 8. pp. 35–50. Cham, Switzerland: Springer.

McCormack, J., Innocent, T., Dorin, A. (2004). *Generative Design: A Paradigm for Design Research*. In Redmond, J. et al. (Eds.) *Proceedings of Futureground*. 534. Melbourne, Australia: Design Research Society.

McFee, G. (2011). *Artistic Judgment: A Framework for Philosophical Aesthetics*. London, UK: Springer.

Moline, K. (2006). Authorship, Entrepreneurialism and Experimental design. Retrieved August 13, 2019 from handle.unsw.edu.au/1959.4/10318.

Munari, B. (2008). *Design as Art*. Creagh, P (trans.). London, UK: Penguin Modern Classics. (Original work published in 1966).

Novitz, D. (2001). *The Boundaries of Art*. Christchurch, New Zealand: Cybereditions Corporation.

Osman, I. Cover Story: A Short History of the Aesthetics of Excess in Hip Hop. In *zweikommasieben*. 16. pp. 105–113. Courtesy of Raphael Schoen.

Parikka, J. (2008). Leonardo book review: The art of artificial evolution: a handbook on evolutionary art and music. Retrieved August 23, 2019, from leonardo.info/reviews/nov2008/parikka_art.html.

Radford, A., Metz, L., Chintala, S. (2015). Unsupervised Representation Learning With Deep Convolutional Generative Adversarial Networks. In Conference Paper *ICLR*, pp. 1–16.

Rani, H. (2019). Style-based GANs—Generating and Tuning Realistic Artificial Faces. On LyrnAI. Retrieved July 24, 2019.

Reas, C., McWilliams, C., LUST. (2010). *FORM+CODE: In Design, Art and Architecture*. New York, USA: Princeton Architectural Press.

Soddu, C., Colabella, E. (2014). *Proceedings of XVII Generative Art Conference*. Rome, Italy: Domus Argenia Publisher.

Soddu, C. (2002). New Naturality: A Generative Approach to Art and Design. In *Leonardo 35:3 (T35)*, pp. 291–294.

St. Laurent, A. (2008). *Understanding Open Source and Free Software Licensing*. O'Reilly Media. pp. 4–5. Retrieved September 17, 2019 from books.google.nl/books?id=04jG7TTLujoC&pg=PA4&redir_esc=y#v=onepage&q&f=false.

Wittgenstein, L. (1953). *Philosophical Investigations*, Anscombe, G. and Rhees, R. (eds.), Anscombe G. (trans.). Oxford, UK: Blackwell.

Woodbury, R. (2010). *Elements of Parametric Design*. Abingdon, UK: Routledge.

Zachariou, R. (2018, December 16). Machine Learning Art: An Interview With Memo Akten. *Artnome*. Retrieved October 20, 2019, from artnome.com/news/2018/12/13/.machine-learning-art-an-interview-with-memo-akten.

(9) APPENDIX

(9.1) SCRAPER

Usage: Save as `scraper.js`. In your Terminal, `$ cd path/to/code` to the folder of this code. Type `npm install` to install dependencies and run the code by typing `node scraper.js` in the Terminal. Edit the genre and style `lets` to change the search query. More info for specified search is on the developer webpage of Discogs.

```
// Settings for the genre and subgenre
let search = ‘’;
let genre = ‘Pop’;
let style = ‘Vocal’;
let link = ‘<api.discogs.com/database/search?q=>’ +
search + ‘&genre=’ + genre + ‘&style=’+ style+’&per_
page=100&key=[your-user-key]&user-agent=[your-dis-
cogs-project]’
let dir = ‘../Covers/’ + genre + ‘/’ + style + ‘/’ ;
let info;

const request = require(‘request’);
const fs = require(‘fs’);

// Options are necessary to access the URL
let options = {
 url: link,
 headers: {
 ‘User-Agent’: ‘request’
 }
};

function callback(error, response, body) {

 if (!error && response.statusCode == 200) {
 info = JSON.parse(body);

  // Loop through the available pages (content is
divided over pages)
  let currentPage = info.pagination.page;
  let lastPage = info.pagination.pages;
  let nextPageUrl = info.pagination.urls.next;

  if( currentPage < lastPage ) options.url = nextPa-
geUrl;
  else options.url = false;

  let data = JSON.stringify(info, null, 2);
  let filename = dir + genre + “-” + style + “-page-
”+ currentPage + “.json”;

  // Write data to .json file
  fs.writeFile(filename, data,(err) => {
   if (err) throw err;
   console.log(‘Data written to file ‘ + filename);
  downloadImages();
  });
 }
}
```

```
// Downloads images and names them with their ID
number
let downloadImages = ()=> {
 if( info.results.length ) {
  let imageUrl = info.results[0].cover_image;
  let id = info.results[0].id;
  let title = info.results[0].title;

  download(imageUrl, dir+id+'.jpg', function(){
   console.log(title, imageUrl, info.results.length
);
   info.results.shift();
   downloadImages();
  });
 } else {
  // callback...
  makeReq();
 }
}

// Used to download files via API
let download = function(uri, filename, callback){
 let url = {
  url: uri,
  headers: {
  'User-Agent': 'request'
  }
 };
 request.head(url, function(err, res, body){
  if(err)console.log(err)
  request(url).pipe(fs.createWriteStream(filename)).
on('close', callback);
 });
};

// the request
function makeReq() {
 if(options.url)request(options, callback);
 else console.log("URL is false, the scraping is
probably finished!")
}

// This starts the whole shebang
makeReq();
```


(9.2) IMAGE RESIZE SCRIPT

Usage: Save as `resizer.js`. In your Terminal, `$ cd path/to/code` to the folder of this code. Edit the folder directory to the path to your images and feel free to change the `let` for target size, threshold for small covers and maximum ratio. More info for specified search is on the developer webpage of Discogs. Type `npm install` to install dependencies and run the code by typing `node resizer.js` in the Terminal.

```
// Node requirements
let Jimp = require('jimp');
let fs = require('fs');
const isImage = require('is-image');
let sizeOf = require('image-size');

// Folder directory, change this for resizing
let dir = "../Covers/Classical/Modern/";

// Size has to be a factor of 2
let size = 512;

// Threshold of small covers
let tooSmall = 256;

// Maximum difference w/h
let ratio = 1.25;

// Set a timeout in case buffer errors
let useTimeout = false;
let walkPath = dir;

let walk = function (dir, done) {
 fs.readdir(dir, function (error, list) {
  if (error) {
   return done(error);
  }

  let i = 0;

  ( function next () {
   let file = list[i++];

   if (!file) {
    return done(null);
   }

   file = dir + '/' + file;

   fs.stat(file, function (error, stat) {
    if (stat && stat.isDirectory()) {
     walk(file, function (error) {
      next();
     });
    } else {
     let dimensions = sizeOf(file);
     let w = dimensions.width;
     let h = dimensions.height;
```

```
      // If the file is not yet the right size
      if( isImage(file) && w != size || h != size ) {
        // Read file
        Jimp.read(file, (err, img) => {
        if (err) {
         console.log( err );
         next();
        };

        img
         .resize(size, size) // resize
         .quality(100) // set JPEG quality
         .write(file); // save
         console.log( file + " resized and saved! " +
"w: " + w + "h: " + h )
        });

      } else {
       console.log('Already right size! ' + file + "w:
" + w + "h: " + h );
      }

      if( useTimeout ) {
       setTimeout( function() {
        next();
       }, 50 );
      } else {
       next();
      }
     }
    });
   })();
  });
};
 process.argv.forEach(function (val, index, array) {
 if (val.indexOf('source') !== -1) {
  walkPath = val.split('=')[1];
 }
});
 console.log('processing...');

walk(walkPath, function(error) {
 if (error) {
  throw error;
 } else {
  console.log('finished!');
 }
});
```


(9.3) INTERPOLATION SCRIPT

Usage: Save as `interpolation.py`. The script is an extension to the StyleGAN, so it needs to be saved in the root folder of the code. In your Terminal, `$ cd path/to/code` to the StyleGAN folder. Make sure python version 3.6 is installed, then install dependencies with `pip`:

- Pickle
- Numpy
- Tensorflow
- Pillow
- dnnlib

Run the code by typing `python interpolation.py` in the Terminal. Edit the steps and endpoints to control the speed of the navigation.

```
"""Interpolation script."""

import os
import pickle
import numpy as np
import tensorflow as tf
import PIL.Image
import dnnlib
import dnnlib.tflib as tflib
from dnnlib.tflib.autosummary import autosummary

from numpy.random import randn
from numpy import linspace

import config
import train
from training import dataset
from training import misc
from metrics import metric_base
from matplotlib import pyplot

# Load my network
# Options for tflib.init_tf().
tf_config      = {}

# Run ID or network pkl to resume training from.
resume_run_id        = '00034'

# Snapshot index to resume training from.
resume_snapshot      = 19800.0

# Number of steps between endpoints
steps   = 30

# Number of points
num_endpoints  = 10
```

```
# Main script
def main():

 # Initialize tensorflow
 tflib.init_tf()

 # Load network
 network_pkl = misc.locate_network_pkl(resume_run_
id, resume_snapshot)
 print('Loading networks from "%s"...' % network_
pkl)
 G, D, Gs = misc.load_pkl(network_pkl)
 progress = 1

 # Generate points in latent space as input for the
generator
 def generate_latent_points(latent_dim, n_samples,
n_classes=10):
 # generate points in the latent space
 x_input = randn(latent_dim * n_samples)
 # reshape into a batch of inputs for the network
 z_input = x_input.reshape(n_samples, latent_dim)
 return z_input

 # Uniform interpolation between two points in la-
tent space
 def interpolate_points(p1, p2, n_steps=steps):
 # interpolate ratios between the points
 ratios = linspace(0, 1, num=n_steps)
 # linear interpolate vectors
 vectors = list()
 for ratio in ratios:
  v = (1.0 - ratio) * p1 + ratio * p2
  vectors.append(v)
 return np.asarray(vectors)


 # Keep track of where in the loop
 index_endpoints  = 0

 # Keep track of the total index for naming reasons
 total_increment  = 0

 # Load the latents vector with random points to
start
 rnd = np.random.RandomState(progress)
 latents = rnd.randn(1, Gs.input_shape[1] )

 # Load the first interpolation vectors
 pts = generate_latent_points(512, num_endpoints)

 # Generate images for all the steps between the
points
 while index_endpoints < num_endpoints:

 # Loop through the points
 if index_endpoints < num_endpoints - 1:

  # Go from the current point to the next
  print("from point " + str(index_endpoints) + " to
point " + str(index_endpoints+1))
  interpolated = interpolate_points(pts[index_end-
points], pts[index_endpoints+1])
```



```
 else:

  # At the end point, interpolate back to point 1 to
form loop
  print("from point " + str(index_endpoints) + " to
point " + str(0) )
  interpolated = interpolate_points(pts[num_end-
points-1], pts[0])

 # Print all the steps between the two points
 print("saving new steps")
 for i in range(steps):

  # Add interpolated points to the latent array
  for n in range(512):
  latents[0][n] = interpolated[i][n]

  # Generate image
  fmt = dict(func=tflib.convert_images_to_uint8,
nchw_to_nhwc=True)
  images = Gs.run(latents, None, truncation_psi=0.5,
randomize_noise=False, output_transform=fmt)

  # Save image
  os.makedirs(config.result_dir, exist_ok=True)
  png_filename = os.path.join(config.result_dir+"/in-
t10x30/", 'latents-'+ str(total_increment) + '-' +
str(index_endpoints) + '-' + str(i) +'.png')
  PIL.Image.fromarray(images[0], 'RGB').save(png_
filename)

  total_increment+=1

  # Increase the point range
  if i == steps-1:
  index_endpoints+=1


if __name__ == "__main__":
 main()
```

(9.4) SURVEY

(1) BASIC INFORMATION

(1) Email*
(text field)
Enter your email address. Only used to avoid spam entries.

(2) Age
(integer)
Enter your current age.

(3) City
(text field)
Enter your city of residence.

(4) Country
(text field)
Enter your country of residence.

(5) Profession(s)
(text field)
Indicate your profession(s).

(6) Gender
(radio)
If you feel comfortable with it, please select your gender.

(2/3) REAL ALBUM COVER / GENERATED ALBUM COVER

(7) Genre
(radio)
Which genre do you think this cover is?

COLOR

(8) Combination
(radio)
I think the combination of colors on this cover is…

(9) Originality (radio)
I think the originality of colour combinations on this cover is…

TYPOGRAPHY

(10) Aesthetics
(radio)
I think the aesthetics/visual appearance of the typography on this cover is…

(11) Functionality (radio)
I think the functionality of the typography on this cover is…

COMPOSITION

(12) Hierarchy
(radio)
I think the hierarchy in the composition on this cover is…

(13) Originality
(radio)
I think the originality of the composition on this cover is…

ASPECT(S)

(14) Favorite
(select)
Which aspects do you like most about this album art?

(15) Least favorite
(select)
Which aspects do you like least about this album art?

STYLE

(16) General style
(radio)
I think the general style of this cover is…

(17) Design
(radio)
Altogether, I think the design of this cover is…

(18) Remarks
(text)
You can add any additional remarks you have (not required).

(19) Favorite
(select)
Out of the following set of 6, select the album artwork you like the best by clicking on it.

(20) Reason
(text)
Why is this album artwork your favorite?

(21) Remarks
(text)
Add any remarks you might have about this survey in general.

(10) WEEK LOG

07/06

- ☐ Tried a couple of thesis log environments.
- ☐ Set up this cool thesis environment, in Notion.
- ☐ Set up a planning.

11/06

- ☐ Built a scraper with Node.js based on tutorial: github.com/IonicaBizau/scrape-it.
- ☐ Made a custom script that runs through a search query and downloads up to 100 images per loop, due to pagination of Discogs search.
- ☐ First search was done using genre “Rock”, 4.321.703 entries, and style “Heavy Metal”, 191.438 entries.

REMARKS

- ☐ Apple API was not really compatible for this research. Discogs worked far better!
- ☐ Using Node.js `request` and `fs` to run through the search.
- ☐ Discogs has only 15 genres, divided in ‘styles’.
- ☐ The scraper timed out after 9,883 entries for some reason, it seems like the API cannot handle more than 10.000 downloads.

ISSUES

- ☐ A lot of covers have type. Wondering whether this type can be reproduced by a GAN?
- ☐ Might need to add a filter for content: there are many racist or offensive covers.

12/06

- ☐ Tried making a script that recognises type and filters it.
- ☐ Using OCR after editing with Imagemagick. Imagemagick is a PHP library to alter or read image files in batch.
- ☐ Accepting that text recognition is not the solution ☹.

REMARKS

- ☐ Tried to make an text-on-image recognition script in multiple ways.
- ☐ Tried this script, github.com/erusev/parsedown, but it wouldn't work.
- ☐ Discogs JSON data looks like this:


```
{
"style": [
  "Alternative Rock"
],
"master_id": 311432,
"thumb": "img.discogs.com/r6a4AfWVb-v119LCN-
NyQNC4DIsA=/fit-in/150x150/filters:strip_ic-
c():format(jpeg):mode_rgb():quality(40)/
discogs-images/R-821572-1348974449-1675.jpeg.
jpg",
"format": [
  "Vinyl",
  "7\""
],
"country": "UK",
"barcode": [],
"uri": "/Iggy-Pop-The-Passenger-Nightclubbing/
master/311432",
"master_url": "api.discogs.com/mas-
ters/311432",
"label": [
  "Golden Grooves",
  "RCA",
  "RCA Corporation",
  "RCA Ltd.",
  "RCA Limited",
  "RCA Limited",
  "Ricky Gardiner Songs",
  "Virgin Music (Publishers) Ltd.",
  "Bewlay Bros. Music",
  "Fleur Music"
],
"cover_image": "img.discogs.com/dyA-
2BF-l7P3HlQK9_7jKPeQOwu8=/fit-in/481x480/
filters:strip_icc():format(jpeg):-
mode_rgb():quality(90)/discogs-imag-
es/R-821572-1348974449-1675.jpeg.jpg",
"catno": "GOLD 549",
"community": {
  "have": 333,
  "want": 467
},
"year": "1982",
"genre": [
  "Rock"
],
"title": "Iggy Pop - The Passenger / Night-
clubbing",
"resource_url": "api.discogs.com/mas-
ters/311432",
"type": "master",
"id": 311432
},
```

17/06

- ☐ Made a script to edit the dimensions of the database.
- ☐ Scraped around 100.000 covers and metadata in different categories.
- ☐ Did a first test round!

REMARKS

- ☐ All images are 512 x 512 px.
- ☐ All images are resized and saved in batch.

ISSUES

- ☐ There were a lot of corrupted images, that I could filter out by ordering the files on size.
- ☐ Will have to build a loop to remove them from or replace the file to a functioning one.

24/06

- ☐ Talked to big audience about being done with thesis soon hahahaha ☹
- ☐ Also there was a lot of people giving their mail addresses for future help.
- ☐ Did a test with a GAN on the current dataset using pytorch.

26/06

REMARKS

- ☐ Collegue of T. has a super strong computer at LIACS.
- ☐ Remove LP covers would improve data drastically.
- ☐ GAN data input hacks: github.com/soumith/gan-hacks.
- ☐ Good article with tips and tricks on building your own GAN medium.com/ai-society/gans-from-scratch-1-a-deep-introduction-with-code-in-pytorch-and-tensorflow-cb03cdcdba0f.
- ☐ Made a start at implementing a GAN on the covers but the progress is slow.
- ☐ For the #WIP meeting where I can test out my survey, it might be good to test what "good" existing album covers are, and why. Initially I wanted to already perform a first version of comparing real and generated covers, but the generation process is slow and that might have been a *bit* too ambitious.

USER TEST 1

Judging 200 – 500 covers on 'goodness'. Of those 500, I have a set of 10 – 15 covers judged per user session, multiple sessions per individual possible. 25 individual users would make the test proper empirical.

The test consists of multiple parts. Per user, discipline, gender and age. Per image, the following things have to be judged:

- ☐ Level of 'goodness' (1 choice)
  1. Very bad
  2. Bad
  3. Medium
  4. Good
  5. Very good
- ☐ Why this level of 'goodness' (multiple choice)
  - Colors
  - Type
  - Composition/layout
  - Style
  - Other, namely ...
- ☐ To which genre/style does it belong? (1 choice)

02/07

DONE

- ☐ Analyse user data.
- ☐ Did a user test, user data entries: 20.
- ☐ Found out why there were so many broken files!
- ☐ Rescraped everything, leaving out the LP scans wherever possible.
- ☐ JSON that is exported looks something like this:


```
{ profession: 'Other',
otherprofession: 'creative technologist',
age: '31-35',
gender: 'Female',
'image-1': [ '103555.jpg',
       'neutral',
       'no-reason'],
'image-2': [ '109002.jpg',
       'good',
       'colors',
       'composition'],
'image-3': [ '129974.jpg',
       'meh',
       'looks like detergent logo' ],
'image-4': [ '76127.jpg',
       'neutral',
       'composition'],
'image-5': [ '16056.jpg',
       'good',
       'colors',
       'composition'],
'image-6': [ '126020.jpg',
       'good',
       'colors',
       'type',
       'fit'],
'image-7': [ '118523.jpg',
       'bad',
       'composition',
       'fit'],
'image-8': [ '81710.jpg',
       'good',
       'colors',
       'composition'],
'image-9': [ '612168.jpg',
       'good',
       'type',
       'composition'],
'image-10': [ '79926.jpg',
       'bad',
       'fit'],
'image-11': [ '1057436.jpg',
       'meh',
       'fit'],
'image-12': [ '10953.jpg',
       'meh',
       'colors',
       'composition'],
'image-13': [ '1431755.jpg',
       'bad',
       'composition'],
'image-14': [ '65088.jpg',
       'neutral',
       'no-reason'],
'image-15': [ '615945.jpg',
       'meh',
       'colors'],
mail: 'example@gmail.com',
btn: 'Save my feedback!' }
```


08/07

- ☐ arxiv.org/abs/1805.00728
- ☐ github.com/TheHedgeify/DagstuhlGAN

Found a paper on Mario levels that also uses a GAN icw a CMA-ES:

This project allows for the unsupervised learning of a Generative Adversarial Network (GAN) that understands the structure of Super Mario Bros. levels. The model is trained on actual Mario levels from the Video Game Level Corpus. The trained model is capable of generating new level segments with the input of a latent vector, and these segments can be stitched together to make complete levels. In order to find the best level segments within this latent space, the evolutionary algorithm Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is used to find latent vectors producing level segments that either optimize some sort of tile distribution or result in a particular level of performance by an artificial agent. The resulting system helps discover new levels in the space of examples created by human experts.

TO DO

- ☐ Redo the entire log
- ☐ Analyse user data

The number of curated entries is listed in this overview:

Covers	146.217
Electronic	
Ambient	2.182
Minimal	2.619
New+Age	2.238
Trance	7.433
Hip+Hop	
Conscious	2.461
Gangsta	4.130
Pop+Rap	4.153
RnB%2FSwing	5.925
Jazz	
Big+Band	1.761
Fusion	2.260
Smooth+Jazz	1.892
Swing	2.356
Pop	
Europop	9.317
J-pop	3.096
Pop+Rock	7.907
Schlager	3.838
Rock	
Experimental	5.830
Heavy+Metal	8.196
Psychedelic+Rock	5.002
Punk	9.672
Country+Folk	
Country	9.970
Folk	9.642
Folk+Rock	5.573
Vocal	4.010
Classical	
Baroque	5.767
Classical	5.903
Modern	3.860
Romantic	9.224

15/07

STYLEGAN

We propose an alternative generator architecture for generative adversarial networks, borrowing from style transfer literature. The new architecture leads to an automatically learned, unsupervised separation of high-level attributes (e.g., pose and identity when trained on human faces) and stochastic variation in the generated images (e.g., freckles, hair), and it enables intuitive, scale-specific control of the synthesis. The new generator improves the state-of-the-art in terms of traditional distribution quality metrics, leads to demonstrably better interpolation properties, and also better disentangles the latent factors of variation. To quantify interpolation quality and disentanglement, we propose two new, automated methods that are applicable to any generator architecture. Finally, we introduce a new, highly varied and high-quality dataset of human faces.

- ☐ High quality GAN results towardsdatascience.com/a-shortcut-to-achieving-high-quality-gan-results-bca64a5b9394.
- ☐ Are GANs created equal? arxiv.org/pdf/1711.10337.pdf.
- ☐ Some covers were already generated at lower quality researchgate.net/publication/318987126_Album_Cover_Generation_from_Genre_Tags.

FRAMEWORK

- ☐ Album covers: researchgate.net/publication/260424430_Death_and_resurrection_of_the_album_cover.
- ☐ Album covers: academia.edu/27428788/After_the_Storm_Hipgnosis_Storm_Thorgerson_and_the_Rock_Album_Cover.
- ☐ Album covers from the sixties, S Thorgerson (2005) Collins & Brown: bol.com/nl/f/classic-album-covers-of-the-60s/30142517/#.
- ☐ Classic LP sleeves: amazon.com/Art-LP-Classic-Album-Covers/dp/145491806.

WHAT MAKES AN ALBUM COVER GOOD?

You Can Judge an Artist by an Album Cover: Using Images for Music Annotation, Janis Lībeks.

Using Convolutional Neural Network Filters to Measure Left-Right Mirror Symmetry in Images

"We propose a method for measuring symmetry in images by using filter responses from Convolutional Neural Networks (CNNs). The aim of the method is to model human perception of left/right symmetry as closely as possible." They do that with album covers as well.

"We validated our algorithm on a dataset of 300 music album covers, which were rated according to their symmetry by 20 human observers..."

GENERATING ALBUM COVERS

Album Cover Generation from Genre Tags" – Alexander Hepburn, Ryan McConville, and Rául Santos-Rodríguez, University of Bristol.

22/07

Titanic hits iceberg

DONE

- ☐ Found a new computer to work on.
- ☐ Mailed the Reddit guy that practically did the same stuff I wanted to do (apart from the research).
- ☐ Found papers for framework.

DEBUG LOG

Code needs to recognise the GPU. wrote a test script, save as `test-gpu.py`

```
// Should show a GPU:0
import tensorflow as tf;
from tensorflow.python.
client import device_lib
print(device_lib.list_lo-
cal_devices())
```

Code works with protobuf v.3.6.0, somehow. run via

```
pip install proto-
buff==3.6.0
```

APPARENTLY ANACONDA DEV ENVIRONMENTS ARE THE THING

IT WORKS NOW

WHAT HAD TO BE DONE TO MAKE IT WORK

- ☐ Install the latest NVIDIA driver for your graphics card
- ☐ Install CUDA + CUDnn (would go for 10 because apparently tensorflow likes that?)
- ☐ Download ANACONDA 3 (Python 3.7) repo.anacoda.com/archive
- ☐ Create an environment in Anaconda prompt:

```
conda create -n
[name] python=3.6
```

- ☐ Activate the environment:

```
conda activate
[name]
```

- ☐ `pip` install:

```
tensorflow-gpu
requests
Pillow
scipy
```

- ☐ Run the test to check if your GPU is recognised by tensorflow:

```
python test-gpu.py
```

- ☐ GO GO GO GO GO

DONE

- ☐ Read about GPU clusters: timdettmers.com/2014/09/21/how-to-build-and-use-a-multi-gpu-system-for-deep-learning/
- ☐ Also transported a bag of computer components to Amsterdam from Belgium and set up a computer from scratch.
- ☐ Found out how to continue training from a previous session (make sure the code is set to making snapshots).
- ☐ In `training/train.py` set the `resume_run_id` to the number of the folder that last created a snapshot, ALSO set the `resume_kimg` value to the latest logged value (check your latest file).
- ☐ Set off the sleep mode to 20000 minutes (13 days).
- ☐ Fixed auto login.
- ☐ Bought power and a Geforce RTX 2080 Titan graphics card.
- ☐ Watched youtube.com/watch?v=LYQEJnDorF4.
- ☐ Debugged the computer for days, it was probably too hot. So I added an external (human) fan.
- ☐ Found this link: easeus.com/data-recovery/fix-whea-uncorrectable-error-windows-10.html

23/07

- ☐ Finding out how to combine a GAN and a CMA-ES to optimise stuff.
- ☐ Finding out how to make a GAN understand what it is training (tags/genre).

A Generative Adversarial Network (GAN) is trained on a set of real fingerprint images, then a Covariance Matrix Adaptation Evolution Strategy (CMA-ES) is used to search for effective latent input variables to the generator network that can optimise the success rate of the GAN.

Taken from "Album Cover Generation from Genre Tags"

The GAN could be implemented as a cover generator for services such as Spotify or Soundcloud.

The variables in the discriminator are optimised to distinguish between real and generated input, whilst the variables in the generator are optimised to fool the discriminator into classifying the generated input as real. The generator thus learns how to create images that looks as realistic as the actual images, while the discriminator learns to discriminate between generates images and images from the dataset.

The GAN can be trained on data throughout the use of latent variables.

Apart from a lot of availability, album covers pose issues as a medium as well. As compared to frequently used image datasets for training GANs such as the CelebA or MNIST datasets, album covers have a enormous variety of elements in them as well as different design styles and different typography. The lack of a labelled training dataset is a challenge as well.

29/07

TO DO

- ☐ Finding a second thesis advisor.
- ☐ Writing, writing, writing...
- ☐ Filter GAN by genre in a way.
- ☐ Somehow dividing the latent space per genre?

POSSIBLE TECHNIQUES

- ☐ AC-GAN: adding tags in neural networks.
- ☐ CMA-ES: finding the most optimal elements.

FRAMEWORK

GANs learn to generate images in an unsupervised fashion. There are two parts to a GAN: a generator and a discriminator. The generator is typically a neural network that inputs random noise and outputs an image.

The discriminator is also typically a neural network, which inputs an image and classifies it as being 'real' or 'generated'. GANS three steps:

- ☐ Provide real images to the discriminator. Train the discriminator to classify them as real.
- ☐ Provide generated images to the discriminator. Train the generator to classify them as generated.
- ☐ Provide the generator with the discriminator's gradients. Train the generator to produce images that are classified as real.

A major difficulty during training is keeping the two networks balanced so one does not become significantly better than the other. Much work, since the invention of GANs, has focused on stabilizing the training process; two popular approaches are the Wasserstein GAN (WGAN) and WGAN with gradient penalty.

03/08

GAN is GANning.

19/08

- ☐ Trained 3 weeks on PUNK covers.
- ☐ Retrained 3 days on ALL.
- ☐ Drastic improvements!

26/08

- ☐ Trained 1 week on ALL.
- ☐ Updated reading list.

02/09

I came back from holidays, the computer is fortunately still working. Training on all covers for now and then on separate genres and subgenres.

09/09

After training on the entire dataset for another 3 weeks, I started training it per genre and subgenre.

Training more:
- ☐ 7 genres, 28 subgenres
- ☐ 28 x 3 = 84 days
- ☐ 7 x 5 = 35 days

Jason Brownlee, Interpolation and Vector Arithmetic for GANs machinelearningmastery.com/how-to-interpolate-and-perform-vector-arithmetic-with-faces-using-a-generative-adversarial-network/.

The important 2015 paper by Alec Radford, et al. titled “Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks” introduced a stable model configuration for training deep convolutional neural network models as part of the GAN architecture.

16/09

The latent space is an abstract model used to navigate the GAN architecture.

I'M INTERPOLATING IN THE LATENT SPACE USING VECTOR ARITHMETIC

- ☐ Wrote a piece of code to generate videos.
- ☐ Linear interpolation between *x* points with *n* steps.

23/09

Just found out that there is a setting called `train.mirror_augment` that was set to `True`, reflecting the images to augment training, BUT WE DON'T WANT THAT WITH TEXT!

The past time I already saw some letters that were mirrored, but I didn't think much of it, maybe the typography in the examples was just eccentric.

I found out there was an actual issue when I was training on Gangsta Hip Hop and half of the Parental Advisory stickers was mirrored... Perfectly crisp and legible, but mirrored. Then I dived into the code and saw in one of the many files that there was the `mirror_augment` setting.

Oh well, trial and error worked out for me so far...

24/09

I trained it one night without mirroring and it seems to be fixed, at least for the parental advisory stickers! Fairly quick adaptation back to unmirrored Parental Advisory stickers.

30/09

The text is much better now, and has become almost entirely unmirrored.

07/10

Trained:

- ☐ Electronic
- ☐ Trance

- ☐ Hip + Hop
- ☐ Gangsta

- ☐ Jazz
- ☐ Swing

- ☐ Pop
- ☐ Europop

- ☐ Rock
- ☐ Heavy+Metal

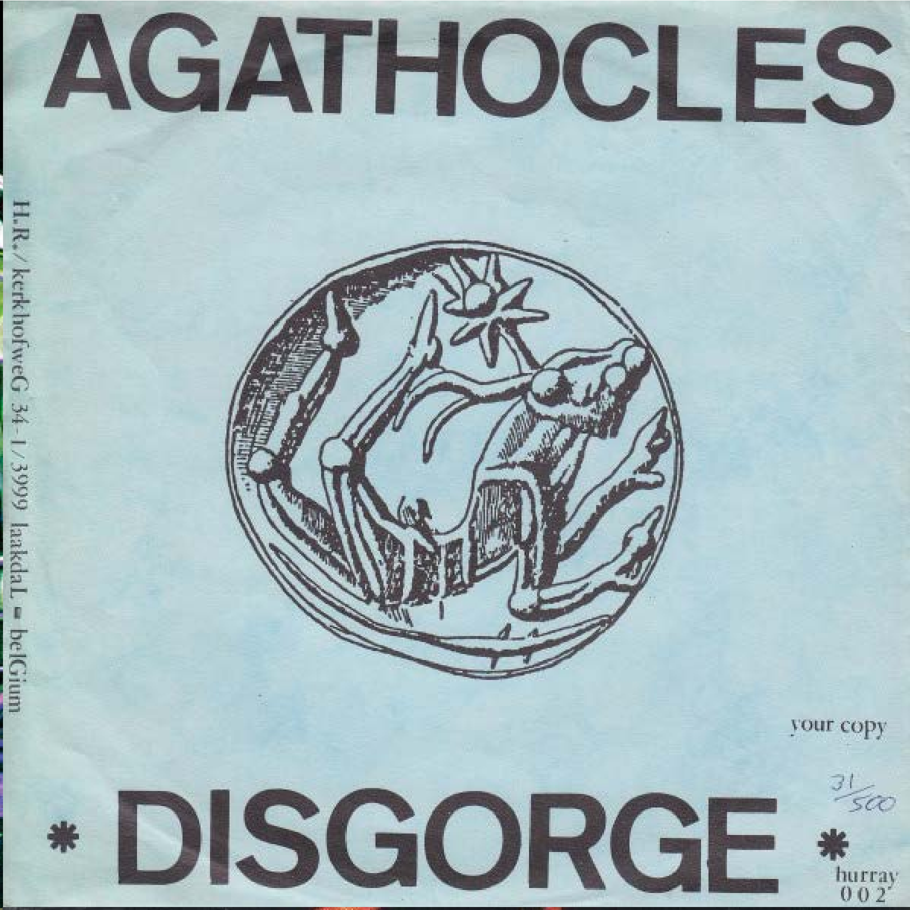
- ☐ Country+Folk
- ☐ Country

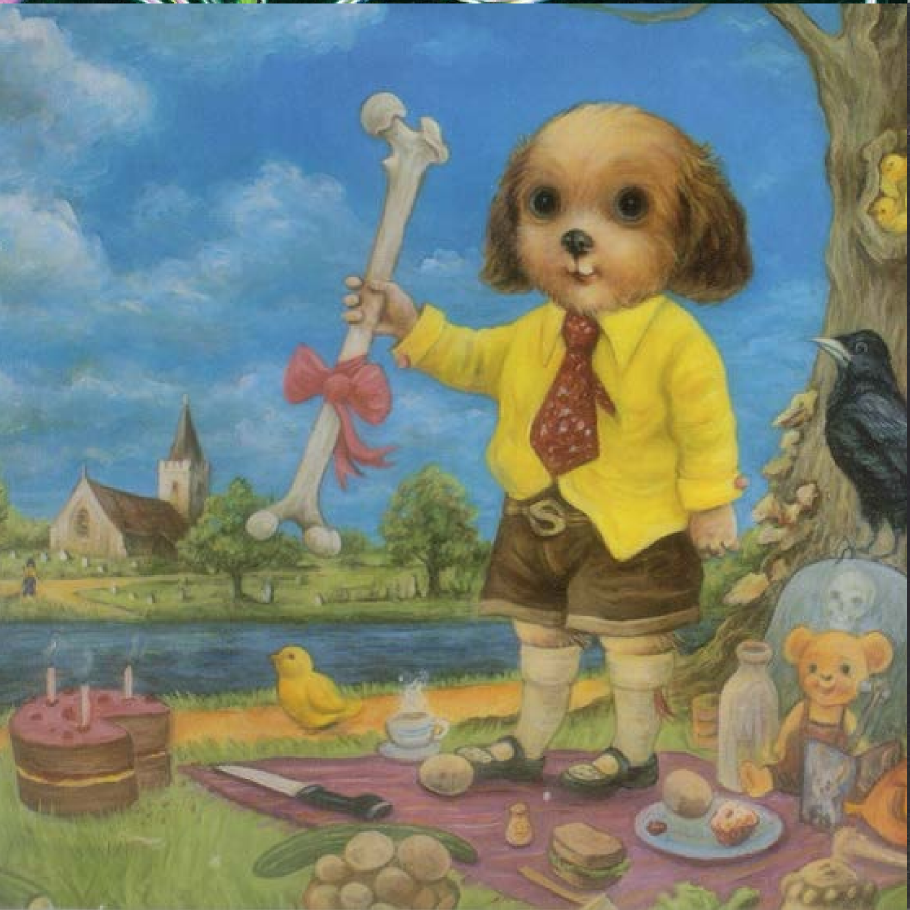
- ☐ Classical
- ☐ Romantic

14/10

I am very happy with the results for now. The different models render quite crisp compositions and the difference per genre is really noticeable.


AN OVERDOSE OF DEATH...

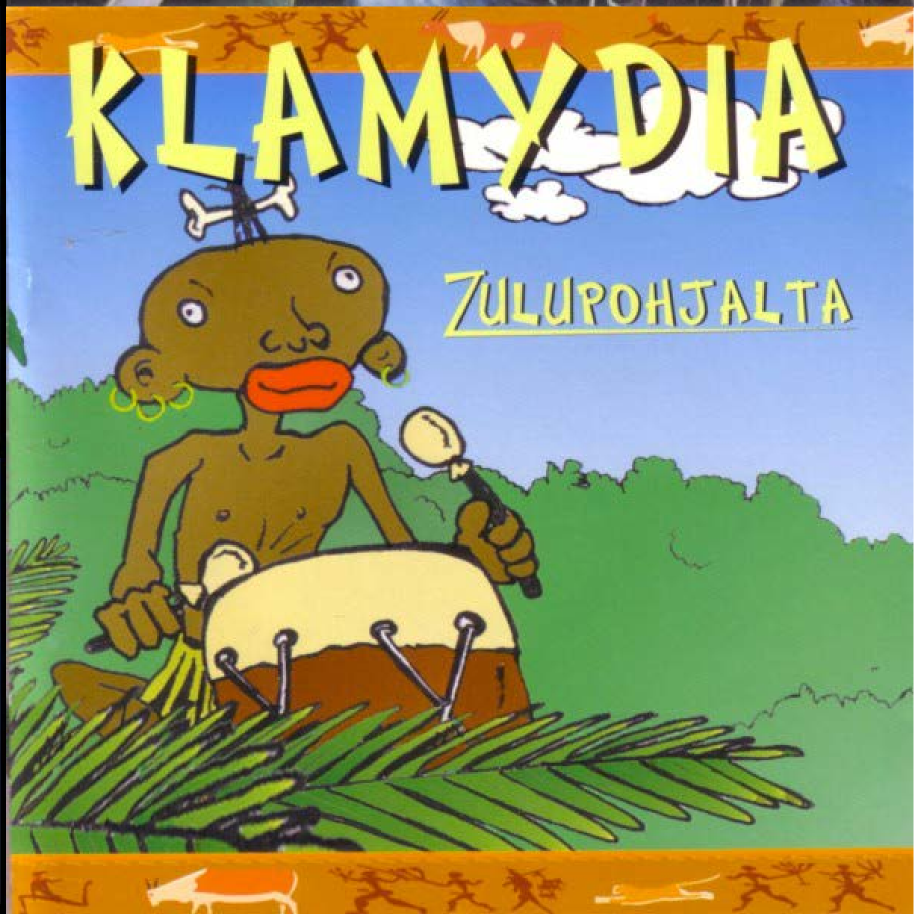

AGATHOCLES
H.R./ kerkhofweG 34 1/3999 laakdaL - belGium
your copy
31/500
* DISGORGE *
hurray
002




VAGINAL JESUS
AFFIRMATIVE APARTHEID

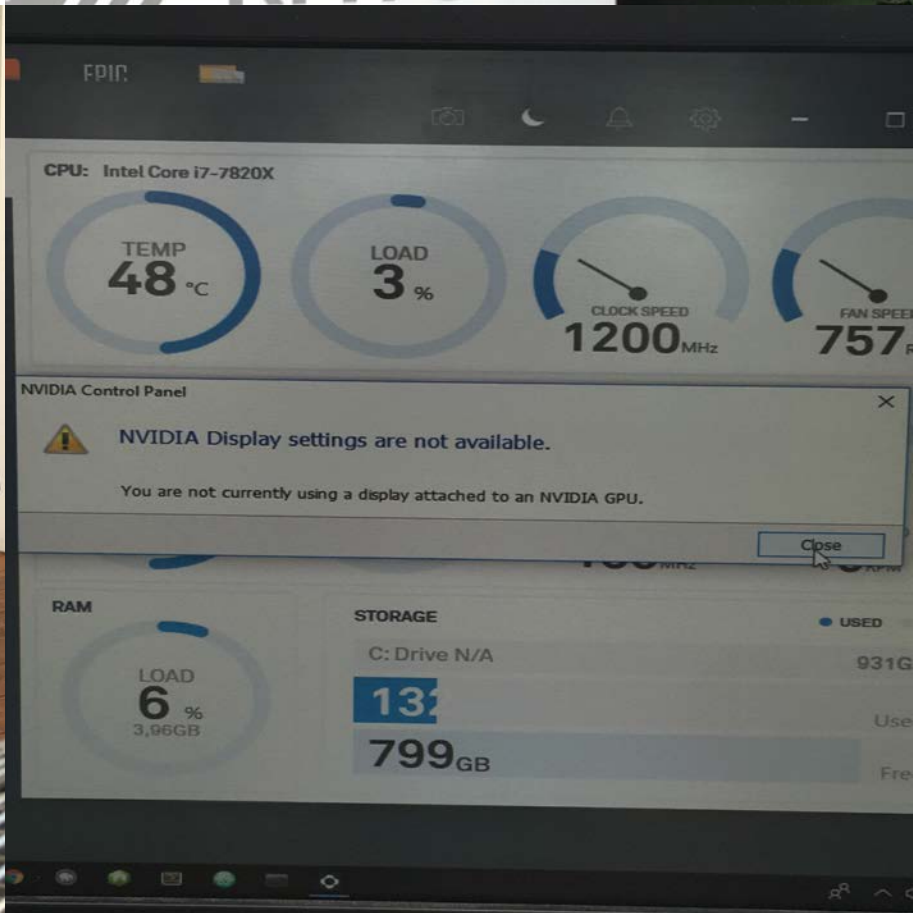

/Covers/Rock/Punk//777759.jpg resized and saved!
/Covers/Rock/Punk//778475.jpg resized and saved!
/Covers/Rock/Punk//779820.jpg resized and saved!
/Covers/Rock/Punk//780315.jpg resized and saved!
/Covers/Rock/Punk//778768.jpg resized and saved!
/Covers/Rock/Punk//77906.jpg resized and saved!
/Covers/Rock/Punk//779547.jpg resized and saved!
/Covers/Rock/Punk//779406.jpg resized and saved!
/Covers/Rock/Punk//780358.jpg resized and saved!
/Covers/Rock/Punk//780346.jpg resized and saved!
/Covers/Rock/Punk//780338.jpg resized and saved!
/Covers/Rock/Punk//780572.jpg resized and saved!
/Covers/Rock/Punk//780653.jpg resized and saved!
/Covers/Rock/Punk//780343.jpg resized and saved!
/Covers/Rock/Punk//780885.jpg resized and saved!
/Covers/Rock/Punk//780470.jpg resized and saved!
/Covers/Rock/Punk//780392.jpg resized and saved!
/Covers/Rock/Punk//780746.jpg resized and saved!
/Covers/Rock/Punk//781575.jpg resized and saved!
/Covers/Rock/Punk//781796.jpg resized and saved!
/Covers/Rock/Punk//782313.jpg resized and saved!
/Covers/Rock/Punk//782689.jpg resized and saved!
/Covers/Rock/Punk//782457.jpg resized and saved!


AURORA

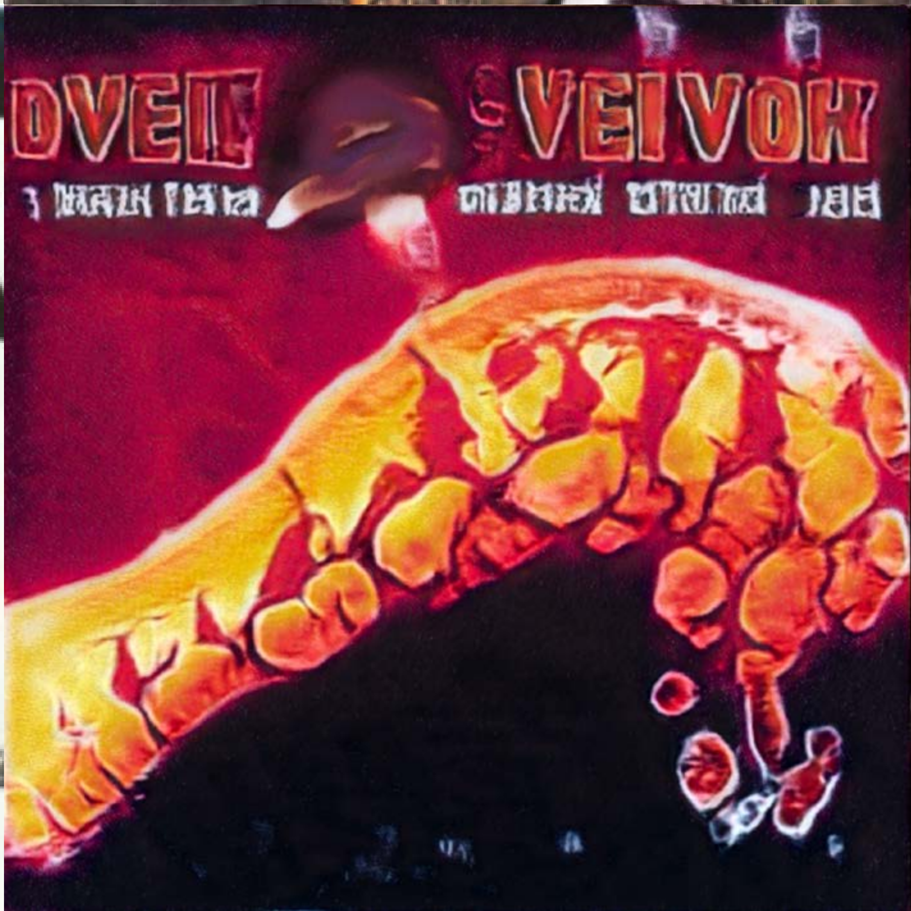

KLAMYDIA
ZULUPOHJALTA


world contact day
GROOVIE
GHOULIES

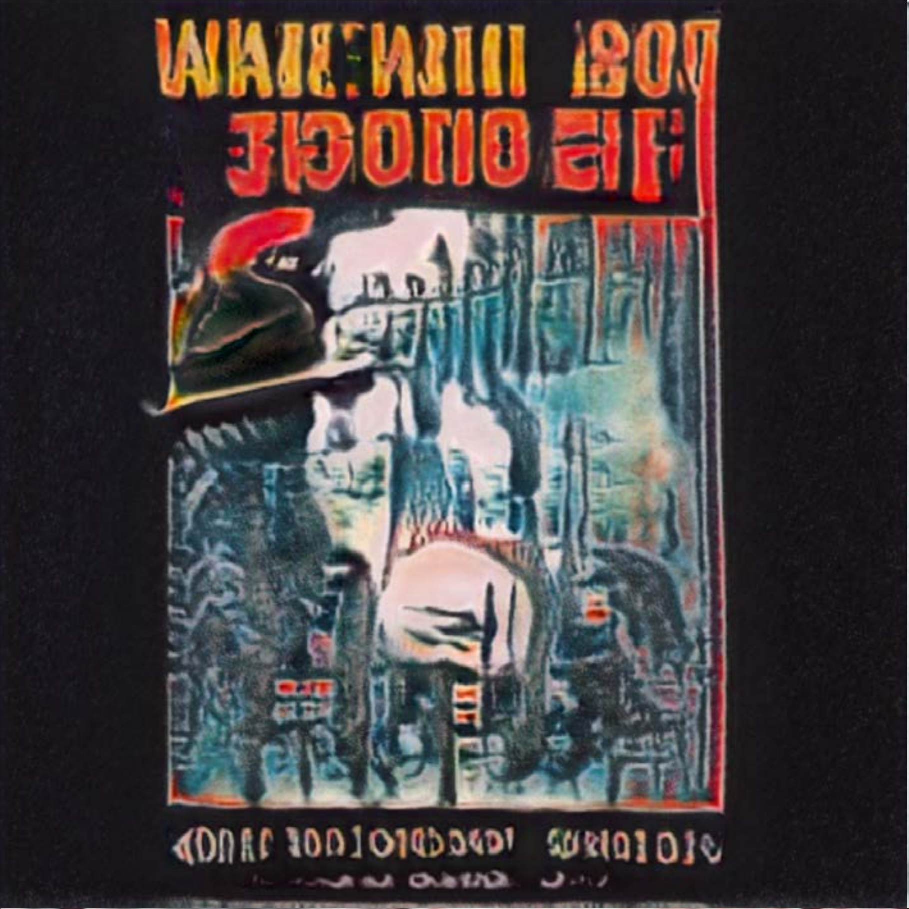
GIGABYTE
FAN STOP
GIGABYTE
FAN STOP
RM750x

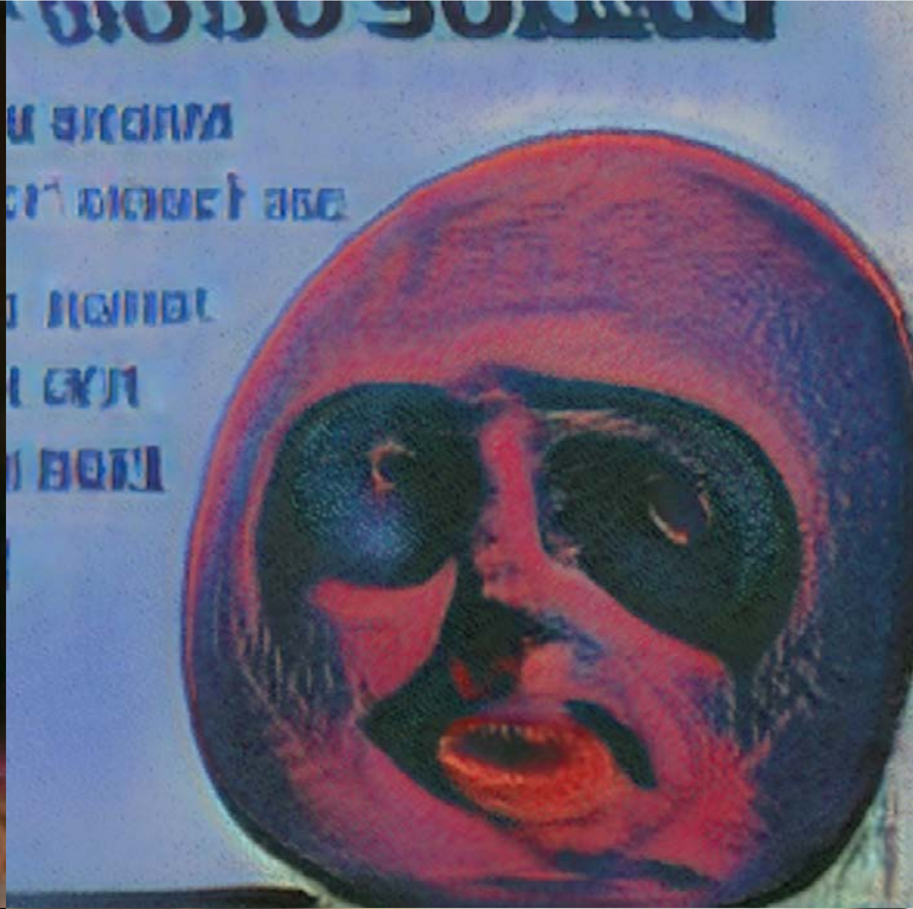
RM750x
GIGABYTE

CPU: Intel Core i7-7820X
TEMP
48 °C
LOAD
3 %
CLOCK SPEED
1200 MHz
NVIDIA Control Panel
NVIDIA Display settings are not available.
You are not currently using a display attached to an NVIDIA GPU.
Close
RAM
LOAD
6 %
3,96GB
STORAGE
USED
C: Drive N/A
799 GB





DVEI
VEIVOH



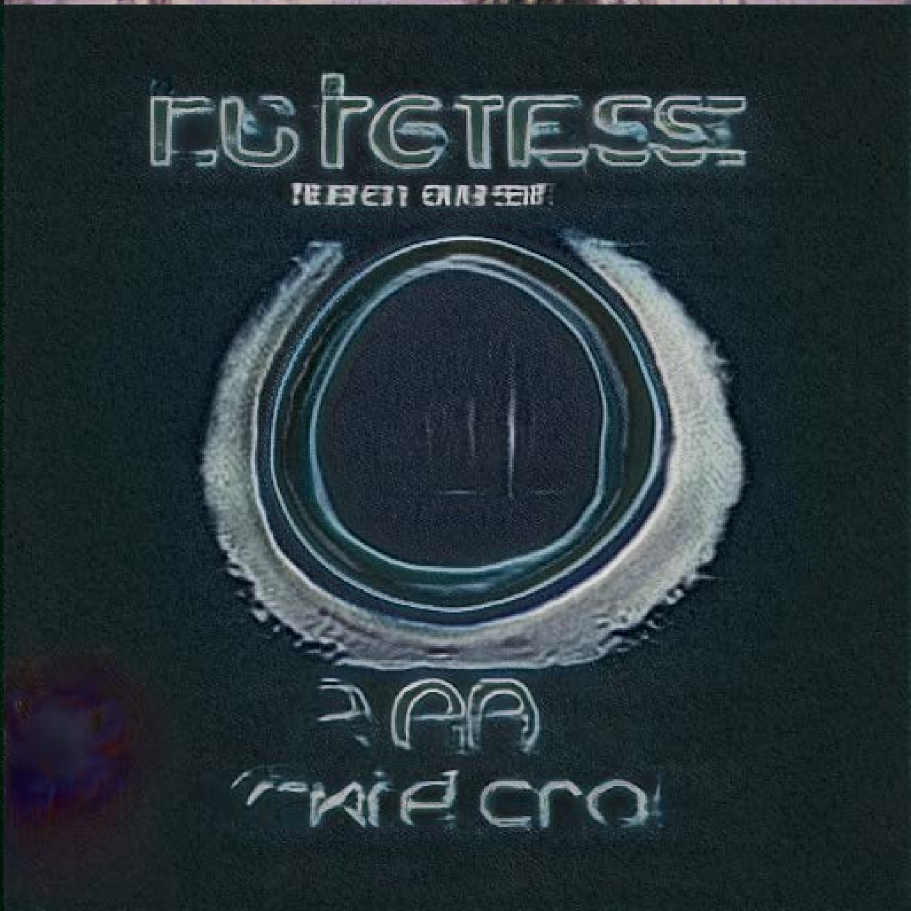



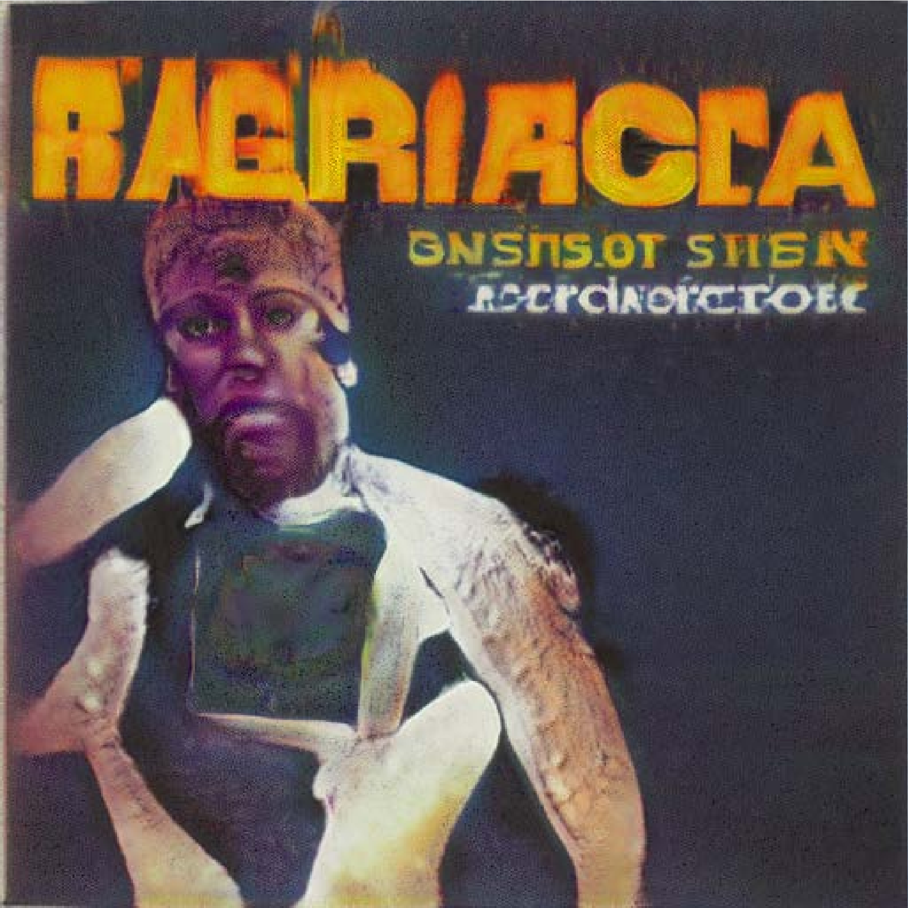



```
k 282   kimg 13500.0  lod 0.00  minibatch 4   time 13d 01h 46m  sec/tick 4204.2  sec/kimg 140.14  maintenance 3.1    gpumem 5.7
k 283   kimg 13530.0  lod 0.00  minibatch 4   time 13d 02h 57m  sec/tick 4206.2  sec/kimg 140.21  maintenance 3.0    gpumem 5.7
k 284   kimg 13560.0  lod 0.00  minibatch 4   time 13d 04h 07m  sec/tick 4205.6  sec/kimg 140.19  maintenance 3.1    gpumem 5.7
k 285   kimg 13590.0  lod 0.00  minibatch 4   time 13d 05h 17m  sec/tick 4207.3  sec/kimg 140.24  maintenance 3.1    gpumem 5.7
k 286   kimg 13620.0  lod 0.00  minibatch 4   time 13d 06h 27m  sec/tick 4210.3  sec/kimg 140.34  maintenance 3.1    gpumem 5.7
k 287   kimg 13650.0  lod 0.00  minibatch 4   time 13d 07h 37m  sec/tick 4208.1  sec/kimg 140.27  maintenance 3.1    gpumem 5.7
k 288   kimg 13680.0  lod 0.00  minibatch 4   time 13d 08h 47m  sec/tick 4207.1  sec/kimg 140.24  maintenance 3.1    gpumem 5.7
k 289   kimg 13710.0  lod 0.00  minibatch 4   time 13d 09h 58m  sec/tick 4205.9  sec/kimg 140.20  maintenance 3.1    gpumem 5.7
k 290   kimg 13740.0  lod 0.00  minibatch 4   time 13d 11h 08m  sec/tick 4205.9  sec/kimg 140.20  maintenance 3.1    gpumem 5.7
twork-snapshot-013740       time 13m 28s      fid50k 31.6295
k 291   kimg 13770.0  lod 0.00  minibatch 4   time 13d 12h 31m  sec/tick 4205.7  sec/kimg 140.19  maintenance 811.9  gpumem 5.7
k 292   kimg 13800.0  lod 0.00  minibatch 4   time 13d 13h 42m  sec/tick 4205.2  sec/kimg 140.17  maintenance 3.1    gpumem 5.7
k 293   kimg 13830.0  lod 0.00  minibatch 4   time 13d 14h 52m  sec/tick 4203.8  sec/kimg 140.13  maintenance 3.1    gpumem 5.7
k 294   kimg 13860.0  lod 0.00  minibatch 4   time 13d 16h 02m  sec/tick 4204.4  sec/kimg 140.15  maintenance 3.0    gpumem 5.7
k 295   kimg 13890.0  lod 0.00  minibatch 4   time 13d 17h 12m  sec/tick 4204.7  sec/kimg 140.16  maintenance 3.1    gpumem 5.7
k 296   kimg 13920.0  lod 0.00  minibatch 4   time 13d 18h 22m  sec/tick 4204.5  sec/kimg 140.15  maintenance 3.1    gpumem 5.7
k 297   kimg 13950.0  lod 0.00  minibatch 4   time 13d 19h 32m  sec/tick 4204.0  sec/kimg 140.13  maintenance 3.1    gpumem 5.7
k 298   kimg 13980.0  lod 0.00  minibatch 4   time 13d 20h 42m  sec/tick 4204.6  sec/kimg 140.15  maintenance 3.0    gpumem 5.7
k 299   kimg 14010.0  lod 0.00  minibatch 4   time 13d 21h 52m  sec/tick 4206.1  sec/kimg 140.20  maintenance 3.1    gpumem 5.7
k 300   kimg 14040.0  lod 0.00  minibatch 4   time 13d 23h 03m  sec/tick 4205.5  sec/kimg 140.18  maintenance 3.1    gpumem 5.7
twork-snapshot-014040       time 13m 30s      fid50k 31.2754
k 301   kimg 14070.0  lod 0.00  minibatch 4   time 14d 00h 26m  sec/tick 4203.3  sec/kimg 140.11  maintenance 813.6  gpumem 5.7
k 302   kimg 14100.0  lod 0.00  minibatch 4   time 14d 01h 36m  sec/tick 4204.0  sec/kimg 140.13  maintenance 3.1    gpumem 5.7
k 303   kimg 14130.0  lod 0.00  minibatch 4   time 14d 02h 46m  sec/tick 4204.7  sec/kimg 140.16  maintenance 3.1    gpumem 5.7
k 304   kimg 14160.0  lod 0.00  minibatch 4   time 14d 03h 57m  sec/tick 4203.8  sec/kimg 140.13  maintenance 3.1    gpumem 5.7
k 305   kimg 14190.0  lod 0.00  minibatch 4   time 14d 05h 07m  sec/tick 4205.1  sec/kimg 140.17  maintenance 3.1    gpumem 5.7
k 306   kimg 14220.0  lod 0.00  minibatch 4   time 14d 06h 17m  sec/tick 4204.9  sec/kimg 140.16  maintenance 3.1    gpumem 5.7
k 307   kimg 14250.0  lod 0.00  minibatch 4   time 14d 07h 27m  sec/tick 4206.3  sec/kimg 140.21  maintenance 3.1    gpumem 5.7
k 308   kimg 14280.0  lod 0.00  minibatch 4   time 14d 08h 37m  sec/tick 4204.4  sec/kimg 140.15  maintenance 3.1    gpumem 5.7
k 309   kimg 14310.0  lod 0.00  minibatch 4   time 14d 09h 47m  sec/tick 4204.4  sec/kimg 140.15  maintenance 3.1    gpumem 5.7
k 310   kimg 14340.0  lod 0.00  minibatch 4   time 14d 10h 57m  sec/tick 4206.6  sec/kimg 140.22  maintenance 3.1    gpumem 5.7
twork-snapshot-014340       time 13m 29s      fid50k 32.7394
k 311   kimg 14370.0  lod 0.00  minibatch 4   time 14d 12h 21m  sec/tick 4205.2  sec/kimg 140.17  maintenance 813.4  gpumem 5.7
k 312   kimg 14400.0  lod 0.00  minibatch 4   time 14d 13h 31m  sec/tick 4207.1  sec/kimg 140.24  maintenance 3.1    gpumem 5.7
k 313   kimg 14430.0  lod 0.00  minibatch 4   time 14d 14h 41m  sec/tick 4206.9  sec/kimg 140.23  maintenance 3.1    gpumem 5.7
k 314   kimg 14460.0  lod 0.00  minibatch 4   time 14d 15h 52m  sec/tick 4208.4  sec/kimg 140.28  maintenance 3.1    gpumem 5.7
k 315   kimg 14490.0  lod 0.00  minibatch 4   time 14d 17h 02m  sec/tick 4207.9  sec/kimg 140.26  maintenance 3.1    gpumem 5.7
k 316   kimg 14520.0  lod 0.00  minibatch 4   time 14d 18h 12m  sec/tick 4209.6  sec/kimg 140.32  maintenance 3.1    gpumem 5.7
k 317   kimg 14550.0  lod 0.00  minibatch 4   time 14d 19h 22m  sec/tick 4210.9  sec/kimg 140.36  maintenance 3.1    gpumem 5.7
k 318   kimg 14580.0  lod 0.00  minibatch 4   time 14d 20h 32m  sec/tick 4209.7  sec/kimg 140.32  maintenance 3.1    gpumem 5.7
k 319   kimg 14610.0  lod 0.00  minibatch 4   time 14d 21h 43m  sec/tick 4210.0  sec/kimg 140.33  maintenance 3.1    gpumem 5.7
k 320   kimg 14640.0  lod 0.00  minibatch 4   time 14d 22h 53m  sec/tick 4207.6  sec/kimg 140.25  maintenance 3.1    gpumem 5.7
twork-snapshot-014640       time 13m 29s      fid50k 33.0560
k 321   kimg 14670.0  lod 0.00  minibatch 4   time 15d 00h 17m  sec/tick 4210.1  sec/kimg 140.34  maintenance 813.1  gpumem 5.7
k 322   kimg 14700.0  lod 0.00  minibatch 4   time 15d 01h 27m  sec/tick 4205.6  sec/kimg 140.19  maintenance 3.1    gpumem 5.7
k 323   kimg 14730.0  lod 0.00  minibatch 4   time 15d 02h 37m  sec/tick 4207.7  sec/kimg 140.26  maintenance 3.1    gpumem 5.7
k 324   kimg 14760.0  lod 0.00  minibatch 4   time 15d 03h 47m  sec/tick 4208.8  sec/kimg 140.29  maintenance 3.1    gpumem 5.7
k 325   kimg 14790.0  lod 0.00  minibatch 4   time 15d 04h 57m  sec/tick 4211.8  sec/kimg 140.39  maintenance 3.1    gpumem 5.7
k 326   kimg 14820.0  lod 0.00  minibatch 4   time 15d 06h 08m  sec/tick 4219.7  sec/kimg 140.66  maintenance 3.1    gpumem 5.7
k 327   kimg 14850.0  lod 0.00  minibatch 4   time 15d 07h 18m  sec/tick 4217.4  sec/kimg 140.58  maintenance 3.1    gpumem 5.7
k 328   kimg 14880.0  lod 0.00  minibatch 4   time 15d 08h 28m  sec/tick 4209.4  sec/kimg 140.31  maintenance 3.1    gpumem 5.7
k 329   kimg 14910.0  lod 0.00  minibatch 4   time 15d 09h 38m  sec/tick 4210.8  sec/kimg 140.36  maintenance 3.1    gpumem 5.7
k 330   kimg 14940.0  lod 0.00  minibatch 4   time 15d 10h 48m  sec/tick 4168.2  sec/kimg 138.94  maintenance 3.0    gpumem 5.7
twork-snapshot-014940       time 13m 07s      fid50k 31.9391
k 331   kimg 14970.0  lod 0.00  minibatch 4   time 15d 12h 10m  sec/tick 4113.8  sec/kimg 137.13  maintenance 791.1  gpumem 5.7
k 332   kimg 15000.0  lod 0.00  minibatch 4   time 15d 13h 18m  sec/tick 4115.0  sec/kimg 137.17  maintenance 3.1    gpumem 5.7
```

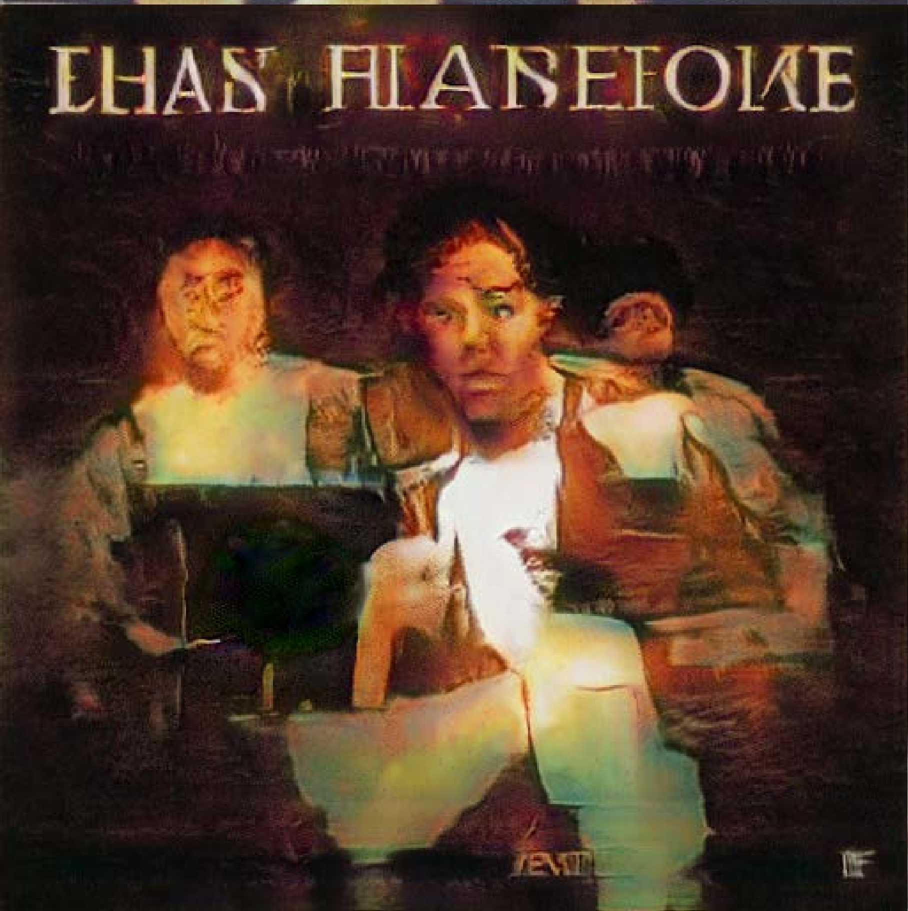

ADVISORY



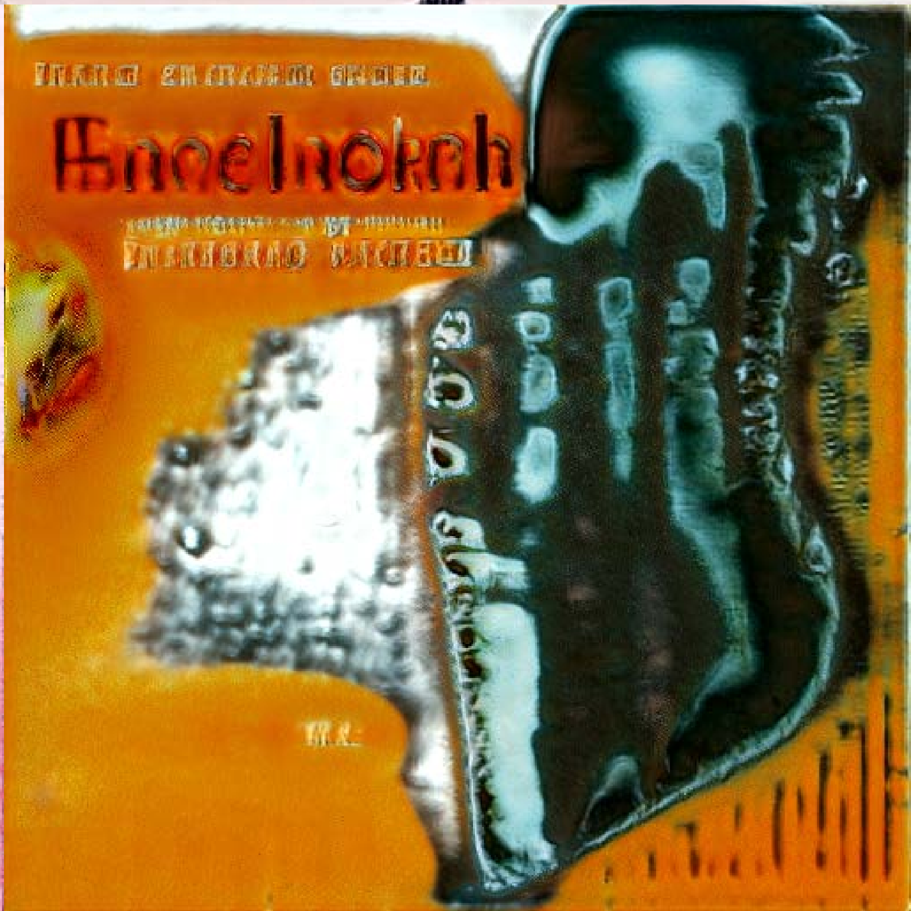


ADVISORY

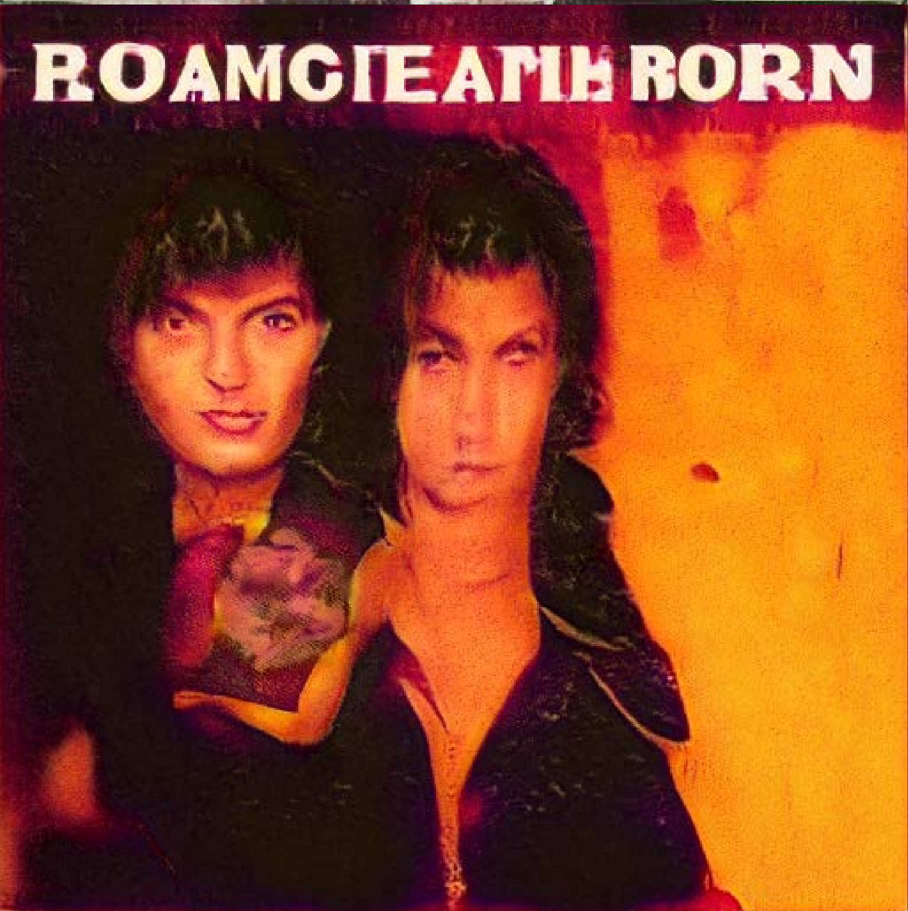


ADVISORY

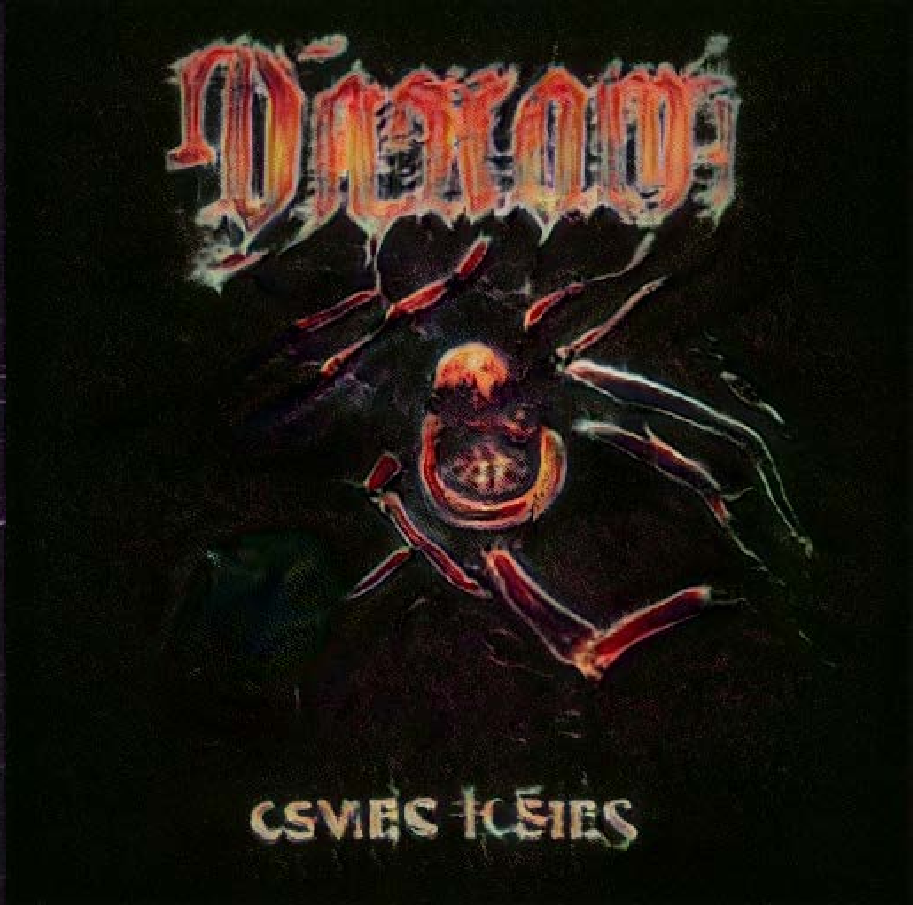


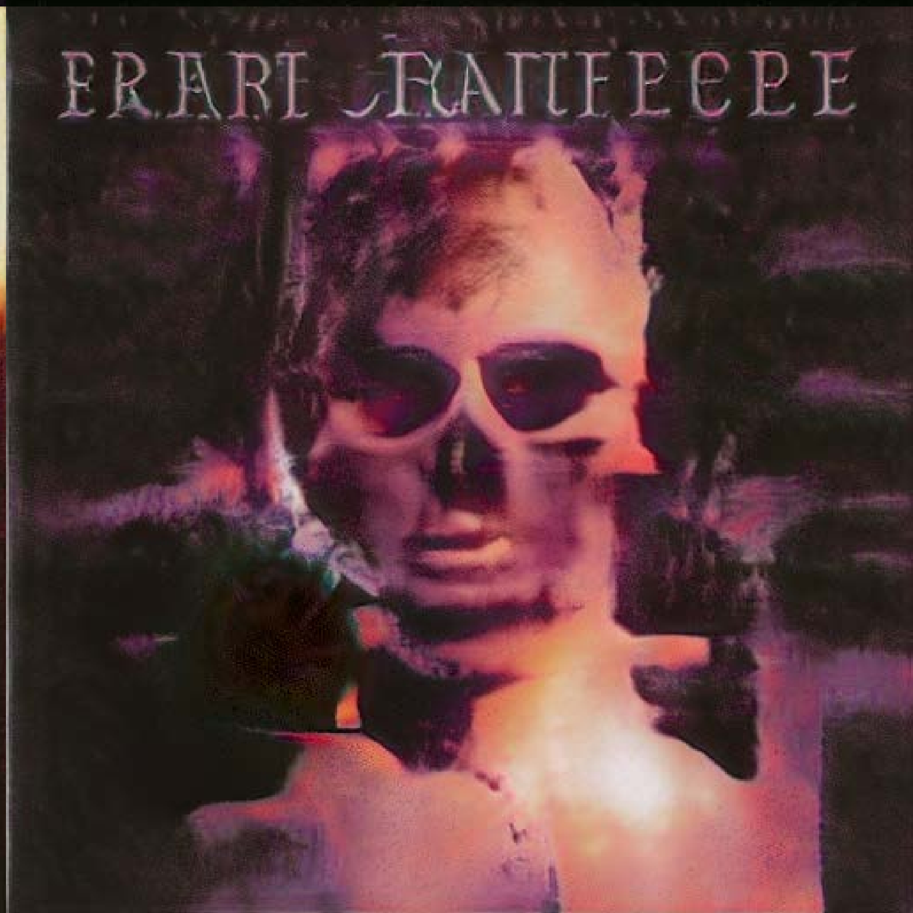














DECCA

(11) ACKNOWL-EDGEMENTS

Tessa Verhoef	Primary advisor
Melvin Wevers	Secondary Advisor

Winke Wiegersma	Proofreading
Sander de Jong	Technical advice
Jae Perris	Technical advice
Gilles de Brock	Hardware advice
Vera Williams	Spell checking
Liduin Stumpel	Spell checking
Studio Zesbaans	Provision of work space

Thanks to the critics, family and friends present at the defense.