Conditional Vector Graphics Generation for Music Cover Images

Ivan Jarsky[a], Valeria Efimova[b], Ilya Bizyaev and Andrey Filchenkov[c]
ITMO University, Kronverksky Pr. 49, St. Petersburg, Russia
fi *fi*

Keywords: GAN, Image Generation, Vector Graphics.

Abstract: Generative Adversarial Networks (GAN) have motivated a rapid growth of the domain of computer image synthesis. As almost all the existing image synthesis algorithms consider an image as a pixel matrix, the high-resolution image synthesis is complicated. A good alternative can be vector images. However, they belong to the highly sophisticated parametric space, which is a restriction for solving the task of synthesizing vector graphics by GANs. In this paper, we consider a specific application domain that softens this restriction dramatically allowing the usage of vector image synthesis. Music cover images should meet the requirements of Internet streaming services and printing standards, which imply high resolution of graphic materials without any additional requirements on the content of such images. Existing music cover image generation services do not analyze tracks themselves; however, some services mostly consider only genre tags. To generate music covers as vector images that reflect the music and consist of simple geometric objects, we suggest a GAN-based algorithm called CoverGAN. The assessment of resulting images is based on their correspondence to the music compared with AttnGAN and DALL-E text-to-image generation according to title or lyrics. Moreover, the significance of the patterns found by CoverGAN has been evaluated in terms of the correspondence of the generated cover images to the musical tracks. Listeners evaluate the music covers generated by the proposed algorithm as quite satisfactory and corresponding to the tracks. Music cover images generation code and demo are available at https://github.com/IzhanVarsky/CoverGAN.

1 INTRODUCTION

Drawing images manually is a time-consuming process, therefore, image synthesis is a trending research direction due to its high demand in many fields. Various Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), Variational Autoencoders (VAEs) (Kingma and Welling, 2019), Autoregressive Models (Oord et al., 2016), and other models (Bond-Taylor et al., 2021) have been proposed to address this challenge. They all work with bitmap generation representing an image as a matrix of pixels. This results in limitations of the image quality when its resolution is meant to be high. Moreover, new networks capable of synthesizing high-resolution bitmap images are very time- and power-consuming.

An alternative to raster images is vector graphics that may help avoid fuzzy lines and artifacts typical of GANs (Hertzmann, 2020) and achieve sufficient image resolution. However, vector image generation from scratch has not been sufficiently studied. Recently, the vector images are generated containing only points or thin curves (Frans et al., 2021; Li et al., 2020), however artists usually draw vector images with color-filled shapes.

Figure 1: Music cover images generated in vector format with the proposed CoverGAN.

[a] https://orcid.org/0000-0003-1107-3363
[b] https://orcid.org/0000-0002-5309-2207
[c] https://orcid.org/0000-0002-1133-8432

Jarsky, I., Efimova, V., Bizyaev, I. and Filchenkov, A.
Conditional Vector Graphics Generation for Music Cover Images.
DOI: 10.5220/0012456100003660

In *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2024) - Volume 2: VISAPP*, pages 233-243
ISBN: 978-989-758-679-8; ISSN: 2184-4321

Being motivated by the potential of vector graphics synthesis and lack of relevant work, we formulate our research question as: is it possible to synthesize visually appealing vector graphic images with machine learning methods?

The main challenge here is that vector images are highly different from raster images. Raster images are represented as a two-dimensional rectangular matrix or grid of square pixels, while a common vector image is an XML (xml, 2024) file containing descriptions of geometric lines, which can be completed or not. The most useful tag of the XML file is the `<path>` tag; it defines a set of rules for drawing straight segments, Bézier curves of the 2nd and 3rd order, and circular arcs. This tag mainly consists of control points specifying the type and shape of the rendered figure.

A promising application area is music cover images, which have retained their importance as a key component of music marketing despite the digital music distribution. Listeners usually navigate music collections by small cover images before actual listening to tracks. The attractiveness of cover image does not determine music quality but draws the attention of online music consumers to discographies (Cook, 2013), and remains one of the most important issues of popular culture, especially on physical media (Medel, 2014). Noteworthy, streaming services require high-resolution cover images (a common recommendation is 3000×3000 pixels). Independent musicians order cover image production from artists, designers, and photographers, or make attempts to create them on their own, which, however, requires special skills musicians usually lack.

Despite the recent advancements in text-to-image generation (Xu et al., 2018; Zhang et al., 2017), few audiovisual models have been developed. Existing models are mostly aimed at correlating sound information with certain real scenes (Qian et al., 2020), actions (Gao et al., 2020), or actors (Oh et al., 2019). Currently, various simplified editors and template constructors exist, but there are only four publicly available Internet services offering musicians computer-generated images as cover images for their musical compositions: Rocklou Album Cover Generator (Gavelin, 2023), Automated Art (aa, 2023), GAN Album Art (Seyp, 2023), and ArtBreeder (art, 2023).

We propose a GAN, which can generate vector graphics using only image supervision. Therefore, we suggest an approach to cover generation in vector format for musical compositions and call it CoverGAN. The samples of generated cover images can be seen in Fig. 1.

The structure of the paper is as follows. In Section 2, we briefly describe the results achieved. In Section 3, we present a novel method for the cover generation task. Experimental evaluation is presented in Section 5. Limitations of the approach presented are discussed in Section 6. Section 7 concludes the paper and outlines future research.

2 RELATED WORK

The construction of an audiovisual generative model using musical composition includes the extraction of its sound characteristics and conditional image generation. Cover generating services have also already been created for musical compositions.

2.1 Music Information Retrieval

Music information retrieval (MIR) from audio data is a well-studied interdisciplinary field of research (Choi et al., 2017; Schedl et al., 2014; Moffat et al., 2015) based on signal processing, psychoacoustics and musicology. Currently, various information can be obtained automatically from audio data. Librosa library (lib, 2024) can extract mel-frequency cepstral coefficients (MFCC) (Davis and Mermelstein, 1980), which measure the timbre of a music piece and are often used as a feature for speech recognition. They are also widely used for classification based on acoustic events in the habitat. Essentia library (Bogdanov et al., 2013; ess, 2024) offers methods for extracting tonality, chords, harmonies, melody, main pitch, beats per minute, rhythm, etc.

2.2 Conditional Image Generation

Conditional image generation is the process of constructing images corresponding to specified criteria based on certain input data (most often categorical). The Conditional Generative Adversarial Network (cGAN) (Mirza and Osindero, 2014) is one of the most popular model architecture applied. Unlike a common unconditional GAN, in this model the condition is passed to the input of both the generator and the discriminator. It becomes possible to generate an image based on a text condition.

Successful modification of the cGAN model is AttnGAN (Xu et al., 2018). This model considers the mechanism of attention (Vaswani et al., 2017) as a learning factor, which allows selecting words to generate image fragments. Due to modifications, this network shows significantly better results than traditional GAN systems. ObjGAN (Li et al., 2019) also uses the attention. However, its basic principle of image

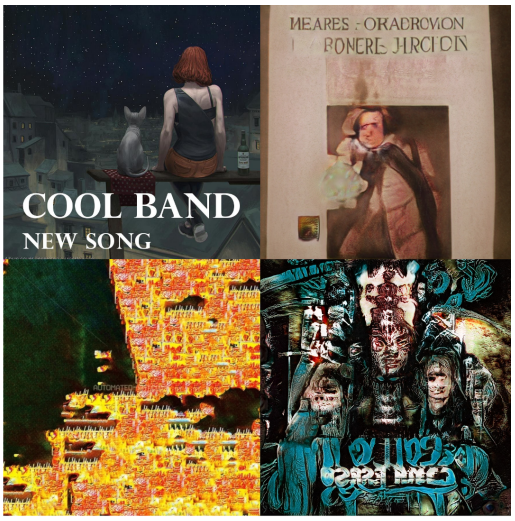

Figure 2: Automated cover generation services examples: left-top: Rocklou Cover Generator, right-top: GAN Album Art, left-bottom: The Automated Art, right-bottom: Art-Breeder.

generation is to recognize and create individual objects from a given text description. MirrorGAN paper (Qiao et al., 2019) uses the idea of learning by re-description and consists of three modules: the semantic text embedding module, global-local collaborative attentive module for cascaded image generation, and semantic text regeneration and alignment module.

Not only cGANs can generate images by condition. A well-known model is Variational Autoencoder (VAE) (Kingma and Welling, 2019). A striking representative of this approach is the NVAE model (Vahdat and Kautz, 2020), which uses depth-wise separable convolutions, residual parameterization of Normal distributions, and spectral regularization that stabilizes training.

A well-known project from OpenAI (ope, 2024) is the DALL-E model (Ramesh et al., 2021), which is a decoder-only transformer (Vaswani et al., 2017) built on GPT-3 (Brown et al., 2020) architecture and capable of generating realistic images based on provided text condition. At the same time, OpenAI has released a Contrastive Language–Image Pre-training (CLIP) model (Radford et al., 2021) that learns the relationship between the whole sentence and the image it describes, and can act as a classifier.

2.3 Vector Images Generation

The generation of vector images, in contrast to the generation of raster images, is still poorly studied. However, several papers on this topic have been recently published.

The well-known SVG-VAE (Lopes et al., 2019) and DeepSVG (Carlier et al., 2020) models are capable of generating vector images. However, they require vector supervision and need collecting large datasets of vector images, which is complicated.

DiffVG (Li et al., 2020) has become a great achievement in creating vector images. This work proposed a differentiable rasterizer for vector graphics, which bridges the raster and vector domains through backpropagation and enables gradient-based optimization. This method supports polynomial and rational curves, stroking, transparency, occlusion, and gradient fills.

In the paper (Reddy et al., 2020) the problem of differentiable image compositing was considered by proposing the DiffComp model. The authors presented a differentiable function, which composes provided discrete elements into a pattern image. Using this operator, vector graphics becomes connected with image-based losses, and it becomes possible to optimize provided elements in accordance with losses on the composited image.

Another successful work is Im2Vec (Reddy et al., 2021), which discusses synthesizing vector graphics without vector supervision. Inspired by ideas of SVG-VAE (Lopes et al., 2019) and DeepSVG (Carlier et al., 2020), authors proposed an end-to-end VAE that encodes a raster image to a latent code and then decodes it into a set of ordered closed vector paths. After that, these paths are rasterized and composited together using the DiffVG and DiffComp solutions mentioned above. An important element is the proposed path decoder, which is capable of decoding the latent code into closed Bézier curves. It becomes possible due to sampling the path control points uniformly on the unit circle, which is then deformed and transformed into final points in the absolute coordinate system of the drawing canvas. In this paper multi-resolution raster loss solves the problem when at the early stages the gradients have a small area of influence using rasterisation at multiple resolutions. The authors claim that Im2Vec shows better results than SVG-VAE and DeepSVG.

ClipDRAW (Frans et al., 2021) paper is devoted to generating vector images based on the input text. This model combines CLIP language-image encoder and DiffVG rasterizer. Initially, a set of random Bézier curves is generated, after that, they gradually transform into understandable silhouettes. Also, the model allows creating more or less realistic pictures depending on a given number of strokes.

2.4 Approaches to the Automatic Creation of a Cover for a Musical Composition

For the best of our knowledge, four services have been proposed to generate covers for musical compositions. The examples of the generated covers for each of them are presented in Fig. 2.

The Rocklou Album Cover Generator (Gavelin, 2023) service allows generating covers with resolution of 900×900 pixels specifying the name of the artist and the title of the track, and selecting the genre of the audio track. To generate the album covers, it picks one of 160 fonts and one of 1500 template images. However, the covers are only for inspiration, they are not allowed to be used commercially.

The Automated Art (aa, 2023) service allows selecting from pre-generated covers using GAN, but they contain a lot of fuzzy lines and artifacts. Two types of licensing are provided: One Time Use and Extended. Both of them limit the cover resolution to a maximum of 480000 pixels in total, permit to use the purchased media in one project only and contain many other prohibitions in use.

The GAN Album Art (Seyp, 2023) website displays a randomly selected low-resolution image from pre-generated covers, with an internal division by genre, but without any input data and without specifying a license. In addition, the generated covers contain a lot of characters from different and seemingly non-existent languages, which are impossible to read and understand.

The ArtBreeder (art, 2023) service allows creating up to 5 high-resolution images per month for free based on user-defined color preferences and random noise. All generated images are considered the public domain. However, the resulting covers contain fuzzy figures, artifacts, as well as fragments of signatures and human bodies.

In addition to the Internet services listed above, the use of GAN for generating covers of musical compositions is found in the paper (Hepburn et al., 2017). The authors report successful generation of images with a resolution of 64×64 pixels for the specified genre labels and with genre-specific visual features. It can be noted that all of these works have significant drawbacks, such as the use of generation methods with a poor diversity of results, low resolution of output images, and the presence of a large number of artifacts. Moreover, they do not analyze a music track itself. As seen from all literature reviewed, all the approaches did not solve the task to generate music cover images of acceptable quality and corresponding to music track. Therefore, we have challenged ourselves to develop such a model.

3 METHOD

The creative nature of the problem has motivated us to use unsupervised learning, and we apply the Conditional Generative Adversarial Network (cGAN) (Mirza and Osindero, 2014). In this work, music cover images are generated based on several sound features, which is a hallmark of this research. Both generator and discriminator inputs are conditioned. The condition can be an embedding of the entire music track or its fragment (Duarte et al., 2019), as well as some additional data, for instance, a track emotion indicated by a musician. It is required to train the mapping both from an audio embedding and a random vector to an output vector to find a correlation of the cover image content with the features obtained from audio data.

3.1 Input Features

We have used several common algorithms for calculating the following music features (Bogdanov et al., 2013): MFCC (Davis and Mermelstein, 1980), spectral contrast (Jiang et al., 2002), spectral peaks (pea, 2024), loudness (Skovenborg et al., 2004), danceability (Streich and Herrera, 2005), beats per minute and onset rate, mean beats volume, Chromagram (Korzeniowski and Widmer, 2016), music key and its scale. These parameters can strongly influence musical perception (Col, 2024; Freeman, 2020; Lindborg and Friberg, 2015; Tsiounta et al., 2013). For example, covers for songs with a rough voice, high volume, non-standard alarming fast rhythm or tonality with abundant musical chromaticisms, usually contain dark shades and sharp lines. At the same time, for the quiet, calm, and harmonious tracks the light and soft colors are prevailing on the covers.

The track is resampled to a frequency of 44100Hz. After that music features are calculated for 10-second track fragments with 5-second overlapping and then normalized.

Emotions in musical compositions are one-hot encoded as a vector with the length equal to the total number of emotions. The positions corresponding to the selected emotions are encoded with ones, and the rest are encoded with zeros.

3.2 CoverGAN

The noise vector, encoded track emotion selected by a musician and audio features are passed as an input

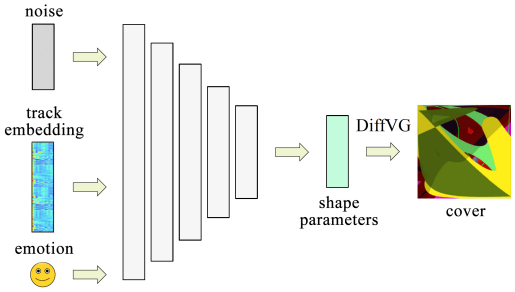

Figure 3: Fully-connected generator architecture.

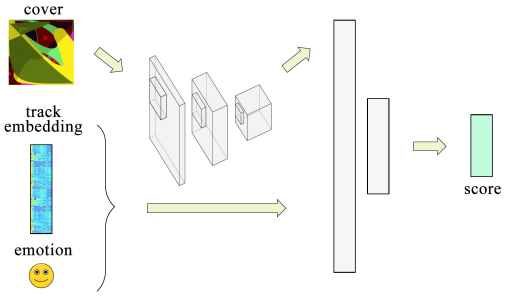

Figure 4: Discriminator architecture.

to the generator that creates a description of an image consisting of vector primitives. We have tested two architectures of the generator. The first one is presented in Fig. 3. It consists of 5 fully-connected layers with LeakyReLU with slope 0.2 for 4 layers and sigmoid activation function for the last one. As the result, output range is $(0;1)$, which allows direct setting of color channels values. The coordinates of the points are set relative to the size of the canvas. However, the actual canvas size available to the generator for each of the dimensions is twice as large as the visible one; this gives the generator the ability to place parts of shapes outside the visible area. The thickness of the outlines is predicted relative to the specified maximum. To prevent internal covariate shift and speed up learning, batch normalization with a momentum of 0.1 is applied.

The second tested architecture of the generator, consisted of a three-layer recurrent neural network with long short-term memory (LSTM) and several fully-connected layers corresponding to the optimized shape parameters: point coordinates, outline thickness, transparency, fill color, and outline color. Although this model has not yet brought an acceptable result, work on its refinement and improvement continues.

For correct error backpropagation during model training, the rasterization of vector cover images must be differentiable, which is achieved by using the diffvg library (Li et al., 2020) as the renderer. However, diffvg library cannot optimize the topology, such as adding and removing shapes, changing their order and type; therefore, the number of shapes is fixed to 3 cubic Bézier curves of 4 segments. Each Bézier curve consists of 13 points: one initial and 3 points by segment. We also create a square to represent the canvas color. For the first model the number of curves is fixed and equal to 3; for the second architecture, it corresponds to the number of embeddings of track fragments received by the generator.

The discriminator is presented in Fig. 4. The model consists of 3 convolutional and 2 fully-connected layers. Tensors of real and generated covers with 3 color channels (RGB) are provided as the input to the first convolutional layer. Real and generated covers are Gaussian blurred to prevent the influence of real covers noise on the discriminator decision. Then, the first layer outputs 24 channels, the subsequent ones – 48 each. The first fully-connected layer takes as input the result of convolution, the mood vector and embedding of a large fragment of the musical composition. The number of features of the next layer is 64 times less. The output of the network is one number – the assessment of the realism of the cover, which takes into account the compliance with audio data. LeakyReLU is the activation function on all layers, except the last one. Its slope for convolutional layers is 0.1, for fully-connected layers – 0.2. On the last layer, in accordance with the recommendation of the Wasserstein GAN (WGAN) (Arjovsky et al., 2017), the activation function is not applied. Layer normalization and normalization by elimination with a probability of 0.2 are used.

The first loss function of the discriminator is the Wasserstein loss with the gradient penalty (Gulrajani et al., 2017). Additional stimulation of the discriminator training in the correspondence of covers to the characteristics of musical compositions is provided by the secondary loss function. It is set as the difference of the average scores on the rearranged cyclic shift and on the corresponding covers from the training set batches, as follows (see Eq. 1):

$$L_2 = D(\tilde{r},a,e) - D(r,a,e), \tag{1}$$

where a is a batch of audio embeddings, e is a batch of emotion embeddings, r is a batch of real cover images, $\tilde{r}$ is a random cyclic shift of the real covers r. The cyclic shift is important, because we do not want at least one cover in a batch to remain in its original place as it might with random shuffling.

We considered the possibility of shape parameters selection in a separate (nested) GAN. Then, the external GAN would operate with vectors of the latent space of the internal one. However, despite the application of the pre-training on a Variational Autoencoder (VAE) to stabilize the training of the internal

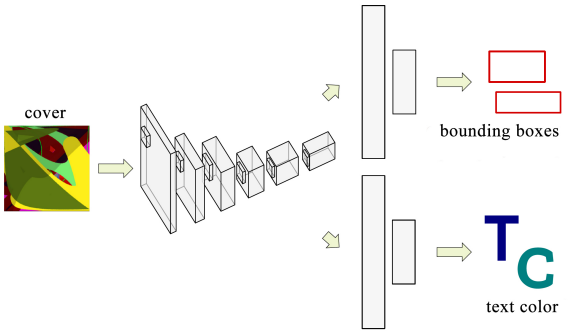

Figure 5: Captioning network architecture.

GAN, this approach led to no results. Currently, the nested GAN is not used.

After captioning (see Subsec. 3.3), the resulting description of the cover is saved in a common vector graphics format (SVG) (Cohn et al., 2000) or rasterized. The generated image of the cover does not violate the rights of the authors of the covers used while training (Gillotte, 2019). After generation, it can be licensed to the performer of the track.

3.3 Captioning

To format generated images as complete covers, we add captions with authorship information, which optimal placement, color and style are determined by an additional neural network presented in Fig. 5. It consists of 6 fully-connected layers with 3×3 and 5×5 convolutional kernels and two 2-layered sub networks, all with LeakyReLU activation with slope 0.01.

Images with four channels are fed to the network input, namely, three color channels (RGB) and the boundaries of the figures in these images, determined by the Canny edge detector (Canny, 1986). The network outputs text bounding boxes and text colors for captions. The optimal font size is selected accounting the predicted bounding rectangles. The font family is currently selected randomly from Google Fonts.

The training dataset is described in details in section 4, it consists of original covers with text captions removed using a graphic editor. For training the captioning network, each image was labelled with text bounding boxes and colors. The labelling was partially automated by comparing the edited covers with the original ones.

4 DATASET

The advantages provided by vector graphics in the field of scalability come at the cost of disadvantages: not every bitmap image can be rationally represented as a set of vector primitives. Thus, if the cover, for example, is a detailed photo, an attempt to vectorize and enlarge such a cover leads to a noticeable change in style. In this regard, it is reasonable to make a training dataset from covers that can be reproduced by a generator that draws vector primitives (basic geometric shapes and curves with strokes and fills). In addition, to better match the covers to musical compositions, it is preferable to collect a dataset from singles or mini-albums, where the album cover obviously refers to one of the tracks.

These limitations determine the need to collect a new dataset. In accordance with the listed requirements, a set of 1500 musical compositions of various genres was collected. With the help of the Adobe Photoshop graphic editor (pho, 2024), the captions were removed from the covers. On the crowdsourcing platform Yandex Toloka (tol, 2024) the tracks were labeled with emotions and included in the dataset. The label contains 2-3 emotions of a musical composition from the following list: comfortable, happy, inspirational, joy, lonely, funny, nostalgic, passionate, quiet, relaxed, romantic, sadness, soulful, sweet, serious, anger, wary, surprise, fear. We can not release the dataset with music cover images due to copyright; however, its part containing 1500 objects in the following format: track author – track title, 2-3 emotions from the list, is publicly available[1].

Furthermore, to obtain additional information about tracks (genres, release date, artist attributes, and others) was prepared a program that extracts available metadata directly from audio files and supplements them with information from public information databases (Wikidata (wik, 2024) and MusicBrainz (mus, 2024)). Track metadata is not currently used by the algorithm, but it is possible to do in the future.

After all, the collected dataset contains a number of covers that is not large enough to train deep learning models. Thus, to enlarge the resulting dataset, augmentation was applied; operations such as horizontal flip and 90 degrees clockwise and counterclockwise rotation were applied to the covers.

5 RESULTS

Unlike the existing solutions, which generates covers that do not take into account musical compositions, the suggested model allows using extracted sound

[1] https://www.kaggle.com/datasets/viacheslavshalamov/music-emotions

features in a generative algorithm. The created covers contain titles with information about authorship and track name and are available in a common vector graphics format without additional licensing restrictions on components. This allows us to speak about their compliance with generally accepted requirements for the design of musical compositions releases.

5.1 Training

Training of neural networks was performed on the capacities of the Google Colaboratory (goo, 2024) service using the CUDA computing architecture. To achieve stable training of the model, various hyperparameters were tried, including different network architectures, the number of layers in models, learning rate, optimization algorithms, gradient penalty coefficient and the number of steps to update discriminator. The final version uses the generator consisting of fully-connected layers, a learning rate coefficient of 0.0005 with a batch size of 64, a five-time repetition of the discriminator training and a canvas size of 128×128 pixels. The training took 7200 epochs based on a manual assessment of the quality of the generated images.

The Adam optimization algorithm (Kingma and Ba, 2015) was chosen for CoverGAN training. Its gradients decay rate control coefficient β_1 was set to 0.9 and for the second moments of gradients $\beta_2 = 0.999$.

To train the captioning network, batch normalization with a following hyperparameters is used: a momentum of 0.1 and normalization by elimination with a probability of 0.2. The Adam algorithm with gradients decay rate control coefficient $\beta_1 = 0.5$ and for the second moments of gradients $\beta_2 = 0.999$ is chosen as the optimization algorithm.

During the training of the captioning network, 256×256 pixel images and a batch size of 64 are used to select the design of signatures. The average quadratic error is used as a loss function. For validation, the Generalized Intersection over Union (GIoU) (Rezatofighi et al., 2019) metric is calculated and averaged for the entire dataset. In order to maximize it, training lasts 138 epochs. The value of GIoU metric of 0.65 has been achieved for the arrangement of rectangles; the average error value for colors on the entire training set reaches about 0.00014 at the end of training.

Table 1: Comparison of the covers generated by the proposed CoverGAN with DALL-E for titles denoted as DALL-E^t, DALL-E for lyrics denoted as DALL-E^l, AttnGAN^l for lyrics, and AttnGAN for lyrics fine-tuned on covers denoted as AttnGAN^t + covers. 'All score' column indicates the normalized score given by all assessors. 'Musicians' column indicates the normalized assessment score given by assessors identified themselves as musicians.

Method	All score	Musicians
DALL-E^t	0.34 ± 0.03	0.31 ± 0.05
DALL-E^l	0.3 ± 0.03	0.26 ± 0.05
AttnGAN^l	0.44 ± 0.04	0.23 ± 0.06
AttnGAN^l + covers	0.36 ± 0.03	0.19 ± 0.04
CoverGAN (Ours)	0.68 ± 0.05	0.71 ± 0.05

5.2 Comparison with Text-to-Image Generation

To assess the effectiveness of the proposed algorithm, we tested it on musical compositions not included in the training set. As a result, for different musical compositions, the algorithm created significantly different covers (Figure 1).

Due to the lack of alternative algorithms that rely on musical features for cover creation, we used images generated by the AttnGAN (Xu et al., 2018) and DALL-E (rud, 2024) based on lyrics for the following quality comparison. AttnGAN takes as input text of arbitrary length, when more modern approaches, such as DALL-E, focus on short texts (up to 200 symbols) that are much shorter than a song lyrics. Thus, we also assess the quality of DALL-E-generated-images by query: 'Cover for the track `<title> <lyrics>`'. Furthermore, we fine-tuned AttnGAN model initially trained on COCO dataset on our dataset of vector covers.

We conducted a survey asking participants to listen to 15 popular musical compositions and rate the covers generated by the proposed algorithm and AttnGAN on a scale of 1 to 5 (1 stands for completely inappropriate, 5 stands for the perfect fit). 110 assessors aged 16 to 40 years took part in an anonymous survey, 24 of them identified themselves as musicians. Normalized survey results are presented in Tab. 1. The examples of the generated images are shown in Fig. 6. As it can be seen from the figure, DALL-E generates unknown symbols and undetermined shapes, AttnGAN generates patterns, whereas shapes generated by our CoverGAN are very simple.

The text-to-image generation models are very sensitive to the input text string, which often does not describe the entire track, and sometimes even contradicts the general mood of the song. Therefore, the

Figure 6: Generated music cover images using different models; each row corresponds to real music track.

resulting images do not often correspond to music.

Although the AttnGAN^t + covers model generates images with a monotonous background, the objects depicted have incomprehensible outlines and artifacts. Moreover, the generated images can have grid pattern. These are probably the reasons that assessors rated this model worse than the previous one, as it can be seen from the table.

The CoverGAN score means that the generated cover images are quite satisfactory, however, further work in this direction is reasonable.

5.3 Different Genres

To assess the significance of the patterns found by the GAN model comparing sound and the generated covers, another survey was conducted using the crowdsourcing project Yandex Toloka. Its participants were asked to listen to 10 musical compositions from various genre categories and choose the most suitable of the two generated covers for each. One of the covers was created by the generator directly based on the specified track, the other – based on a track from another category with the replacement of the caption text.

According to the results based on the assessments collected from 110 listeners, the probability of the respondents, who have chosen the cover created by the generator for the track, was 0.76 ± 0.03. This score confirms the presence of significant differences between the proposed covers for listeners and indicates a moderate tendency of survey participants to agree with the conclusions of the generator. At the same time, there is a need for further improvement of the generative algorithm, including to increase the diversity of generated images.

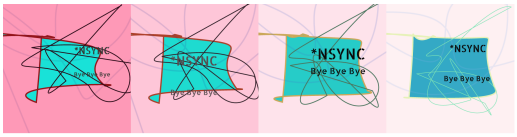

Figure 7: The results of running the algorithm on several versions of the same musical composition with different volume levels (from left to right volume level: 100%, 75%, 50%, 25%).

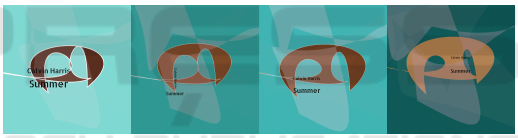

Figure 8: The dependence of the cover on the specified emotion of the track (from left to right: comfortable, passionate, relaxed, wary).

5.4 Qualitative Analysis

To check the generalization ability of the CoverGAN, we analyzed the results of running the algorithm on musical compositions, which are not included in the training set. It was revealed that the algorithm creates significantly different covers for different tracks (see Fig. 1). By changing the noise vector, we achieve the generation of alternative images to provide the user with a choice.

For individual sound features, it can be traced how the generated cover image changes with a gradual change in the feature (for example, for the same musical composition at different volume levels as in Fig. 7).

Moreover, user-specified emotions have a significant impact on the generated covers. Figure 8 shows the change of the cover when the specified emotion changes.

Finally, on the test dataset, it was found that the covers of similar tracks also have some similarities.

Figure 9: Covers generated for lullabies and rock music.

Therefore, for several lullabies, the algorithm suggested covers of light tones, while the covers generated for heavy metal genre turned out to be dark (Fig. 9). Such connections are not universal, although their presence is consistent with the conclusions of the work (Hepburn et al., 2017). In the future, such correspondences can be used to detect and analyze patterns found by a trained generative network.

6 LIMITATIONS

The major limitation is the simplicity of generated paths. They usually do not have any regular shape such as a circle or a square. Moreover, it is difficult for CoverGAN to generate objects with visual semantics (i.e., person images, natural scenes). The generated images are abstract and cannot cover all scenarios in art design.

The generated covers are diverse and often quite well suited to musical compositions. Nevertheless, it is not always easy to interpret the connections taken into account by the generator for choosing the shape parameters. Sometimes, the covers of two completely different tracks may turn out to be visually similar. It is assumed that in order to obtain more stable and explicable patterns in the generation of covers, it is necessary to enlarge the dataset, as well as significantly complicate the generator architecture used. The basis of such a model can be an implemented recurrent generator, modified to transform a sequence of musical fragments embeddings into a sequence of figures, for example, using the transformer architecture (Vaswani et al., 2017). However, such developments require preliminary theoretical research in the field of vector image generation.

For the additional network engaged in the creating of captions, the main difficulty in practice is to ensure that the color is sufficiently contrasting with the background. In real covers, artists often use additional visual effects (shadows, contrasting strokes, glow, background fill) to achieve matching of slightly contrasting colors; a similar approach may be implemented for this solution in the future. Another difficult task is to wrap text inside the bounding box estimating the best number of lines. Currently, this is not provided and the text may not fit the canvas completely. In the future, it is supposed to use the functionality of the HarfBuzz (har, 2024) text generation library or the Pango (pan, 2024) rendering library.

7 CONCLUSION

There exist music services offering musicians computer-generated images as cover images for their musical compositions; however, they do not consider the music track itself. In this work, we have developed the CoverGAN model to generate vector images conditioned by a music track and its emotion. We have compared the model suggested to AttnGAN and DALL-E models for text-to-image generation. The reported results prove that the proposed algorithm is competitive in the task of music cover synthesis. This is also good evidence that vector graphics synthesis is a promising research direction. The approach applied is limited only with the parametric space where the image is located. In the future, it is planned to generate Bézier curves with an arbitrary number of segments based on an approach similar to Im2Vec (Reddy et al., 2021), as well as potentially some concrete shapes (natural scene, human silhouette).

ACKNOWLEDGEMENTS

The research was supported by the ITMO University, project 623097 ”Development of libraries containing perspective machine learning methods”

REFERENCES

(2023). Artbreeder. https://www.artbreeder.com/compose/albums. Accessed: 2023-01-15.

(2023). Automated art. https://automated-art.co.uk/. Accessed: 2023-01-15.

(2024). Adobe photoshop. https://www.adobe.com/products/photoshop.html. Accessed: 2024-01-07.

(2024). Color-music-association. https://prezi.com/4asqw6p_wbpl/color-music-association/. Accessed: 2024-01-07.

(2024). Essentia. https://essentia.upf.edu/algorithms_reference.html. Accessed: 2024-01-07.

(2024). Google colaboratory. https://colab.research.google.com. Accessed: 2024-01-07.

(2024). Harfbuzz. https://harfbuzz.github.io/. Accessed: 2024-01-07.

(2024). Librosa. https://librosa.org/doc/latest/feature.html. Accessed: 2024-01-07.

(2024). Musicbrainz. https://musicbrainz.org. Accessed: 2024-01-07.

(2024). Openai. https://openai.com/. Accessed: 2024-01-07.

(2024). Pango. https://pango.gnome.org/. Accessed: 2024-01-07.

(2024). Ru dall-e. https://rudalle.ru/. Accessed: 2024-01-07.

(2024). Spectral peaks. https://ccrma.stanford.edu/~jos/parshl/Peak_Detection_Steps_3.html. Accessed: 2024-01-07.

(2024). Wikidata. https://www.wikidata.org. Accessed: 2024-01-07.

(2024). Xml documentation. https://www.w3.org/TR/xml/. Accessed: 2024-01-07.

(2024). Yandex toloka. https://toloka.ai/. Accessed: 2024-01-07.

Arjovsky, M., Chintala, S., and Bottou, L. (2017). Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR.

Bogdanov, D., Wack, N., Gómez, E., Gulati, S., Herrera, P., Mayor, O., Roma, G., Salamon, J., Zapata, J. R., and Serra, X. (2013). Essentia: An audio analysis library for music information retrieval. In *ISMIR*.

Bond-Taylor, S., Leach, A., Long, Y., and Willcocks, C. G. (2021). Deep generative modelling: A comparative review of vaes, gans, normalizing flows, energy-based and autoregressive models. *arXiv preprint arXiv:2103.04922*.

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. *CoRR*, abs/2005.14165.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence*, (6):679–698.

Carlier, A., Danelljan, M., Alahi, A., and Timofte, R. (2020). Deepsvg: A hierarchical generative network for vector graphics animation. *CoRR*, abs/2007.11301.

Choi, K., Fazekas, G., Cho, K., and Sandler, M. B. (2017). A tutorial on deep learning for music information retrieval. *CoRR*, abs/1709.04396.

Cohn, R., Dodds, D., Donoho, A. W., Duce, D. A., Evans, J., Ferraiolo, J., Furman, S., Graffagnino, P., Graham, R., Henderson, L., Hester, A., Hopgood, B., Jolif, C., Lawrence, K. R., Lilley, C., Mansfield, P., McCluskey, K., Nguyen, T., Sandal, T., Santangeli, P., Sheikh, H. S., Stevahn, R. E., and Zhou, S. (2000). Scalable vector graphics svg 1.0 specification.

Cook, K. (2013). Music industry market research-the effect of cover artwork on the music industry.

Davis, S. and Mermelstein, P. (1980). Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences. *IEEE transactions on acoustics, speech, and signal processing*, 28(4):357–366.

Duarte, A., Roldan, F., Tubau, M., Escur, J., Pascual, S., Salvador, A., Mohedano, E., McGuinness, K., Torres, J., and Giro-i Nieto, X. (2019). Wav2pix: Speech-conditioned face generation using generative adversarial networks. In *ICASSP*, pages 8633–8637.

Frans, K., Soros, L. B., and Witkowski, O. (2021). Clipdraw: Exploring text-to-drawing synthesis through language-image encoders. *CoRR*, abs/2106.14843.

Freeman, S. (2020). Musical variables and color association in classical music. *Journal of Student Research*, 9(1).

Gao, R., Oh, T.-H., Grauman, K., and Torresani, L. (2020). Listen to look: Action recognition by previewing audio. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10457–10467.

Gavelin, D. (2023). Rocklou album cover generator. https://www.rocklou.com/albumcovergenerator. Accessed: 2023-01-15.

Gillotte, J. L. (2019). Copyright infringement in ai-generated artworks. *UC Davis L. Rev.*, 53:2655.

Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., and Bengio, Y. (2014). Generative adversarial nets. *Advances in neural information processing systems*, 27.

Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., and Courville, A. (2017). Improved training of wasserstein gans. *arXiv preprint arXiv:1704.00028*.

Hepburn, A., McConville, R., and Santos-Rodrıguez, R. (2017). Album cover generation from genre tags. In *10th International Workshop on Machine Learning and Music*.

Hertzmann, A. (2020). Visual indeterminacy in gan art. *Leonardo*, 53(4):424–428.

Jiang, D.-N., Lu, L., Zhang, H., Tao, J., and Cai, L. (2002). Music type classification by spectral contrast feature. *Proceedings. IEEE International Conference on Multimedia and Expo*, 1:113–116 vol.1.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Kingma, D. P. and Welling, M. (2019). An introduction to variational autoencoders. *arXiv preprint arXiv:1906.02691*.

Korzeniowski, F. and Widmer, G. (2016). Feature learning for chord recognition: The deep chroma extractor. In *ISMIR*.

Li, T.-M., Lukáč, M., Michaël, G., and Ragan-Kelley, J. (2020). Differentiable vector graphics rasterization for editing and learning. *ACM Trans. Graph. (Proc. SIGGRAPH Asia)*, 39(6):193:1–193:15.

Li, W., Zhang, P., Zhang, L., Huang, Q., He, X., Lyu, S., and Gao, J. (2019). Object-driven text-to-image synthesis via adversarial training. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12166–12174.

Lindborg, P. and Friberg, A. K. (2015). Colour association with music is mediated by emotion: Evidence from an experiment using a cie lab interface and interviews. *PloS one*, 10(12):e0144013.

Lopes, R. G., Ha, D., Eck, D., and Shlens, J. (2019). A learned representation for scalable vector graphics. *CoRR*, abs/1904.02632.

Medel, I. L. (2014). The death and resurrection of the album cover. *index.comunicación*, 4(1):37–57.

Mirza, M. and Osindero, S. (2014). Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*.

Moffat, D., Ronan, D., and Reiss, J. D. (2015). An evaluation of audio feature extraction toolboxes.

Oh, T.-H., Dekel, T., Kim, C., Mosseri, I., Freeman, W. T., Rubinstein, M., and Matusik, W. (2019). Speech2face: Learning the face behind a voice. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7539–7548.

Oord, A. v. d., Kalchbrenner, N., Vinyals, O., Espeholt, L., Graves, A., and Kavukcuoglu, K. (2016). Conditional image generation with pixelcnn decoders. *arXiv preprint arXiv:1606.05328*.

Qian, R., Di Hu, H. D., Wu, M., Xu, N., and Lin, W. (2020). Multiple sound sources localization from coarse to fine. *arXiv preprint arXiv:2007.06355*.

Qiao, T., Zhang, J., Xu, D., and Tao, D. (2019). Mirrorgan: Learning text-to-image generation by redescription. *CoRR*, abs/1903.05854.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., and Sutskever, I. (2021). Learning transferable visual models from natural language supervision. *CoRR*, abs/2103.00020.

Ramesh, A., Pavlov, M., Goh, G., Gray, S., Voss, C., Radford, A., Chen, M., and Sutskever, I. (2021). Zero-shot text-to-image generation. *CoRR*, abs/2102.12092.

Reddy, P., Gharbi, M., Lukac, M., and Mitra, N. J. (2021). Im2vec: Synthesizing vector graphics without vector supervision. *arXiv preprint arXiv:2102.02798*.

Reddy, P., Guerrero, P., Fisher, M., Li, W., and Mitra, N. J. (2020). Discovering pattern structure using differentiable compositing. *ACM Transactions on Graphics (TOG)*, 39(6):1–15.

Rezatofighi, S. H., Tsoi, N., Gwak, J., Sadeghian, A., Reid, I. D., and Savarese, S. (2019). Generalized intersection over union: A metric and a loss for bounding box regression. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 658–666.

Schedl, M., Gómez Gutiérrez, E., and Urbano, J. (2014). Music information retrieval: Recent developments and applications. *Foundations and Trends in Information Retrieval. 2014 Sept 12; 8 (2-3): 127-261.*

Seyp, V. (2023). Gan album art. https://ganalbum.art/. Accessed: 2023-01-15.

Skovenborg, E., Quesnel, R., and Nielsen, S. H. (2004). Loudness assessment of music and speech. *Journal of The Audio Engineering Society*.

Streich, S. and Herrera, P. (2005). Detrended fluctuation analysis of music signals danceability estimation and further semantic characterization. In *In Proceedings of the AES 118th Convention*.

Tsiounta, M., Staniland, M., and Patera, M. (2013). Why is classical music yellow: A colour and sound association study. In *AIC 2013-12th Congress of the International Colour Association*.

Vahdat, A. and Kautz, J. (2020). Nvae: A deep hierarchical variational autoencoder. *ArXiv*, abs/2007.03898.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Xu, T., Zhang, P., Huang, Q., Zhang, H., Gan, Z., Huang, X., and He, X. (2018). Attngan: Fine-grained text to image generation with attentional generative adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1316–1324.

Zhang, H., Xu, T., Li, H., Zhang, S., Wang, X., Huang, X., and Metaxas, D. N. (2017). Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 5907–5915.