MuGAN - Generating Adversarial Modified Music

Ankush Pratap Singh
Dept. of ECE
NYU Tandon
ax2047@nyu.edu

Monisha Gnanaprakasam
Dept. of ECE
NYU Tandon
mg6955@nyu.edu

Vaishnavi Rajput
Dept. of ECE
NYU Tandon
vr2229@nyu.edu

Abstract

The Oxford dictionary defines **music** as sounds that are arranged in a way that makes listening enjoyable or exciting. Having good taste in music or being able to play a musical instrument is a skill. Unfortunately, not everyone is endowed with this skill since not every audio signal can be labeled as a music. In this work, we are attempting to allow one to generate music of different emotions or genres from existing music of various duration without being subject to a copyright claim as it easier for an amateur to create music from an existing music rather than create a new one. The **GAN** architecture is one of the most commonly used approaches for creating novel, synthetic instances of data that can be passed off as real data at first glance. We have developed an architecture based on GANs that trains on music of different genres in order to generate music associated with those genres. While the adversarial modified music has a longer duration than the one liable for a copyright claim, it cannot be considered theft because it has been modified into a different genre.

1 Introduction

There are numerous practical applications of synthesizing audio for specific domains in the field of creative sound design for music and film. It is common practice for musicians and Foley artists to search large databases of music for specific recordings that are appropriate for specific situations. This strategy is labor intensive and may result in a negative outcome if the ideal sound effect does not exist in the library or if it is legally not allowed to be used due to Copyright Law [14]. It may be more appropriate for a sound artist to take a broad approach to locate the types of sounds they are seeking (e.g. footsteps) and to make small adjustments to latent variables to fine-tune them (e.g. a large boot lands on a gravel path). Audio signals, however, have a high temporal resolution, and strategies that learn such representations must be able to operate effectively on high-dimensional data sets.

One such unsupervised strategy is the use of Generative Adversarial Networks (GANs) (Goodfellow et al., 2014). Audio synthesis based on GANs may offer numerous advantages. The first advantage of GANs is that they could be used to augment data in systems that require a great deal of data. Secondly, GANs can be used to sample large amounts of audio rapidly and easily. And most importantly, this provides an escape from copyright infringement acts that prohibit using other composers' music for more than 30 seconds. While the usefulness of generating static images from GANs could be questioned, generating sound effects is an immediate benefit for many applications (e.g. mediation camps, health camps etc).

1.1 Music Information Retrieval

Directly working on mp3 or wav audio files for computation is not feasible, hence it becomes important to convert it to a form which can be understood by our model like images and pixel intensities. So, it becomes necessary to work on the spectrogram or MFCC of an audio.

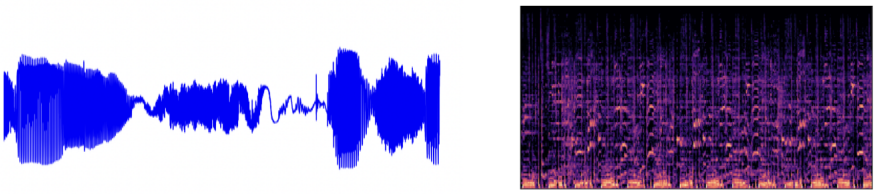

Figure 1: A second of generated speech(left), Sample Transformed image of an audio wave(right)

The purpose of Music Information Retrieval (MIR), as its name suggests, is to retrieve information from music. Various characteristics of audio are used for this purpose, such as wave forms (amplitudes over time), spectrograms, sample rates, etc. Industries and academicians use MIR to categorize, manipulate, or create music.

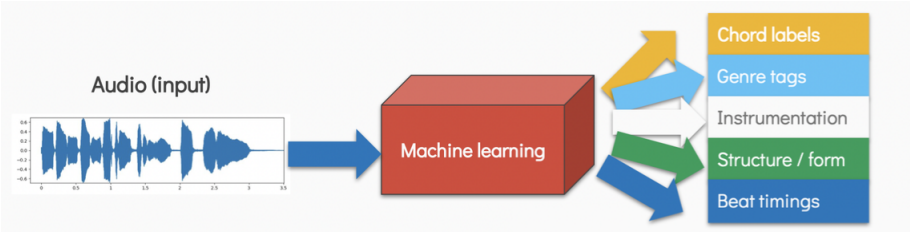

Figure 2: Semantic Representation of a simple audio-based MIR

The nominal Machine Learning strategy involves collecting paired examples of input and output and training the model. The quality and quantity of the dataset are always limiting factors.

1.2 Generative Adversarial Networks

GANs learn mappings from low-dimensional latent vectors $z \in Z$, i.i.d. samples from known prior P_Z , to points in the space of natural data X . In their original formulation, a generator G : Z $\rightarrow$ X is pitted against a discriminator D : X $\rightarrow$ [0, 1] in a two-player mini max game. G is trained to minimize the following value function, while D is trained to maximize it:

V (D, G) = $E_{x \sim P_X}$ [log D(x)] + $E_{z \sim P_Z}$ [1 - log D(G(z))]

In other words, D is trained to determine if an example is real or fake, and G is trained to fool the discriminator into thinking its output is real.

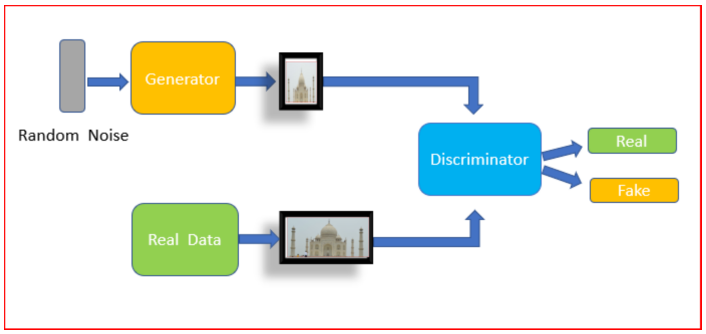

Figure 3: GANs block diagram (taken from https://www.analyticsvidhya.com/blog/2021/04/generate-your-own-dataset-using-gan/

1.3 Related Work

Recent studies (van den Oord et al., 2016; Mehri et al., 2017) have demonstrated that neural networks can be trained to operate on raw audio with auto-regression. The advantage of such approaches is that they do not require the use of engineered feature representations. The auto-regressive setting, however, results in slow generation since output audio samples must be fed back into the model one at a time rather than in batches like with GANs.

As far as we are aware, GANs have been used (kowski et al., 2019; Engel et al., 2019; Kumar et al., 2019) to generate speech signals and music from a particular particular instrument, but little has been done in order to generate genre-based music, especially Indian music.

2 Methodology

For our final architecture, we used a Deep Convolution based GAN which has a discriminator and a generator network. We trained our model on multiple epochs for a particular music genre making our discriminator train to classify whether an audio signal correspond to that particular music genre or not. Additionally, we kept improving our generator network which is fed random noise as an input.

Once trained on several epochs or when the error went below a particular threshold , we stopped training and used the generator network to produce some music based on some input.

Our generator network now knows how to modify a particular music to match it with the genre it was trained on. We take advantage of this and provide music of any genre of length 30 seconds (more than copyright claim threshold) and adversarial modify the original music.

2.1 Music images

In order to convert a song to its corresponding spectrograms, we used libroasa library with song duration as 30 seconds. Librosa library has standardized DSP tools in Python for music and audio analysis.

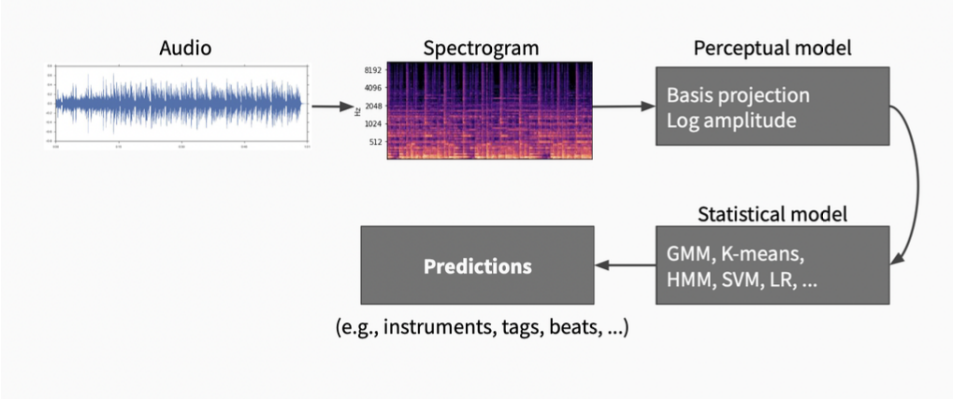

Figure 4: Librosa based MIR

2.2 GANs

For our discriminator network in GAN, we included a 10 layered network having conv2d with ReLu activation as our non linear activation function and normalized it across our batch. The discriminator is fed original images (spectrograms) having dimension 3 (number of channels) X 556 X 556.

As our generator network, we made another network consisting 10 layers having ConvTranspose2d with Relu as our activation function, normalized across the batch. The generator is fed random noise of size 3 (number of channels) X 64 X 64.

2.3 Data Augmentation

In the real world scenario, we may have a dataset with limited set of relevant data. As in our case, it was tough finding datasets having well defined and labeled genre wise songs collection. So, data augmentation played a crucial part.

Data augmentation helps in creating synthetically modified data relevant data. We took help from of data augmentation by using Random horizontal flip, cropping, and normalizing the images. This helped us model better

3 Experiment

For working on our proposed architecture, we used NYU HPC machines and local training to conduct this project. TensorFlow, Librosa, IPython, PIL, and other important python libraries.

3.1 Dataset

Finding the correct dataset was a challenge and after careful considerations, we used a combination of 3 datasets **GACMIS (Genre Automated Classification using Machine Learning of Indian Songs)** consisting of 6 genres Indian songs each having 100 songs, **Indian Music Genre** Dataset (Kaggle) consisting of 5 genres Indian songs each having 100 songs, **FMA (Free Music Archive)** consisting of 8 gneres English songs each having 1000 songs, but since we are training mainly of Indian Hindi songs, we have used just a small amount of data from FMA dataset.

3.2 Music Information Retrieval

After segregating songs based on different genres, we converted them to their corresponding images (spectrograms).

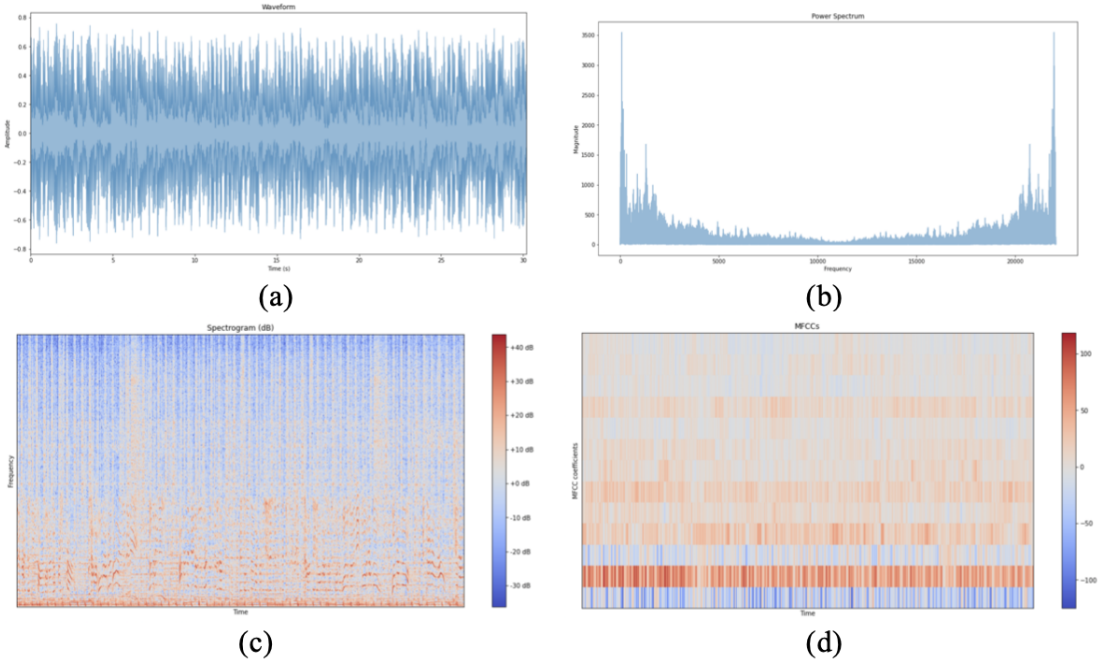

Figure 5: a) A sample Wave plot (Amplitude vs Time) b) Power spectrum of the same waveform (Magnitude vs Frequency) c) Spectrogram of the same waveform (Frequency vs Time) d) MFCCs (MFCC coefficients vs Time)

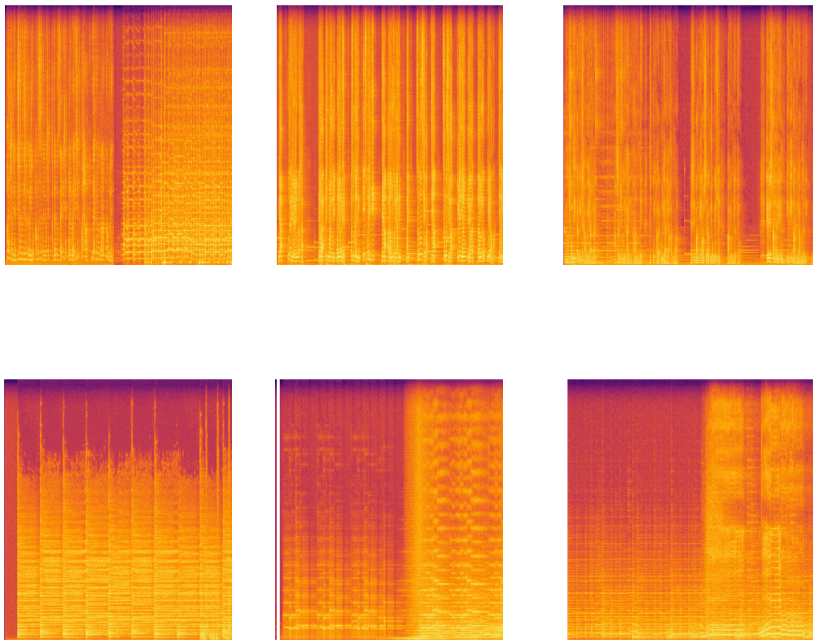

Figure 6: Six random spectrograms on from each genre (a) Bhojpuri, (b) Garhwali, (c) Ghazal, (d) Rap, (e) Romantic, (f) Sufi

3.3 GANs Architecture

Our GAN network uses a UNet-styled architecture for the Generator network and a convolution-based network for the Discriminator network. It is a non-autoregressive model with a full convolutional structure that is well suited for the inversion of unknown Mel-spectrograms (Melody Spectrograms).

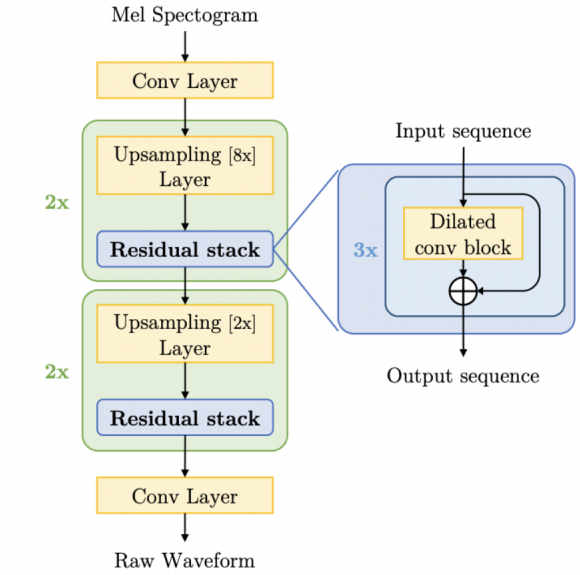

Figure 7: Generator Architecture

A generator architecture is composed of a stack of transposed convolutional layers in order to upsample the input sequence. Each transposed convolutional layer is followed by a stack of residual layers (Figure 7).

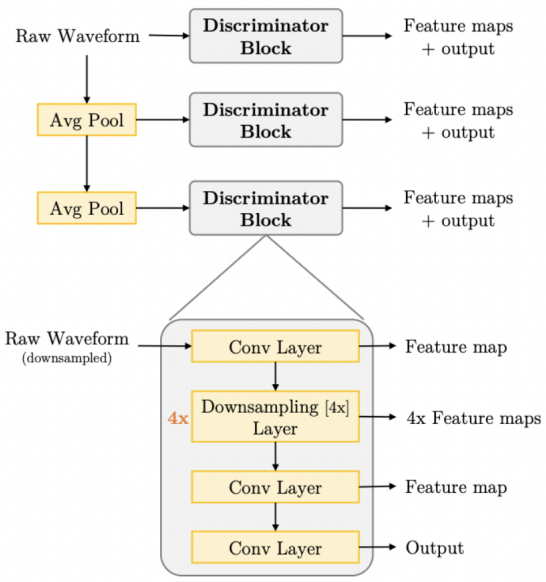

Figure 8: Discriminator Architecture

The entire architecture of the discriminator consists of three discriminators, which operate on three audio scales, namely the original audio scale, a 2X downsampled audio scale, and a 4X downsampled audio scale. In general, each discriminator is biased so that it will learn features that are pertinent to different frequency ranges within the audio signal (Figure 8).

3.4 GANs Training

As a training method, we used the hinge loss formulation (Lim Ye, 2017; Miyato et al., 2018) to train our GANs.

$$\min_{D_k} \mathbb{E}_x \Big[\min(0, 1 - D_k(x)) \Big] + \mathbb{E}_{s,z} \Big[\min(0, 1 + D_k(G(s, z))) \Big], \ \forall k = 1, 2, 3$$

$$\min_{G} \mathbb{E}_{s,z} \left[\sum_{k=1,2,3} -D_k(G(s, z)) \right]$$

Moreover, we also use a feature-matching objective (Larsen et al., 2015) to help train our generator in a more efficient manner.

$$\mathcal{L}_{\text{FM}}(G, D_k) = \mathbb{E}_{x,s \sim p_{\text{data}}} \left[\sum_{i=1}^{T} \frac{1}{N_i} ||D_k^{(i)}(x) - D_k^{(i)}(G(s))||_1 \right]$$

Overall the Generator loss becomes

$$\min_{G} \left(\mathbb{E}_{s,z} \left[\sum_{k=1,2,3} -D_k(G(s, z)) \right] + \lambda \sum_{k=1}^{3} \mathcal{L}_{\text{FM}}(G, D_k) \right)$$

4 Results

Our model has been trained on a variety of genres of music. Depending on the genre, we receive a set of weights that can be loaded onto the generator model to generate music that is characteristic of that genre. Our training consisted of 6 genres in total, so we were able to generate 6 different sets of weights for the generator network as a result.

Below is an example of a generated and source music spectrogram.

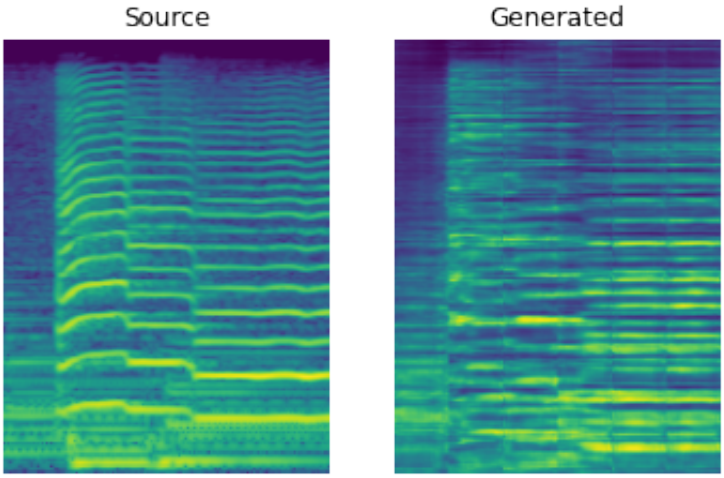

Figure 9: A Ghazal trained source music and generated music spectrogram

Having trained the model on 50 epochs, we can see that the model has begun to capture the essence of the genre in a very accurate manner.

In spite of the fact that this model can work on random noise, it is also capable of transferring the genre style of existing music as well. In case a song has a duration of over 30 seconds, this can very well help to avoid copyright issues.

It is also worth noting that this model is also capable of converting songs of any length. Unlike a cropped mp3 file, it does not require cropping because it converts each second of the song separately, thereby it can handle mp3 files of various lengths.

We have uploaded some samples on YouTube as well, including the original song, along with the newly transferred genre song, as well.

Link: https://www.youtube.com/playlist?list=PLHiou0RPL9WpsI25u-yCeSAfYMpKSphgm

GitHub link to the repo: https://github.com/ankushpratap95/gan-based-music-genre-transfer

5 Conclusions

Our work presents a GAN-based architecture that is tailored for music genre transfer using Bollywood (Indian) music datasets. Using the proposed method, we demonstrated qualitative results that proved the effectiveness and generality of the method presented in the paper. Among the advantages of our model are its ability to work with a variety of song lengths, and its ineligibility for the claim of copyright infringement. Our idea is that our generator can be used as a plug-and-play replacement for style transfer tasks related to music genres in the near future.

In spite of the fact that the model is producing reasonable results, there is still a lot of work to be done when it comes to improving and fine-tuning the audio that is generated. In order to achieve this, training with better hyper-parameters, more number of songs and for longer duration will be required, but we believe this is a positive step in the direction of transferring music styles, especially Indian music without actually copying the music which would be liable to be easily disputed by the owner of the music.

6 References

[1] **Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio**, "Generative Adversarial Nets", arXiv:1406.2661v1 [stat.ML] 10 Jun 2014

[2] **Aa ron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu**, WaveNet: A generative model for raw audio, arXiv:1609.03499, 2016.

[3] **Soroush Mehri, Kundan Kumar, Ishaan Gulrajani, Rithesh Kumar, Shubham Jain, Jose Sotelo, Aaron Courville, and Yoshua Bengio**, SampleRNN: An unconditional end-to-end neural audio generation model, In ICLR, 2017.

[4] **Mikołaj Bin kowski, Jeff Donahue, Sander Dieleman, Aidan Clark, Erich Elsen, Norman Casagrande and Luis C. Cobo, Karen Simonyan**, HIGH FIDELITY SPEECH SYNTHESIS WITH ADVERSARIAL NETWORKS, arXiv:1909.11646v2 [cs.SD] 26 Sep 2019

[5] **Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue and Adam Roberts**, GANSynth: Adversarial Neural Audio Synthesis, In ICLR 2019

[6] **Hao-Wen Dong, Wen-Yi Hsiao, Li-Chia Yang andYi-Hsuan Yang**, "MuseGAN: Multi-track Sequential Generative Adversarial Networks for Symbolic Music Generation and Accompaniment", arXiv:1709.06298v2 [eess.AS] 24 Nov 2017

[7] **Victor Sim**. "MuseGAN: Using GANs to generate original Music", Source https://towardsdatascience.com/bachgan-using-gans-to-generate-original-baroque-music-10c521d39e52

[8] **Eerola, Tuomas and Vuoskoski, Jonna**, "A comparison of the discrete and dimensional models of emotion in music", Psychology of Music. 10.1177/0305735610362821. January 2011

[9] **M. Zentner, D. Grandjean, and K. R. Scherer**, "Emotions evoked by the sound of music:Characterization, classification, and measurement", Psychology of Music:Emotion Vol.8 No.4: 494–521.2008.

[10] **Sawan** (Jun 14, 2021). Genre-Classification-using-Deep-learning/. Source Code. https://github.com/sawan16/Genre-Classification-using-Deep-learning . Jun 14, 2021

[11] **Michael Defferreard** (Sept 6, 2021). fma/. Source Code. https://github.com/mdeff/fma

[12] **Aayush Bhaskar** (April 10, 2021). Source Code. https://www.kaggle.com/winchester19/indian-music-genre-dataset . April 10, 2021

[13] **Ujjwal**, GACMIS/Dataset/. Source Code. https://github.com/ujjwalll/GACMIS/tree/master/Dataset . Dec 20, 2020

[14] **Copyright Law** https://www.copyright.gov/help/faq/definitions.html

[15] **Jae Hyun Lim and Jong Chul Ye** . "Geometric gan", arXiv:1705.02894. 8 May, 2017

[16] **Anders Boesen Lindbo Larsen, Søren Kaae Sønderby, Hugo Larochelle and Ole Winther** . "Autoencoding beyond pixels using a learned similarity metric", arXiv preprint arXiv:1512.09300, 2015. 10 Feb, 2015

[17] **Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brebisson, Yoshua Bengio and Aaron Courville** . "MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis", arXiv:1910.06711, 2019. 8 Oct 2019