Music Enhancement using GAN (MEGAN)

Final Project Report

Kriti Aggarwal
kriti@ucsd.edu

Digvijay Karamchandani
dkaramch@ucsd.edu

Rushil Nagda
runagda@ucsd.edu

Rishab Gulati
r1gulati@ucsd.edu

ABSTRACT

In this project we have tried to enhance the quality of music using neural networks. We take as input a piece of music that has natural noise such as crowd cheering or applause, or artificial Gaussian noise and try to remove the noise from network. This is based on the Speech Enhancement Generative Adversarial Network where the authors try to improve the quality of speech. We also try our hands on Audio Super-Resolution where we feed as input to the network a downsampled version of a piece of music and output an upsampled high quality music. Our initial intent was to combine the two networks and create an end-to-end denoising and upsampling network. Although we could not achieve this, we managed to individually produce very promising results for both these tasks

1 BACKGROUND AND MOTIVATION

Signal enhancement techniques today, such as Speech Signal Enhancement, Speech Signal Degradation, Speech Filtering Techniques etc., generally operate on spectral domain and exploit the high level features. A majority of these techniques tackle a limited number of noise conditions and rely on the first order statistics. This is a major disadvantage as there are multiple types of noise in an audio clip and in order to remove all the noise and enhance the quality of speech, the techniques we use should be able to understand the structure of noise and be able to separate it from the actual speech and at the same time predict whether a new sound is noise or not and if it is, remove it from the speech. Deep Neural Networks are gaining a lot of popularity these days because of their ability to learn complex functions from a large set of examples. Due to this, they can be used to detect noise in speech and improve the quality of speech as is done in [1]. In [1], the authors use Generative Adversarial Networks or GANs to enhance the speech. They train the model end-to-end by incorporating 28 speakers and 40 different noise conditions and having shared parameters across them. We try to emulate their network that works on speech for musical data. This problem becomes a lot harder when we consider musical data instead of speech because speech is usually monophonic but music is generally polyphonic.

The main motivation to work on this project comes from the idea that usually a majority of the video recordings in concerts or any other outdoor events that play loud music are full of noise that make the recording almost inaudible. Once our network is trained properly, it could be extended to remove this background noise from these videos and allow people to reminisce by looking at the videos without having a disappointing audio. So similar to [1] we try to use GANs to remove artificial noise and white noise from music. We aim to improve the intelligibility and quality of music that is contaminated by natural and artificial noise. We are working with raw audio waveform files (.wav) in the time domain. Our network works end-to-end with raw audio and a major advantage of using Deep Neural Networks is that we don't have to handcraft any features or require any special understanding of the input. Our network takes as input audio clips from different genres, namely blues, jazz ,and classical, which have noise added to them and learns different patches of artificial noise such as Gaussian noise or natural noises like crowd noise and applause noise. The main aim of our network is to learn these types of noises and remove them from the input.

2 NETWORK ARCHITECTURE

The component within our GAN which sieves out the clean signal from the noisy signal is the generator. It's main task is to learn the effective mapping from X' $(original + noise)$ to X, the enhanced version of the original. Ideally, we would want X to be as close to original as possible. Importantly, G does so not by memorizing input-output pairs, but by mapping the data distribution characteristics to the manifold defined in our prior Z.

The generator network G is designed to be fully convolutional, with no dense layers at all, this forces the network to focus on temporally close correlations. The generator G is structured similarly to auto-encoder network in fig 1 . In the encoding stage, the input signal is compressed through various convolutional layers followed by parametric rectified linear units (PReLUs). We decimate the input to get a condensed representation , called the thought vector c, which we concatenate with the latent vector z. This is followed by the decoder network, which decodes $(z + c)$ by means of fractional-strided transposed convolutions (sometimes called deconvolutions), followed again by PReLUs.

The G network also features skip connections, connecting each encoding layer to it's corresponding decoding layer, bypassing the layers in the interim. This is because we could lose multiple low level details, while trying to force all information to flow through the compression bottleneck. Skip connections help in passing this fine grained information directly to the decoder stage from the encoding stage.

The way G learns mapping from noisy to clean is by adversarial training , where we have another component, the discriminator D. D is typically a binary convolutional classifier which learns how to discriminate between the clean signal sifted from noise by G (fake) and the original clean signal (real). G tries to fool D, by adapting it's parameters such that D classifies the generator's output as real. Error back-propagation allows D to get better at finding realistic features in its input and, in turn G corrects its

Figure 1: Encoder-decoder architecture for speech enhancement (G network). The arrows between encoder and decoder blocks denote skip connections.

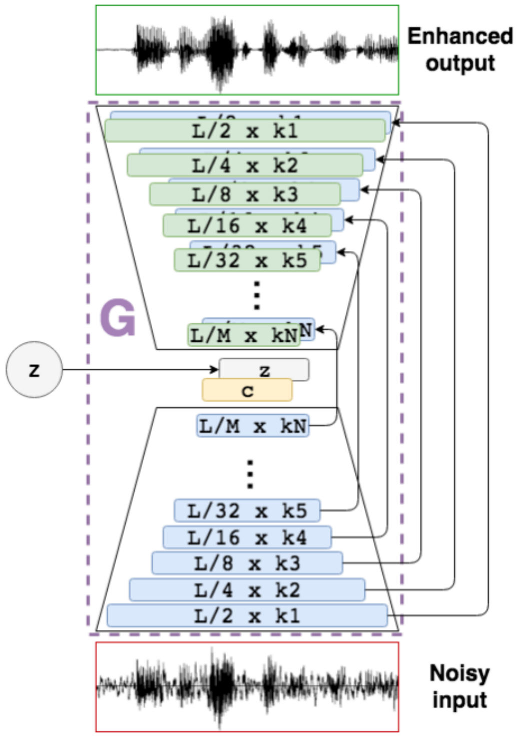
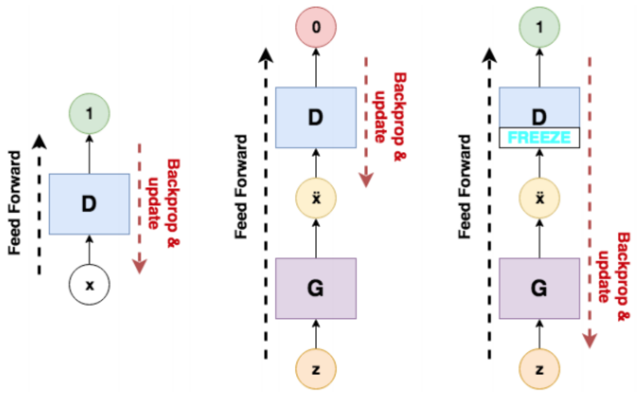


Figure 2: Encoder-decoder architecture for speech enhancement (G network). The arrows between encoder and decoder blocks denote skip connections.



parameters to improve the distribution being learnt, and moving close to the input manifold.

In fig 2, we can observe that the adversarial training is performed as a minimax game, between G and D. The first column in the diagram, we can see the Discriminator learns to classify the true input X as real(1). Following which, the discriminator network D, learns to classify the clean signal from generator G as fake(0), by backpropagates the error to tune itself during training (second column). In the final column, the generator is trained to become better at fooling the discriminator, and learns a better representation of the input in the process.

3 TRAINING

The dataset we used for our project is GTZAN Genre Collection which has about 1000 audio tracks and covers 10 genres and we pick 3 out of those 10: Blues, Classical and Jazz. We train the model on all the Classical and Jazz songs and we take the first 15 seconds of each song. We used raw waveform input sampled at 22,050 Hz so at each second we have a vector of dimension 22,050. We fed this into our network in chunks of length 1024. For adding the noise, we experimented with both artificial noise and natural noise. For artificial noise we used a Gaussian noise and for natural noise we used ”crowd noise” and ”applause noise”. We used 30 seconds of noise and divided it into chunks of size 1024 to match our input size. At each training epoch, a random piece of music was combined with a random piece of noise. We trained our network for about 150 epochs at a learning rate of 0.0003 and used RMSProp optimizer and a batch size of 128 music chunks. To evaluate our network we used the Least Square GAN error loss function given by eq. 1

$$\min_G V_{LSGAN}(G) = \frac{1}{2}\mathbb{E}_{z\sim p_z(z),\tilde{x}\sim p_{data}(\tilde{x})}[(D(G(z,\tilde{x}),\tilde{x})-1)^2] + \lambda||G(z,\tilde{x})-x||_1. \quad (1)$$

4 EVALUATION AND RESULT

We trained MEGAN for both artificial gaussian noise and different natural noises like standard concert/applause/crowd noise. The model was trained for 150 epochs and the model was able to remove test noise with varying accuracy. We used a 80:20 train, test split.

The noises were added at different Signal to Noise ratios. The SNR was decided according to the type of the noise. For the training epochs, we compared the loss with the last 3 epochs and stopped the training if the loss was consistently higher on the validation set.

The table below summarizes the comparison of the results obtained by running MEGAN on different noises at epoch 0 and epochs 150.

4.1 Evaluation

4.1.1 Objective evaluation. To train MEGAN, we used different discriminator and generator losses. For the discriminator we used mean squared loss and for the generator, we used mean absolute loss. For the overall GAN loss, we are minimizing a weighted loss function with 0.5 weight to discriminator loss and 100 weight for the generator loss. Since, the generator loss minimizes the absolute difference between the reconstructed music signal and the original signal. GAN loss provides a reliable window into the performance of our model. Hence, for comparing the performance of our model on different dataset, we used GAN loss as a valid evaluation metric.

4.1.2 Subjective evaluation. A total of 8 listeners were presented with the sample test samples. For each sample, the following three

Table 1: Results

Model	Epochs	Discriminator Loss	GAN Loss
MEGAN on artificial Gaussian Noise	0	0.120204	3.95293
MEGAN on artificial Gaussian Noise	150	0.0548	1.2688
MEGAN on concert noise	0	0.116363	1.11717
MEGAN on concert noise	150	0.003112	0.27615
MEGAN on applause	0	0.1393	1.247
MEGAN on applause	150	0.01493	0.2177
MEGAN on crowd noise	0	0.114692	1.2688
MEGAN on crowd noise	150	0.00229	0.1934

versions were presented, also in random order: noisy signal, original signal, and MEGAN-enhanced signal. For each signal, listener was asked to rate the overall quality, using a scale from 1 to 5. For the description of the 5 categories, they were instructed to pay attention to both the signal distortion and the noise intrusiveness (e.g., 5=excellent: very natural music with no degradation and not noticeable noise). The listeners could listen to any sample as many number of times and the listeners were asked to pay more attention to the comparative rate of the three signals.

Listeners gave on average rating of 4.5 for the artificial noise, 3.5 rating for the samples with concert noise, 2.5 rating for the samples with applause and 2.75 rating for the crowd noise. The results show that the model was able to remove the artificial noise with high efficiency and the performance dipped for the natural noises. Out of all the natural noises, applause was the hardest to remove. This can be attributed to the fact that applause is a sharp sound and results in a higher distortion of the music signal as opposed to the crowd and general concert noise.

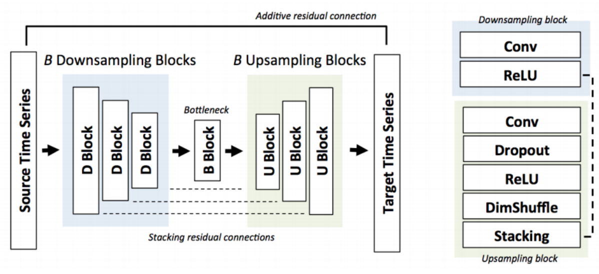
5 SUPER RESOLUTION

We also tried our hands on Audio Super-Resolution, building on [2] which is basically a bandwidth extension problem. Our intention originally was to combine these two functions into one network. The main idea here is to "fill the gaps" by interpolating music between two given points. We down-sample a given piece of music to a low frequency such that it contains only 25% of the original data points and then reconstruct the high quality music. This method is inspired from super-resolution algorithms for image where a low resolution image is converted to a high resolution image.

Our network consists of 4 Downsample Convolution blocks with ReLU activation function followed by 4 Upsample blocks. The Upsample blocks are similar to Downsample blocks other than the fact that they have a dropout layer and have skip connections from corresponding Downsample blocks. The architecture of the network can be seen in the Figure 3.

We took pieces of music sampled at 16000 Hz, downsampled them to 4000 Hz, which basically means we kept only every 4th data point in the vector. We then used cubic spline interpolation to reconstruct this downsampled waveform back to 16000 Hz and fed it into the network in chunks of 2048. This is basically to keep the input and output sizes of the network the same. Our results were significantly better than baseline interpolation techniques and our network was able to reconstruct higher frequency notes.

Figure 3: Network architecture for super-resolution



6 CONCLUSION

We successfully adapted modified Speech enhanced GAN for the music dataset. We constructed a new dataset by combining GTZAN Genre Collection with natural concert and artificial gaussian noise. The model was successfully able to remove both the artificial and natural noise with varying accuracies. We were also able to successfully adapt the speech super resolution paper for reconstructing low resolution music signals.

Due to time and resources constraints we were not able to train a joint model which could remove noises from different natural and artificial sources together. For future work, we further modify the model architecture and fine tune hyper parameters to better suit our case. We can also modify the dataset to have chunks of signals with no noise to make the model be more robust.

REFERENCES

[1] Santiago Pascual, Antonio Bonafonte, Joan Serr, SEGAN: Speech Enhancement Generative Adversarial Network, 2017
[2] V. Kuleshov, Z. Enam, and S. Ermon. Audio Super Resolution Using Neural Networks. ICLR 2017 (Workshop track) V. Kuleshov, Z. Enam, P. W. Koh, and S. Ermon. Deep Convolutional Time Series Translation, ArXiv 2017
[3] GTZAN Dataset: http://marsyasweb.appspot.com/download/data_sets/