Generation of vocals for accompanying music

COREY BOND, REECE WALSH, and PAULA WONG-CHUNG

1 INTRODUCTION

The challenge of generating audio using neural networks has been a topic of discussion for more than twenty years now [8]. The difficulty arises from the sequential nature of audio, along with the multiple outputs and channels created by various musical elements such as instruments, melodies, and vocals, which adds another layer of complexity.

Neural networks have frequent applications in image generation, but less so with images representing frequency such as spectrograms. As such, applying deep neural networks to databases of sound representations provided an interesting challenge for us to confront in this project. We investigated the possibility of generating vocal accompaniment to instrumental tracks with the purpose of eventually generating novel lyrics for instrumental pieces. Popular songs were split into separate instrumental and vocal tracks using Spleeter[1], a library provided by deezer[2]. The files were then converted into Mel spectrograms that could be used as an image input to a neural network. Initial trials with a Variational Autoencoder (VAE) network[9] indicated that the spectrograms representing the music were too complex and contained too much detail to be sufficiently generated with a simple VAE. As spectrograms are visual representations of the audio files, any loss in the image results in noticeable noise in the generated audio file. As a possible solution to this problem, two additional structural improvements were investigated: an AutoVC[5] and a VAE/GAN architecture[1].

Based on the above, our goals were to:

- Generate a private database composed of pairs of 10-second audio files to facilitate supervised learning of the network. Each pair is comprised of an instrumental input and a vocal output file.
- Generate an associated database of Mel spectrograms that represent the audio files.
- Develop a model that could produce interpretable vocals for an associated accompaniment audio clip.

Section 3 will detail the development of an appropriate private database, and the extension of that database to Mel spectrogram form. Sections 4, 5, and 6 detail the development process through different model iterations. We finish up by detailing the future work in Section 7.

2 RELATED WORK

Attempts to use neural networks with various audio inputs over the years have resulted in a variety of different methods and structures. VAE networks were consistently represented in the literature, and became a focus of our efforts to identify relevant research in the field. We were motivated by the approaches in a few key papers.

Our initial research was driven by the work detailed in “Generation of lyrics lines conditioned on music audio clips” [9]. The authors proposed a bimodal VAE structure that generated lyrics based on spectrograms generated from audio clips. Since their goal was to produce lyrics which would evoke an emotion suitable for a given audio clip, clips were split into categories based on general emotion. Their system was intended as a creative aid for musicians to generate lyrics line

[1] https://github.com/deezer/spleeter
[2] https://www.deezer.com/us/

by line. Since the VAE encoder and decoder structure used was fairly simple, it provided a good baseline for our model.

Similarly, a method was proposed in [1] that implemented an autoencoder, but leveraged a generative adversarial network (GAN) to minimize noise in the generated images. The idea was to minimize the loss, and to create images with feature-wise errors rather than element-wise errors. The network learns to generate realistic images that have greater detail than similar models, and is illustrated with generated images of faces.

Another possible method that was able to preserve detail in audio generation with minimal loss was proposed in [5]. Due to the challenging nature of training GANs, a vanilla VAE that trains as easily as a conditional variational autoencoder (CVAE) was suggested. This was possible due to a carefully curated bottleneck in the network that retrieves style-independent code used for vocal generation and conversion. The AutoVC is a style-transfer model built specifically for speech/voice. Its structure includes two encoders that are fed into a decoder: one for the speaker and one for content. Together, these form the VAE base of the system. The VAE output can then be fed into a spectrogram inverter that converts the spectrogram of the audio into a speech signal which can then be transformed into an audio file. The authors used the WaveNet vocoder [4] pretrained on the VCTK corpus[3].

The WaveNet vocoder was introduced in 2016, and represents a "... deep neural network for generating raw audio waveforms" [4]. It is a generative model that works directly with the raw audio waveform. The system architecture is based on dilated causal convolutions - convolutions that Oord *et al.* proposed as a way to maintain the temporal structure of the audio data. This is similar to a masked convolution in image files [2]. With WaveNet, the generation can be conditioned to result in audio with specific desired characteristics [4]. The authors completed three different experiments, one of which had positive results on music, which furthered our interest in this model.

3 DATABASE DEVELOPMENT

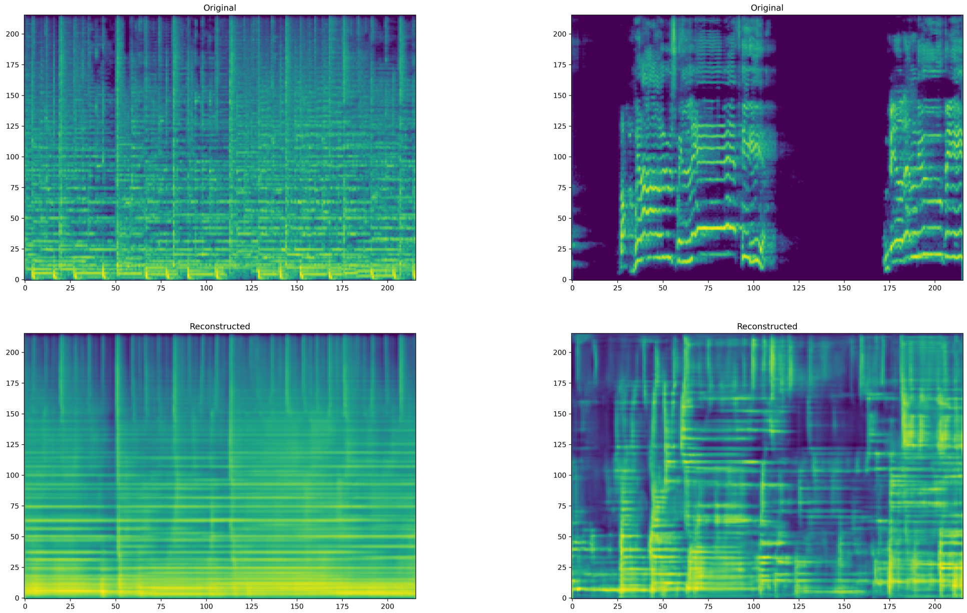
This research was driven by a desire to extend the related work that we identified into a purely musical/audio format. To the best of our knowledge, while there is a strong representation of research related to style transfer for audio [5], as well as lyric generation based on audio [9], representative work related to generating accompanying vocals for instrumental tracks is almost nonexistent. This represented a challenge that we were not fully aware of when undertaking this project.

Due to the novel nature of this research, no appropriate existing database could be identified. The difficulties associated with music generation via neural networks are primarily the sequential nature of the music, the complexities associated with melody and multiple instrumental representations, and the additional channels represented by voices [8].

To address this, the primary focus of our early development was to identify music catalogues of interest that not only had enough audio clips to properly train a model, but were also fairly homogeneous in style. The VAE in [9] utilized music catalogues consisting of 239 songs. This seemed like a reasonable target size to aim for. Taylor Swift and Elton John were identified as possible artists due to their extensive music catalogues and the appeal of vocals from a solo performer. This would set the network up for success by requiring less distinction between different overlapping voices. Since each artist has over 200 songs in their catalogue, a wide range of training data was easily gathered.

Once all of the required music clips were acquired, the focus was on using the Spleeter tool to separate the vocal and accompanying audio from the files. Since the resulting files were large in

[3] https://datashare.ed.ac.uk/handle/10283/3443



(a) Generated Elton John accompaniment spectrogram produced by VAE trained on Elton John music

(b) Generated Elton John vocal spectrogram produced by VAE trained on Elton John music

Fig. 1. Example of Elton John spectrograms generated from VAE using ResNet-18 trained on Elton John data

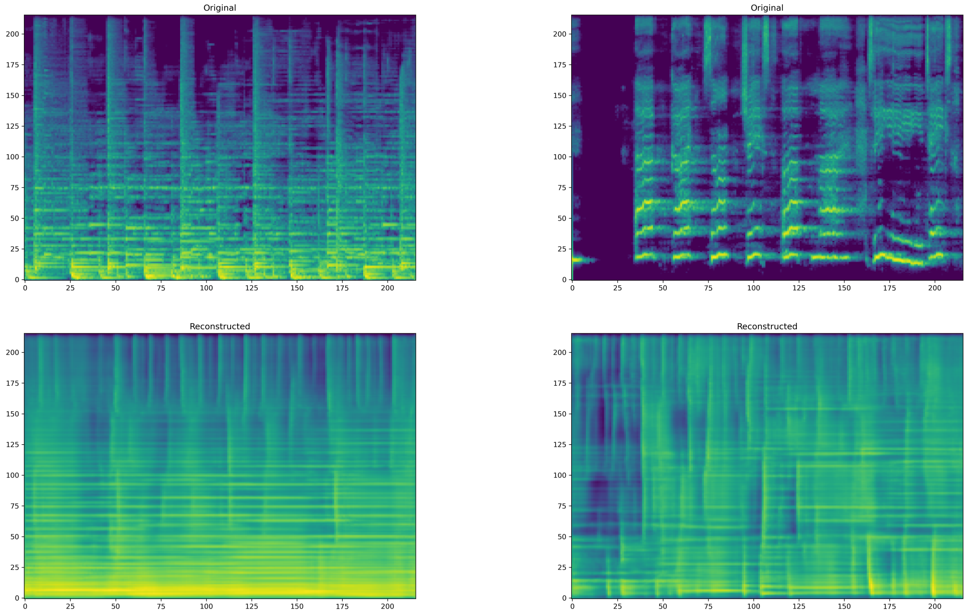
size, a data handling function was built to split each song into ten second clips, and to generate an associated spectrogram. This was made possible by the librosa package for python [4]. The generated spectrogram was a power spectrogram commonly called a Mel spectrogram, where Mel is simply a scale for the spectrogram [6]. In order to be properly processed by our VAE, the Mel spectrogram was converted to a typical dB (or decibel) spectrogram. This spectrogram was then normalized by approximating the decibel range represented by the spectrogram before being converted to a tensor and stored in our database. Prior to the normalization process, a filter was applied to exclude clips that had a maximum volume less than 27 decibels as they were less likely to include the singer vocalizing.

The process resulted in two datasets that could be leveraged for our model training. The dataset containing Elton John music resulted in 4,505 items which each had an accompaniment and vocal component. The Taylor Swift dataset was similar, and resulted in 3,778 items.

4 VAE APPROACH

The first VAE model attempted was fairly simple. The encoder and decoder both made use of ResNet-18 and ResNet-50 models respectively in two VAE architectures. The convolutional layer dimensions defined in this architecture were specific to our data. The model was first trained for

[4] https://librosa.org/doc/



(a) Generated Taylor Swift accompaniment spectrogram produced by VAE trained on Elton John music

(b) Generated Taylor Swift vocal spectrogram produced by VAE trained on Elton John music

Fig. 2. Example of Taylor Swift spectrograms generated from VAE using ResNet-18 trained on Elton John data

550 epochs using ResNet-50 on the Taylor Swift dataset, and resulted in a spectrogram with poor audio fidelity. It was determined that the model started to overfit after 300 epochs, and that the loss was still causing a problem when the network overfit. We then trained the same VAE on the Elton John dataset for 300 epochs using ResNet-18, but with different scaling in the dataset spectrograms. This resulted in less loss in the basic conversion from audio to spectrogram and then back to audio, but the resulting spectrograms still had had lost too much audio fidelity to be usable (see Figure 1). Notice that in Figures 1 and 2, the poor reconstruction capabilities are causing horizontal blurring or banding across the spectrogram - masking significant audio features such as high frequency, percussive instruments. This is much more pronounced in Figure 2 due to passing in data that the network had not been trained on. This reconstruction issue translated to noticeable noise and fewer recognizable instrumental features in the audio when a wavefile was generated. We were thus motivated to explore further options.

5 LEVERAGING THE GAN

A model incorporating a GAN was approached next, but presented challenges due to the complexity of the structure. We originally attempted to replicate the structure of [1], and referenced the implementation provided in the paper as a starting point. The motivation was to feed the encoder and decoder into the GAN discriminator, and to create a hybrid model. The suggested structure

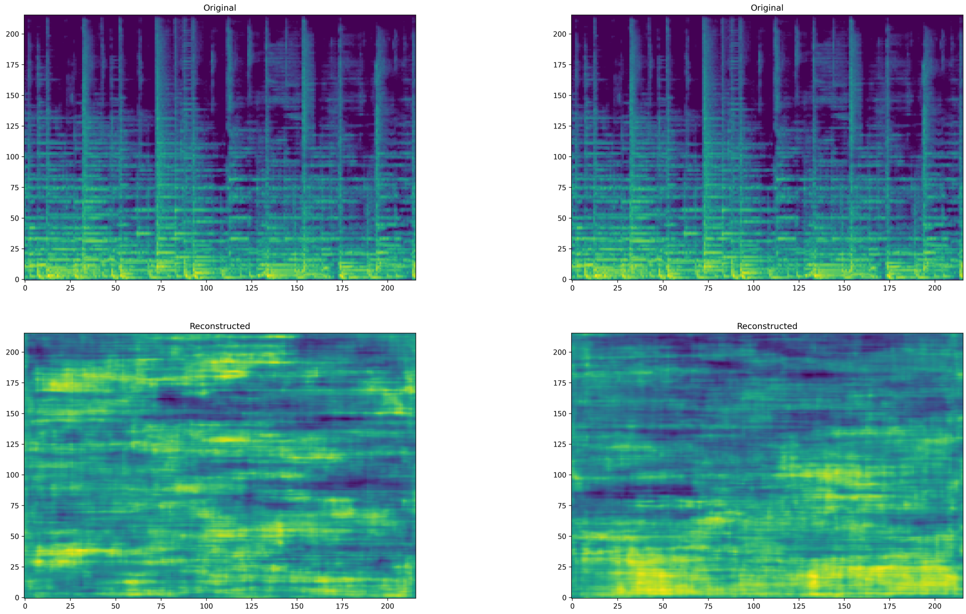


Fig. 3. Interim training results from the VAE/GAN-style approach. Notice that there are still significant issues with loss, although there are more details captured than the original VAE approach.

used repetitive convolutional, batchnorm, and ReLU layers. However, due to increased performance expectations, we reversed the order of the ReLU and batchnorm layers [3]. The comparatively high resolution of the spectrogram images led to some dimensionality issues. In early development, the dimensionality was so high that we continuously ran out of memory on the GPU and could not process the spectrograms. Various steps of downsampling and upsampling were required, and a balance had to be found between appropriate downsampling to manage memory and size and maintaining enough features for the network to have usable data. It was found that upsampling in the decoder caused significant loss in the image when the dimensions of the encoder output were too small. This indicated that an image-based approach could be inappropriate for the audio spectrogram. For example, when we look at the images in Figure 4, we can see that while the VAE/GAN approach is appropriate to get rid of the blurriness in the images, it also ends up adding artefacts to the images. Initially, the hope was that an appropriate loss function would counteract the development of those artefacts. We attempted to implement the loss function detailed in [1], but the significant complexity of the various calculations involved, along with the appropriate time to apply them during training, resulted in varying degrees of failure when attempting to train the model. We referenced the code for the loss function from [7] by cross-referencing the calculations against the various equations provided in [1]. With this approach, we were able to trace how the loss could be applied at various steps. However, the difficulties in training a GAN, along with the inappropriate nature of applying the image-based VAE/GAN to an audio setting, and the exploding loss resulting from the loss function indicated that we needed to pivot to another approach.

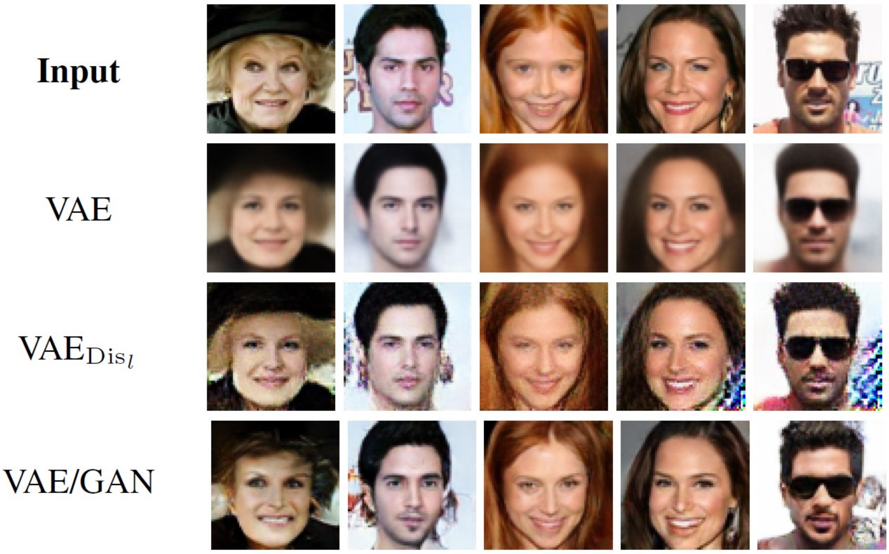


Fig. 4. From [1]: "Reconstructions from different autoencoders." An example of how the VAE/GAN could increase sharpness in an image, but potentially introduce artefacts.

6 AUTOENCODER UTILIZING WAVENET

At this point, our attention shifted to an autoencoder approach based on the autoVC proposed in [5]. The conditional VAE proposed an easier to train method, but with the drawback that distribution matching could not be guaranteed. In order to provide measurable results, the structure from [5] was largely followed. Our primary concern was determining the best approach to train the model to suit our vocal generation endgoal. The original autoVC approach was meant for style transfer from one speaker to another and so required training both a content encoder and a style encoder. This was determined to be unnecessary for our initial approach as the focus was on vocal generation matching existing style, rather than application of an alternative style. Due to this, the training pipeline was adjusted to reflect vocal and accompaniment encoders replacing the content and style encoders detailed in [4]. The updated model structure can be found in Figure 5. The resulting spectrogram would still be affected by the loss, but by leveraging the WaveNet vocoder at the end of the pipeline, the conversion back into audio could be improved.

We had initially hoped to use the data from the original trained VAE model for the vocal and accompaniment encoders. However, the selected spectrogram generation settings did not match what was expected for AutoVC's approach. The hop length, window size, and Mel scale were originally selected such that square spectrograms were produced to match the VAE's input expectations. AutoVC, on the other hand, placed greater emphasis on time-based representation (smaller hop length) and less emphasis on signal representation (a smaller Mel scale).

Appropriately and quickly training the vocoder required simultaneous use of both the accompaniment and vocals spectrograms, rather than training two separate vocoders. Training individual vocoder networks on the vocals and accompaniment, respectively, for later combination would be more accurate. However, this was determined to be less relevant than proof-of-concept results that could be obtained by the combined training method.

The interim vocoder results shown in Figure 6 were promising. Although the output of the vocoder is a waveplot rather than a spectrogram, we can see noticeable feature matching, and

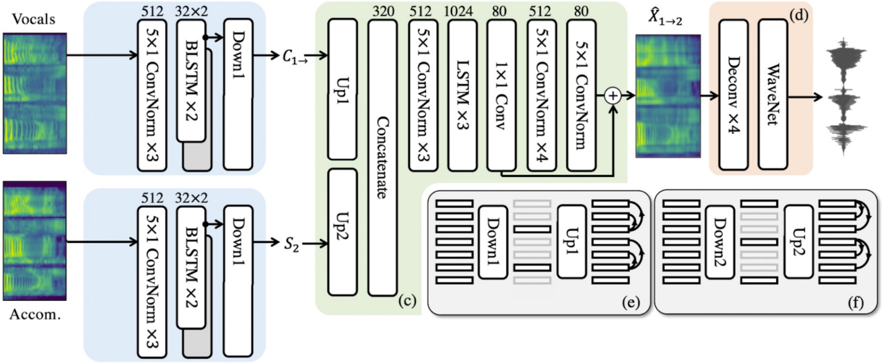


Fig. 5. The updated model architecture for our autoencoder approach. The original diagram can be found in [5]

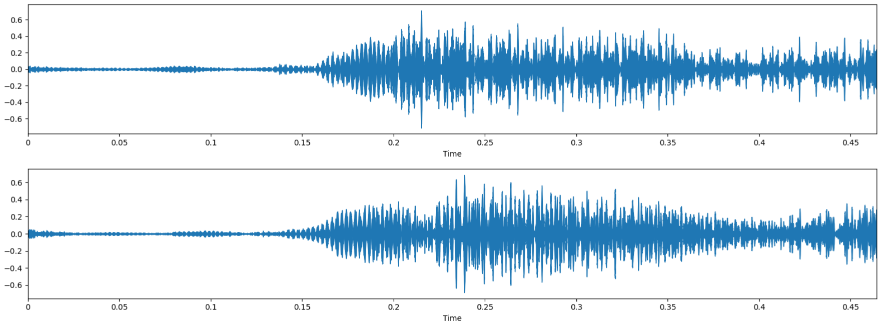


Fig. 6. A waveplot produced during training of the vocoder used to support the autoVC-style approach

the generated audio was of a higher quality than previous results. Once the vocoder was trained, attention shifted back to the autoencoder. A key focus during training of the autoencoder structure was determining the appropriate bottleneck size. When the bottleneck was too small, random vectors passed in for vocal synthesis produced noise that was unrecognizable as human words or vocalizations, although inference on existing data produced more recognizable results. Widening the bottleneck too much resulted in audio that sounded like a mashup of a large number of small independent audio clips. These issues were due to the latent space produced by the model being uninterpretable. Shrinking the bottleneck size to a 1-dimensional vector of length 256 and enforcing VAE-based training loss resulted in better results. At this point, it became apparent that the clipping occurring due to the shorter audio samples being used was causing additional audio disturbance in the results.

After training the VAE-based model, testing was performed to confirm equal contribution between the vocal and accompaniment input spectrograms for interpretable vocal output generation. Accompaniment contribution was of particular concern due to the ease by which the network could reconstruct vocal output through heavy reliance on vocal input. Ideally, the narrower network bottleneck selected would enforce reliance on accompaniment data in order to produce an accurate reconstruction. Experimentation to test this hypothesis was initially performed by replacing the accompaniment vector with random data while maintaining legitimate vocal input. We found that

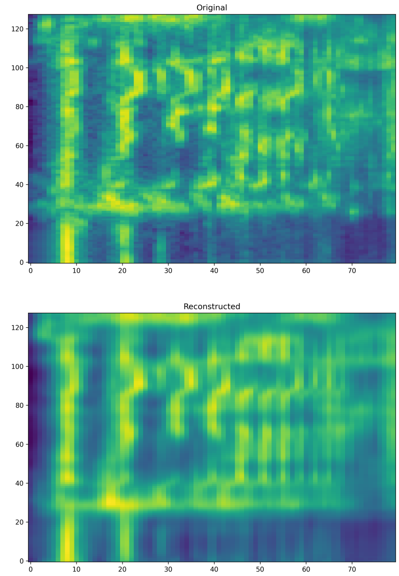


Fig. 7. The spectrogram resulting from the trained autoVC-style approach

the subsequently generated waveform output was uninterpretable for all generated samples. Additionally, mismatched accompaniment-vocal data produced interpretable but distorted waveform output. These findings confirmed that use of the smaller bottleneck encouraged some form of reliance on accompaniment data to produce accurate output.

With the previously proposed network and training additions, we were able to produce significantly improved results. This included synthesized vocals produced by appending random tensors to accompaniment data.

7 FUTURE WORK

The results from this project are promising, and leave room for further improvement and development. Primarily, steps should be taken to continue improving the autoencoder model based on the vocoder/autoVC structure, so that vocals producing English words can be generated from random input. Further to this, we could consider adding in text vocals to help produce a more comprehensible audio result. By increasing our dataset size and continuing to make architectural improvements, we can work to further improve the model to the point where vocals will be produced without the audio artefacts that we are hearing in the current results, and clipping issues will be minimized. Recombining the vocal audio is currently completed outside of the model, so further work could be completed to build the combination into the output, or database function of the model.

Once the model starts to produce more interpretable results, exploring the latent space could provide another avenue for research to support the improvement of the model. To start with, it would be an interesting exercise to fully map the latent space, and perhaps grouping by years to see how the artist's style may have changed over the years. We noticed in preliminary mapping that

Taylor Swift's latent space was quite consistent, but difficult to interpret at the level of individual songs.

It is also worth exploring the possibility of applying style transfer using the trained model. As the original focus of the autoVC structure was style transfer on voices, it will be interesting to explore style transfer on music. This would allow us to attempt to change the genre or style of an artist, perhaps changing a pop artist into a rock artist, or creating a 'mash-up' style for a particular song or artist.

Our work is nowhere near done, and the applications of neural networks to music are plentiful. As a result, the potential future work with these results could take many different approaches, and be widely applicable. Once an accurate model is trained to produce vocals based on accompaniment, we will be able to use that model to explore future uses of the latent space.

8 CONCLUSION

Throughout this research, we have identified significant support for the generally accepted difficulty of producing intelligible audio files. The work is still in the early stages, and has many possible avenues for future work. The commonly used VAE models result in too much loss, and attempts to improve the results leveraging a GAN were inappropriate for audio use. Further work using an autoencoder result had more promising results that can provide the baseline for continued research.

A key factor in the difficulties faced when developing the models was that reconstruction loss was the only appropriate measure of accuracy of the output. This did not provide a realistic idea of how the audio would sound, and resulted in the employment of more subjective measures of model improvement. By listening to the audio output after each model change, we were able to determine whether the model had improved in a way that was noticeable to each of us.

The iterative process of development allowed us to produce a model that was reasonably good at reproducing input from both the Taylor Swift and Elton John datasets. However, we found that this does not necessarily translate to good results from latent space exploration. We were able to consistently improve by adjusting the model, but were held back by the lack of an accuracy metric that could quantify whether a sound clip is audibly interpretable. This is a really interesting field, and the possibilities for continuing the work are well worth exploring.

REFERENCES

[1] Larsen, A. B. L., Sønderby, S. K., Larochelle, H., and Winther, O. Autoencoding beyond pixels using a learned similarity metric. In *International conference on machine learning* (2016), PMLR, pp. 1558–1566.

[2] Misc. Causal convolution. Available at https://paperswithcode.com/method/causal-convolution (2022/04/03), 2022.

[3] Mishkin, D. caffenet-benchmark/batchnorm. Available at https://github.com/ducha-aiki/caffenet-benchmark/blob/master/batchnorm.md (2022/04/11), 2016.

[4] Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. Wavenet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499* (2016).

[5] Qian, K., Zhang, Y., Chang, S., Yang, X., and Hasegawa-Johnson, M. Autovc: Zero-shot voice style transfer with only autoencoder loss. In *International Conference on Machine Learning* (2019), PMLR, pp. 5210–5219.

[6] Roberts, L. Understanding the Mel Spectrogram. Available at https://medium.com/analytics-vidhya/understanding-the-mel-spectrogram-fca2afa2ce53 (2022/04/03), 2020.

[7] Scuccimarra, E. A. vaegan-pytorch. GitHub repository available at https://github.com/escuccim/vaegan-pytorch (2022/04/04), 2020.

[8] Tham, I. Generating Music Using Deep Learning. Available at https://towardsdatascience.com/generating-music-using-deep-learning-cb5843a9d55e (2022/04/03), 2021.

[9] Vechtomova, O., Sahu, G., and Kumar, D. Generation of lyrics lines conditioned on music audio clips. *arXiv preprint arXiv:2009.14375* (2020).