Paired Timbre Transfer

Madhur Sudarshan
17D070009
Dept. of Electrical Engineering
Indian Institute of Technology Bombay
Mumbai, India
madhurs@iitb.ac.in

Harsh Prashant Dolhare
17D100009
Dept. of Electrical Engineering
Indian Institute of Technology Bombay
Mumbai, India
harshdolhare99@iitb.ac.in

***Abstract*—In this project we take on the task of musical timbre transfer, where the goal is to manipulate the timbre of a sound sample from one instrument to match another instrument while preserving other musical content, such as pitch, rhythm, and loudness. Existing approaches involve image based style transfer on unpaired audio samples, or Spectrogram representations. We propose using a feed-forward variant of the WaveNet deep neural network [1] to work entirely in the audio domain, which avoids addition of unwanted elements in the final audio waveform that are associated with reconstruction of audio from an image. As we borrow a model typically used for real time audio processing, rather than focusing on unpaired audio samples we work on pairs of digitally generated instrument wave forms to investigate effectiveness of a shallow net. Our results show some promise but are limited by model depth and artifacts that keep it from being usable in the industry.**

***Index Terms*—Timbre Transfer, Wavenet, Real-time signal processing, Generative Synthesis**

I. Introduction

Timbre is the perceptual quality of sound that is used to differentiate between different musical instruments that might otherwise be playing at the same fundamental frequency, note duration and loudness. Modelling timbre is a multidimensional challenge with inherently subjective ideals, but there has been considerable research into synthesis of such sounds. Even with state of the art techniques for modelling timbre, when it comes to acoustic or analog instruments there has been resistance over decades to replace live instruments. This is due to limitations in accurately simulating all parameters involved with an acoustic instrument. Consequences of this limitation are that sampling audio remains the only practical alternative for most studios in place of depending on Live musicians.

Live musicians and samples come with a large overhead in resources, including equipment to record at scale, budget and time taken to painstakingly compile sample libraries. Even with the best quality samples it takes expert manipulation to extract subtle dynamics in the audio corresponding to musical features like - muting, pitch bends, vibrato etc. This motivates our interest in the field of neural network based timbre transfer.

Timbre transfer is the concept that one can take the audio from one instrument and output the audio of another instrument while preserving musical content like pitch, rhythm and loudness. The concept of neural network timbre transfer hence holds significance in the musical industry as it allows a single musician to track audio on an instrument that they possess proficiency in, enabling them to now single handedly produce music that otherwise required experts and large production budgets. Additionally it allows electronic music producers to experiment with sounds that may not otherwise exist in the real world.

Research in this field has so far been limited to converting Audio into spectrograms or other image based representations, and borrowing from extensive research into style transfer for images as is seen in Timbretron [2] that uses CycleGAN [3] for unpaired instrument to instrument timbre transfer. More recent research into "Attention-based timbre transfer" [4] that utilises a Mel-GAN [5] which maps audio to mel-frequency spectrograms in real time along with WaveNet that can be used to extract high quality real time mappings between two instruments. We propose analysing the capabilities of WaveNet [1] on timbre transfer without invoking an image representation. Our model is inspired by a feed-forward variant of WaveNet as proposed by [6], which attempted to create a real time transformation between a clean guitar signal and a distorted signal, treating it as a black box modelling problem.

We use the same black box modelling approach as taken by [6] to map two instruments together given paired training data. Our work differs from other research as it stays entirely in the audio domain. Conversion to Image based representations fails to utilise underlying properties directly from the time series data, and conversion between images and audio in high quality becomes a bottleneck. Our model does not suffer from these issues. We chose to use a lightweight model to meet latency requirements that are crucial in the music industry.

Metrics to evaluate such a task are harder to define, a large part of our analysis shall be subjective using vocabulary in terms of usefulness in the music Industry. In order to train the model we use error to signal ratio as will be elaborated upon later. As this process is limited in computational capabilities due to real time behaviour, one of our goals is to identify by-product mappings that may not be perfect representations of the target instrument, but lie in between the source and target timbre. If such by products (due to incomplete training or lack of depth of model) are pleasing to the ear they hold promise for creation of new instruments and timbres that do not exist in the real world.

II. Related Work

The field of timbre transfer has seen a lot of prominence and active research 2018 on wards. The most cited of which is "TimbreTron: A WaveNet(CycleGAN(CQT(Audio))) Pipeline for Musical Timbre Transfer" [2]. As the hefty title suggests this paper works with a Constant Q Transform (CQT) representation of an input audio sample, which is their choice of image representation for Audio. It operates on unpaired CQT representations by style transfer as implemented in CycleGAN. In order to observe high quality audio outputs it uses WaveNet at the final stage to reconstruct audio. The limitation of this model is that it requires the entire audio sample to be converted to an image for it to apply style transfer. This means that it cannot be implemented in real time, but shows promising results otherwise.

A more recent paper tackling timbre transfer is "Attention-based Timbre Transfer" [4] which uses Mel-GAN [5] for spectrogram inversion, which they claim provides a fast and parallel alternative to other auto-regressive (i.e. time series prediction approaches that use history) music generation approaches. At its essence they add attention to the image to image translation required for style transfer by modelling it as an unsupervised image translation problem as motivated by [7]. MelGAN comprises of generative adversarial networks to produce high quality audio wave-forms given a Mel-frequency spectrogram, this transformation allows for performance 100x faster than real time on a 1080 GPU, enabling real time timbre transfer. Though this paper is real time it still works in the image domain, and is trained only on monophonic sounds. The Mel scale is a perceptual scale of pitch distances which they claim is better for timbre analysis.

There does exist more research into timbre based synthesis, that offer high quality audio like Google's Nsynth [8], WaveNet AutoEncoders [9] and research at our own institute by Krishna S. et al [10] who built a Variational Parametric Model for Audio Synthesis among many other papers. Some methods assume that the user provides musical inputs encoded in MIDI information to generate tones. Certainly one approach to timbre transfer is to extract MIDI information from audio generated on one instrument and which can be used to play a virtual instrument with timbres generated (using any orthogonal method). We were more interested in approaches that perform no explicit abstraction or recovery underlying musical note information and could solve the problem of transferring timbre in a single neural net. This brings more restrictions on the generality of the model as we neeeded paired samples but seemed worth exploring.

Vocal Style transfer is another application akin to timbre transfer, that has benefited majorly from Spectrogram + CycleGAN inspired approaches. A recent paper [11] utilises TravelGAN which introduces a siamese network in addition to the Generator and Discriminator, thus avoiding the emphasis on pixel wise losses seen with CycleGAN. Other popular approaches require parallel speech data (akin to our paired instrument data) with RNNs [12], or with CNNs [13]

Our intention is to realise a real time timbre transform that avoids conversion to the image domain. Within the limitations of a course project we decided not to attempt an unpaired timbre transfer problem, and instead investigate the capabilities of a WaveNet like model for specific instruments to analyse the potential applications, as there are an endless set of instruments and dynamic details for which transfer could be investigated.

Our inspiration for this task comes from the paper "Deep Learning for Tube Amplifier Emulation" [6] which has had high impact on the industry of Signal Processing for Guitars, enabling musicians to produce high quality guitar tones from home without access to expensive tube based amplifiers. Their feed-forward variant of WaveNet was able to learn the non-linear distortions produced by a guitar amplifier that was largely indistinguishable to the tones generated by real amplifiers to a MUSHRA test. This Neural network based approach quickly unseated modelling based approaches to simulate an Amplifier, and has since become the industry standard even for musicians with access to expensive recording gear. Their work also sees large application in live as the network can run in real time, saving musicians the burden of carrying heavy amplifiers and speaker cabinets.

Our hope was to utilise their model on the domain of timbre transfer, which was largely unexplored. Although many other papers do utilise WaveNet increasingly for creation of high quality audio, most of them transfer to the image domain, so we hoped to explore whether a direct WaveNet model could do the task, and analyse its strengths and weaknesses.

III. Datasets

Our requirements for this task were audio files generated from performances on different instruments that played the exact same notes with matching intensities at the same time.

Rather than depending on finding a high quality recorded audio dataset for this investigative project, we generated all the audio files used in this project using VST (Virtual Studio Technology) plugins in the industry standard DAW (Digital Audio Workstation) Ableton Live. This allows us to generate samples of audio that correlate with each other exactly in terms of pitch and loudness, only differing in timbre as long as they are generated using the same underlying MIDI (Musical Instrument Digital Interface) information, which is an industry standard to notate musical information in terms of pitch, loudness note duration etc.

In order to generate audio files we used the MIDI-BACH dataset [14] which is a catalog of 18^{th} Century Classical composer J.S Bach's compositions using different instruments. The recommended size of training data set in [6] was 5 minutes of audio so we began with that constraint on the size of training data.

Given a multitude of choices for which instruments to pair and train for this project we decided to use acoustic instruments that exhibited similar ADSR curves (Attack-Delay-Sustain-Release) without echo and reverb. For us this classification broadly differentiates between wind/brass instruments

that may sustain indefinitely to percussion driven instruments that provide impulses to a string or resonant metal tube. This choice was made as the model [15] we were using had been reported to perform poorly at modelling reverb and delay effects due to the small size of the neural network used. To normalise further we set the attack on these instruments to be the same, and equalised gain in post, finally converting all instrument files to mono, 32bit Floating point at 44.1 kHz. The songs used were - Bach Cello Suite (Prelude + Allemande) for monophonic audio and Fugue 3,4 and Prelude 4 for polyphonic audio. The generated audio files were further split into 60% training, 20% validation and 20% test.

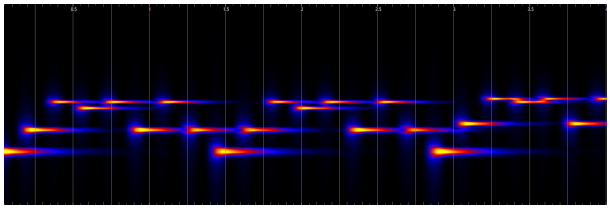
Each piece belongs to only one key signature (which implies that each piece majorly only uses a subset of 7 notes from the total available 12 notes). Additionally each piece is played at one fixed tempo. Although there are sections where the song may speed up and slow down it only does so in integer multiples of the base tempo. To mitigate these properties of classical music from the era of Bach we effectively create a medley or a remix of different tracks by appending and mixing different MIDI files, so as to provide variations in tempo and scale in the training audio files.

Since we are entirely working with paired digital instruments there is no question of a noisy dataset or multiple sources. But there are a lot more subtleties and nuance to a Data-set that can be ascribed to features like dynamics of the parts, inherent non-linearities in the tone generation, stochasticity in the note onset as compared to harmonic stability after, and responses to monophonic vs. polyphonic audio signals. These dynamics also vary between musical genres and instruments. Within the scope of this project and the fact that we were restricted to a shallow neural network we decided to analyse results on Western classical music as it is well studied by musicologist and has a rich vocabulary of terms we could use to describe the models performance.

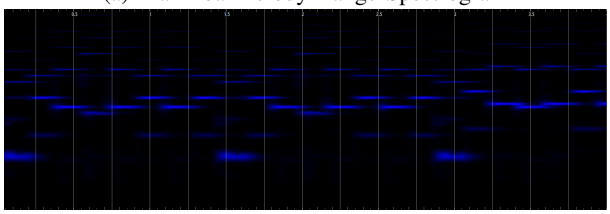
The final aspect of dataset generation are the instruments and direction of mapping. We chose to generate audio corresponding to a Grand Piano, a Jazz Guitar and a Marimba on both the monophonic and polyphonic MIDI data. All three are popular acoustic instruments, with differing timbres. We chose the mappings Piano to Guitar and Piano to Marimba as explained below-

A. *Properties of Selected Instruments*

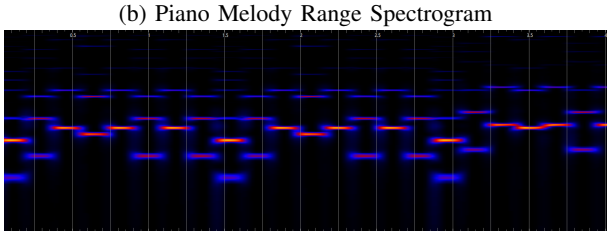
We analyse the timbre of the given instruments based on a melody range spectrogram representation. Marimba are percussive instruments with pipes for resonance. This leads to a very linear behaviour and a distinct lack of overtones making for a pure sinusoid like sound. The guitar as a contrast is rich in overtones as most of the tone is determined by string resonance, where a string consists of a core with an external wrapping. This leads to many different harmonics and overtones. The grand piano falls in-between these two instruments. It is similar to a guitar in that its tone is derived from vibrating strings, but it has less emphasis on higher harmonics, which is why the timbre of an average guitar is brighter than that of a standard grand piano. Additionally an FFT analysis reveals that our Piano tone had no frequencies above 4kHz, whereas the guitar's range extended till 8kHz. These inferences are evident in the spectrograms as seen in Fig. 1 and 3.



(a) Marimba Melody Range Spectrogram



(b) Piano Melody Range Spectrogram



(c) Guitar Melody Range Spectrogram

Fig. 1. The first four seconds of Bach's cello suite
X-axis represents time, Y-axis represents identified melody note, intensity captured in colour. Window Size of 8192 samples

IV. Analysis Pipeline

Our goal was to stick to a low complexity model as far as possible, so the steps in our training are -

1) Training a Low Complexity Model
    a) Tune hyperparameters without changing model
    b) Evaluate performance on monophonic and polyphonic samples
    c) Interpret Results for each instrument
2) Increase complexity of model based on insights from 1.
    a) Tune hyperparameters without changing model
    b) Evaluate performance on monophonic and polyphonic samples
    c) Interpret Results for each instrument

Our code source [15] had implemented this variant of WaveNet in such a manner that it could be used in any DAW with low compute demands by loading the model into a VST plugin. This approach works great for simulating non-linear distortion effects attributed to both solid state and tube based amplifiers, and the Impulse Responses corresponding to Guitar Cabinet Speaker. Thus our expectations were that even with a low complexity model we should be able to capture correct frequencies when mapping two instruments. Their

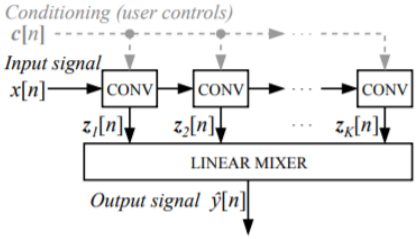


Fig. 2. WaveNet-style architecture(Image taken from [])

documentation also described poor performance on delay and reverb sound effects which we hoped would not effect sound quality of our acoustic instruments emulation.

A. Architecture Description

WaveNet is a 1-D fully convolutional neural network in which convolutional layers with varying dilation are stacked up together. Usual multiply and add operation of each layer is followed by dynamically gated non-linear activation function [1]. Each successive layer have increased dilation factor which allows the receptive field to grow exponentially.

The activations from each convolution layer are fed into another network with 1*1 convolutions and non-linear activation function, which then produces a continuous output. This output is fed into the network to produce the output of next time-step. This network is trained to reduce the ESR loss (error to signal ratio) between target and output signals. We used Adam Optimizer with constant learning rate.

We used 10 convolutional layers with with dilation pattern $1, 2, 4, \ldots, 1024$. This gives a receptive field of 2047 samples. Number of channels in each convolutional layer is set 12, which Wright et. al. [16] shows to be the optimal choice.

V. Results

A. Low Complexity Model-

Optimal Learning rate was chosen. The model used Adam Optimizer withfor the Marimba we observed $3e^{-3}$ to be appropriate, but for mapping Piano to Guitar we observed better results with a learning rate of $1e^{-5}$. We observed that the model performed poorly in modelling the stochastic start of a note, and had trouble when notes overlapped. Even though the dataset was monophonic there were sections where notes blended into each other due to non zero sustain and decay duration. These regions add additional non-linearities that our model struggles to emulate.

1) Piano-Marimba: Strengths of the transfer from Piano to Marimba was an accurate representation of high frequency content. Our model also contained a pre-emphasis filter that was necessary to preserve high frequency content when emulating distortion from a guitar, which doesn't contradict our results. The only other major artifact in this mapping was a high frequency buzzing noise. This is due to the fact that our WaveNet model could not remove all overtones present in the Piano input audio.

Naturally this motivates the question whether a simple low pass filter could compete with our WaveNet results as a well designed frequency filter could remove such noise directly from the piano audio with minimal artifacts. All attempts to tune such a filter on the DAW revealed that although low-pass filter on Piano was aesthetically pleasing, it retained more stochasticity from the piano and the WaveNet model sounded closer to a Marimba subjectively. This makes removal of the buzzing noise essential for successful timbre transfer. This makes timbre transfer it as a good use case for a more complex model.

2) Piano-Guitar: Similar to the Piano Marimba transfer we observed good quality frequency information retention, i.e. the FFT of the predicted signal and input signal were close in shape, but this transfer faced issues in producing high frequency information that did not exist in the original piano track. Where the guitar had frequencies ranging upto 8kHz, the Piano only had frequencies closer to 3.5kHz. This model could not construct information in the band between 3.5kHz to 8kHz that didn't exist in the piano audio. This emulation of guitar sound was also struggling to generate stochastic onset noises, and otherwise results were similar to what we observed for the Marimba. Here the buzzing noise could even be interpreted as distortion on a guitar so this reconstruction wasn't entirely without use.

B. Impact of Increasing Complexity - Dilation Depth

Dilation Depth both increases the perceptive field of the model and the depth of the model in a layer, due to its tree like structure. This translates to more history being available to the model at each prediction. Very low dilation depth implies poorer understanding of low frequency information. In the low complexity model we work with a dilation depth of 10, which implies a receptive field of 2047 samples at 44.1kHz. This corresponds to 0.046 seconds, or starting from 40Hz which is within human limits of 20-20kHz, and most acoustic instruments and compositions are not capable of producing significant sub 40Hz audio. We still increased dilation depth to 20 hoping this would provide the model more information about note onsets and note durations, as all relevant frequency information was already available to the network. This means means the perceptive field of one stack of dilated convolutions is roughly 20 seconds.

1) Piano-Marimba: There was no observable difference as the validation loss plateaued around 0.3 for the increase in dilation depth. This highlights a limitation of the Neural Net when dilation is varied keeping all other parameters fixed. Larger Dilation does not help.

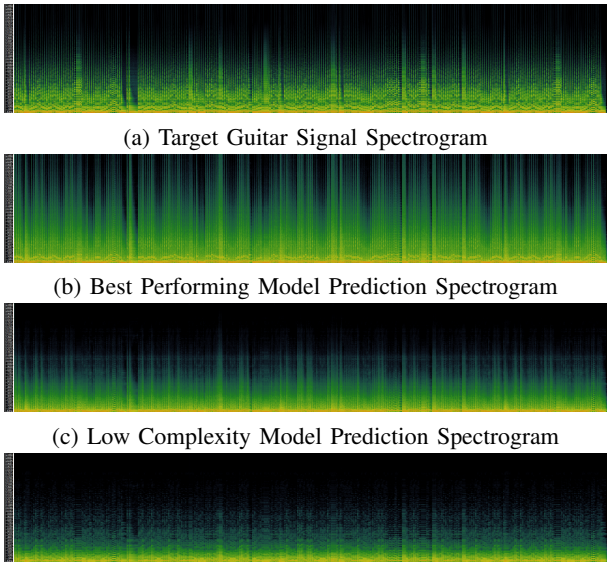
2) Piano-Guitar: We had more success on training a model on the guitar, which is more complex both in its note onset and overtones present. With less dilation layers we achieved a minima on validation loss of 0.6, contrasted with 0.45 for increased dilation layers. The model trained faster (on loss vs. time) than an equivalent model with lower dilation depth, although each epoch took considerably longer. This was further improved in the next section.

C. Impact of Increasing Complexity - Stacked Dilated Convolutions

So far our model grew in complexity by increasing the depth of the tree of dilations. This was extremely expensive computationally, both for training and for running the model as we ran out of RAM resources very quickly. Another approach to increasing perceptive field is to have more repetitions of such dilated convolution stacks. This reduces the depth of each layer, allowing us to build deeper neural networks that are less expensive computationally. Unlike the previous section where we increased depth to 20 layers by increasing dilation depth, we stick to the default dilation depth of 10 and stack 3 repetitions, effectively leading to a depth of 30 layers, whose dilation pattern is $1, 2, 4, \ldots, 1024, 1, 2, 4, \ldots, 1024, 1, 2, 4, \ldots, 1024$.

1) Piano-Marimba: After tuning hyper parameters we observed better performance by this model than all previous attempts in terms of validation error, reaching a loss of 0.20 compared to 0.30 with the shallow net. This improvement is not due to removal of buzzing from the audio but because of better reconstruction of stochastic note onsets. This leads to more defined notes during sections of rapid note interchange, also known as a trill in classical music terms. Previously the we'd hear distortion due to overlapping note tails and that effect is diminished.

2) Piano-Guitar: We observed faster tuning and lower validation loss compared to both the low complexity model and increased dilation depth model. This mapping still doesn't match the quality of Piano-Marimba transfer, as it achieved a best loss at 0.35. But on the positive side the model is now able to generate higher frequency audio signals that weren't available prior. Aside from the buzzing noise the model was reasonable at generating note onsets which was only possible due to increased model depth.



(a) Target Guitar Signal Spectrogram

(b) Best Performing Model Prediction Spectrogram

(c) Low Complexity Model Prediction Spectrogram

(d) Input Piano Spectrogram

Fig. 3. Validation Data Spectrograms. Note the lack of high frequency content in part (c) and the abundance of high frequency audio in part (b). Our best performing model (b) has stacked dilated convolutions. These results help us motivate stacked convolutions being necessary for meaningful timbre transfer.

VI. Discussion

The major strengths are -

1) Good spectral transfer (from existing frequencies in input)
2) Learns non-trivial characteristics like volume, and the ADSR curve well.

Major Drawbacks

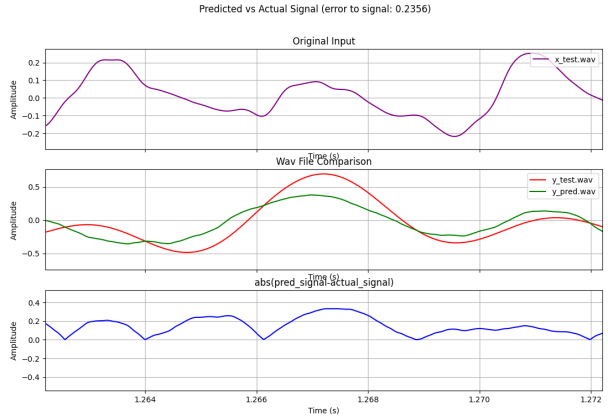
1) Artifacts like buzzing noise
2) Poor Performance on polyphonic audio samples
3) Cannot generate frequencies missing in the input
4) Can't transfer stochastic noises well in shallow nets

The buzzy noise is attributed to over-smoothing in features [13], due to a lack of depth of model. Such buzzing artifacts have also been reported in other research like [9] that report this noise is due to random perturbations in phase even when correct harmonic series are generated. Our model works well for preserving non-stochastic audio content, but our model finds note onsets difficult to map between instruments without increasing depth.

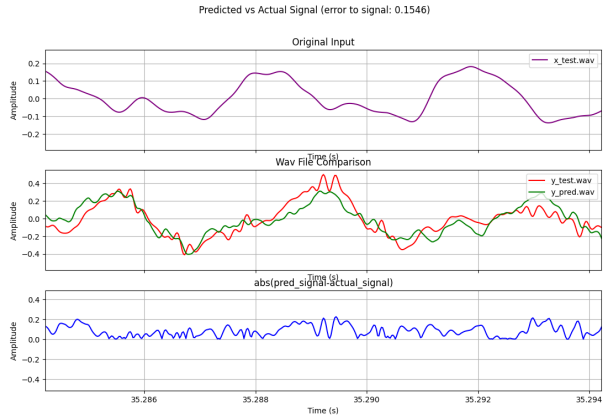
A point worth remembering is that the network [6] was initially optimised and validated for modelling distortion and overdrive effects for guitars. It is extremely successful at doing so leading to the its wide commercial application through the Finnish company Neural DSP. Not only is the task of modelling distortion a simpler non-linearity to model than timbre transfer, which is why networks produced extremely low losses in that domain, but also buzzing noises are inherent to the required output sound. So we cannot comment on limitations of this model to remove buzzing based on our own experience. These buzzing artifacts are also absent in other generative approaches that use images representations of audio, but those models face other artifacts due to down-sampling, quantisation and poor reconstruction capabilities of certain spectrogram representations.

We were not able to train the model on polyphonic samples sufficiently outside of initial tests on the shallow model. This remains an open problem worth investigating.

The fact that the shallow Model could not reconstruct frequencies that were missing in the input audio indicate that such a model is good for mapping instruments with a very bright timbre, which are instruments that contain a lot of information in higher frequencies, to instruments with darker timbres. As the shallow model was only successful at manipulating frequencies that it has available to it this represents a drawback of real time audio to audio mappings without first compressing to features or transforming to the Image domain. In the image domain even with shallow models we can explicitly provide the model space to add high frequency information by appropriate choice of spectrogram and window sizes. Encoding for different frequencies is not such a simple task when working with WaveNet, and it is more difficult to interpret how the network learns without visual representations



(a) Piano-Marimba Mapping



(b) Piano-Guitar Mapping

Fig. 4. Inferences from wav files generated by our best performing models (Stacked Dilate Convolutions). 0.01 second window at highest intensity location is shown with a comparison between x_test (input piano audio), y_test (output instrument audio) and y_pred (predicted audio). Note the retention of high frequency noise when absolute difference of input and output is compared for Guitar Audio. Phase errors are extremely evident in Marimba Audio, that are indicative of the buzzing noise.

of the impact of each layer that can be inferred in image based timbre transfer methods. Adding more layers or stacks of dilated convolution was sufficient to infer this detail, and holds promise for real time applications though it requires more compute. There is still a need to search less compute intensive models. This is a moderately strong conclusion.

Stochastic noises are clearly difficult to learn with a shallow network, as improvements were seen increasing depth. Differences in phase between input and output make for difficult non-linearities to model. As a contrast the task of modelling distortion is a lot more tangible in the relationship between input and output. Here the stochastic onset of one instrument may have nothing in common with the stochastic onset generated by another instrument. This stochasticity may vary between different notes played on the same instrument and techniques used to generate audio (e.g. playing a guitar with a plectrum vs. a finger). We believe the model should not learn to exactly duplicate the stochasticity using Signal to error ratio, but perhaps should learn to create a realistic sounding onset. This is worth investigating in the form of a modified loss function, perhaps using a more complex Discriminator Neural Network that learns to identify realistic sounding stochastic note onsets. This is relatively weaker inference as deeper networks performed better, so we cannot conclude that it is a flaw of the model.

For future work one could look at starting with an optimal bright pitch instrument (need not be a timbre that currently exists) to map to other instruments. Alternative methods that can lead to a deep but less compute intensive method are also worth investigating along with changes to the loss function, as error to signal ratio behaves similar to MSE loss while we might require a more domain specific learning.

ACKNOWLEDGMENT

We'd like to thank Krishna Subramani (Graduate 2020 Dept. of Electrical Engineering IITB) for engaging with us to discuss feasibility of undertaking such a project, and encouraging us with insights from his own research work, and guidance to the correct literature before we began our project.

REFERENCES

[1] A. van den Oord, S. Dieleman, H. Zen, K. Simonyan, O. Vinyals, A. Graves, N. Kalchbrenner, A. Senior, and K. Kavukcuoglu, "WaveNet: A Generative Model for Raw Audio," *arXiv e-prints*, p. arXiv:1609.03499, Sep. 2016.

[2] S. Huang, Q. Li, C. Anil, X. Bao, S. Oore, and R. B. Grosse, "TimbreTron: A WaveNet(CycleGAN(CQT(Audio))) Pipeline for Musical Timbre Transfer," *arXiv e-prints*, p. arXiv:1811.09620, Nov. 2018.

[3] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks," *arXiv e-prints*, p. arXiv:1703.10593, Mar. 2017.

[4] D. K. Jain, A. Kumar, L. Cai, S. Singhal, and V. Kumar, "Att: Attention-based timbre transfer," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–6.

[5] K. Kumar, R. Kumar, T. de Boissiere, L. Gestin, W. Zhen Teoh, J. Sotelo, A. de Brebisson, Y. Bengio, and A. Courville, "MelGAN: Generative Adversarial Networks for Conditional Waveform Synthesis," *arXiv e-prints*, p. arXiv:1910.06711, Oct. 2019.

[6] E.-P. Damskägg, L. Juvela, E. Thuillier, and V. Välimäki, "Deep learning for tube amplifier emulation," in *44th IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2019; Brighton; United Kingdom; 12-17 May 2019 : Proceedings*, ser. Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing. United States: IEEE, May 2019, pp. 471–475, iEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP ; Conference date: 12-05-2019 Through 17-05-2019.

[7] Y. A. Mejjati, C. Richardt, J. Tompkin, D. Cosker, and K. In Kim, "Unsupervised Attention-guided Image to Image Translation," *arXiv e-prints*, p. arXiv:1806.02311, Jun. 2018.

[8] J. Engel, C. Resnick, A. Roberts, S. Dieleman, D. Eck, K. Simonyan, and M. Norouzi, "Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders," *arXiv e-prints*, p. arXiv:1704.01279, Apr. 2017.

[9] J. Engel, C. Resnick, A. Roberts, S. Dieleman, D. Eck, K. Simonyan, and M. Norouzi, "Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders," *arXiv e-prints*, p. arXiv:1704.01279, Apr. 2017.

[10] K. Subramani, P. Rao, and A. D'Hooge, "Vapar synth - a variational parametric model for audio synthesis," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 796–800.

[11] M. Pasini, "MelGAN-VC: Voice Conversion and Audio Style Transfer on arbitrarily long samples using Spectrograms," *arXiv e-prints*, p. arXiv:1910.03713, Oct. 2019.

[12] L. Sun, S. Kang, K. Li, and H. Meng, "Voice conversion using deep bidirectional long short-term memory based recurrent neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4869–4873.

[13] T. Kaneko, H. Kameoka, K. Hiramatsu, and K. Kashino, "Sequence-to-sequence voice conversion with similarity metric learned using generative adversarial networks," in *Proc. Interspeech 2017*, 2017, pp. 1283–1287. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2017-970

[14] *A complete MIDI catalogue of works by Johann Sebastian Bach*, 2018 (accessed December 5, 2020). [Online]. Available: http://www.bachcentral.com/midiindexcomplete.html

[15] K. Bloemer and T. Koker, *Code for Wavenet Used*, 2020 (accessed December 9, 2020). [Online]. Available: https://github.com/GuitarML/PedalNetRT

[16] A. Wright, E.-P. Damskägg, L. Juvela, and V. Välimäki, "Real-time guitar amplifier emulation with deep learning," *Applied Sciences*, vol. 10, p. 766, 01 2020.

APPENDIX

The notebook used is derived from [15] and we used all of the python files provided. To listen to Audio Samples you can mail either Author for access to a drive folder containing audio. The drive folder is linked here.