Week 7 Report

Kieran Grant - 2357351G

What I've done this week

- Not had quite as much time dedicated to the project this week due to coursework, might be similar next week but should be back to normal by week 9.
- Read and summarised *Blind Reverb Matching (2020).*
  - Research paper released from iZotope and presented at DAFX 2020.
  - Short summary and takeaways below.
- Created a sample architecture diagram based on the one implemented in [1].
  - Image attached to end of report.
- Had a brief look at the code for the Style Transfer paper from last week.
  - Quite a lot going on, but think I understand the rough structure.
- Read and summarised *Lightweight and Interpretable Neural Modeling of an Audio Distortion Effect Using Hyperconditioned Differentiable Biquads (2021).*
  - Another paper from iZotope.
  - Authors use blocks of trainable biquad IIR filters to emulate a distortion pedal.
  - Model is hyperconditioned on user interpretable parameters (in the above case 6 of them).
    - Seems like an interesting alternative to exploring a latent space after model training.
  - Model is shown to perform almost as well as a WaveNet-based model, but with 1000x fewer parameters (210 vs 22960).

Questions

- I would be interested in hearing your feedback on how you think the project is going so far, and whether I am 'on track'?
  - Are there things that you think I should be doing more/less of (e.g. reading papers, implementing models etc)?
  - Anything you would like included/excluded in the weekly report?

Plan for next week

- Read more style transfer papers.
- Organise papers read/to be read into groups for interim report literature review.

Current state of project

- Not too much change from last week, continuing with background reading and thinking about where the project can fit in with the literature.

Blind Reverb Matching (2020)

Short Summary

This paper presents a method for 'reverb matching', that is, making an input audio signal sound as though it was recorded in the same acoustic space as a target audio signal. This is done by optimising the parameter settings of a reverb 'generator' - in practice, a VST plugin (Phoenix Verb). Two approaches are explored, one where the parameter values are estimated using a regression model, and the other which uses a classification model to predict a preset from a predefined list.

The model itself takes raw (reverberated) audio as input. This is transformed to the frequency domain using a STFT. Audio frames are grouped into subsequences which are then processed through a stack of bidirectional gated recurrent layers. These recurrent layers are followed by a fully connected layer and pooling. The output of the model is either the predicted preset or parameter settings and a mixing ratio between 0 and 1. The loss function is a weighted sum of MSE for the mixing ratio and either MSE (regression) or cross-entropy loss (classification) for parameter loss.

For data generation, the authors begin with a corpus of non-reverberated audio. For each example of audio in the dataset, one or more reverberation settings and mixing ratios are uniformly sampled. The effect is applied to the audio and the parameter settings stored. In the case of the classification task, a random preset is selected from a list selected by an audio expert.

Evaluation of model performance is based on speech audio from the VCTK speech corpus, which consists of subjects reading newspaper articles in a monotone-voice, the Berlin Database of Emotional Speech and the Surrey Audio-Visual Expressed Emotion Database. The actual evaluation of the regression and classification models consisted of MUSHRA listening tests and the models compared against human tuned parameter settings and random parameter settings (the anchor). In general, the regression model was shown to perform as well as the human tuned parameter settings, and better than the classification model.

Takeaways

- The approach presented seems quite generalisable - I imagine that reverb could easily be swapped out for any other audio effect with a similar pipeline.
- The authors mention that using ML for audio effect style transfer hadn't been widely researched up to that point - most of the relevant work focussed on tuning synthesiser parameter settings for audio matching.
  - The only previous method of 'reverb matching' relied on predicting impulse responses - which can only model linear time-invariant systems/acoustic spaces.
- The architecture seems a little bit more convoluted with gated recurrent layers - using CNNs with spectrograms seems like a slightly easier way to capture temporal information in audio encodings.

Architecture Diagram

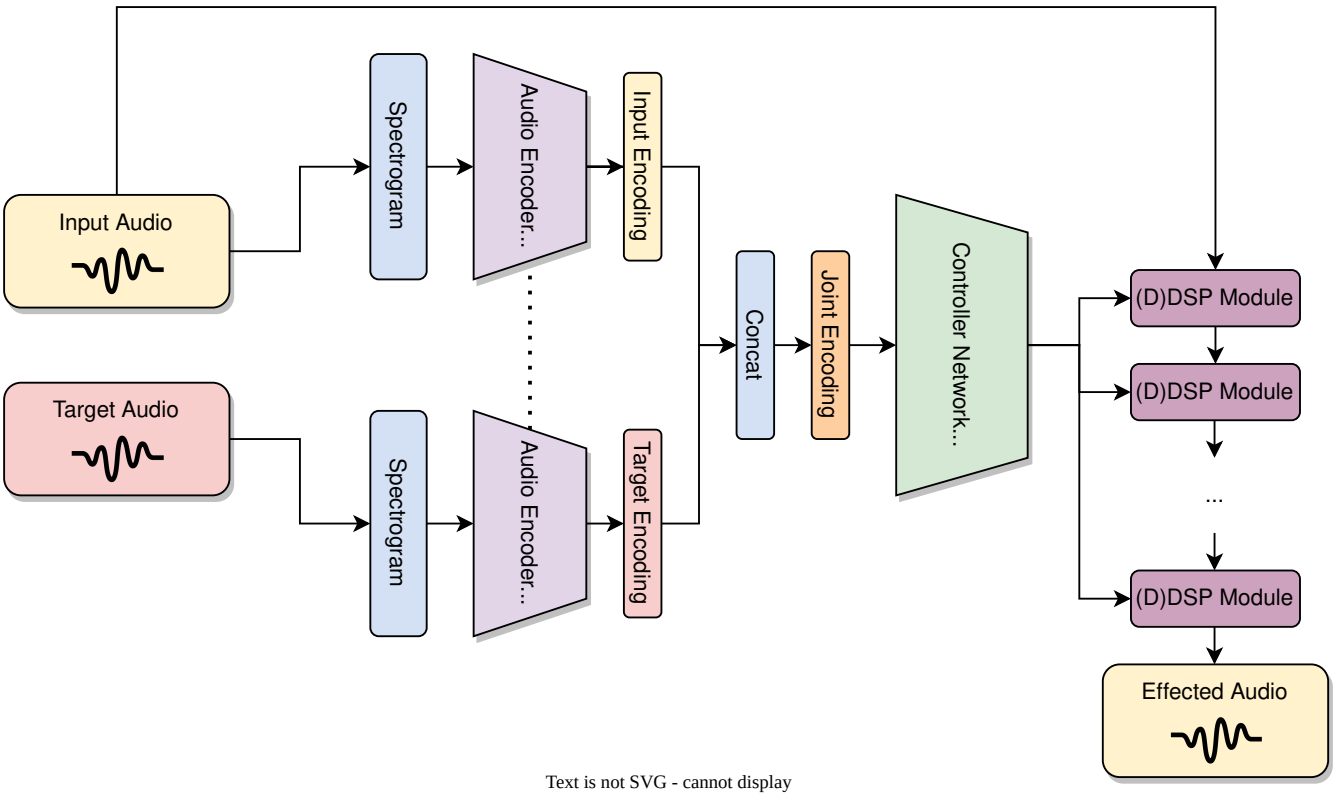

Figure: An example model diagram for style transfer based on the architecture in [1]. Here the audio encoder (f) is implemented with a CNN architecture (MobileNet/EfficientNet) and the Controller Network (g) is a simple feed-forward NN which predicts the P parameter settings for the DDSP modules.

References

[1] Steinmetz, Christian J., Nicholas J. Bryan, and Joshua D. Reiss. "Style Transfer of Audio Effects with Differentiable Signal Processing." arXiv preprint arXiv:2207.08759 (2022).