BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

Excel@FIT 2024
http://excel.fit.vutbr.cz

Machine Learning in Audio Effects

Jakub Sychra*

Abstract

Reverse engineering audio effects from mixed tracks is a complex topic that requires signal processing and music engineering experience. This work aims to create a system capable of identifying the sequence and parameters of guitar effects from a mixed audio track. Training data was created using clean guitar sounds from IDMT-SMT-Audio-Effects, augmented by known effects (BitCrush, Chorus, Clipping, Compressor, Delay, Distortion, High-pass filter, Ladder filter, Low-pass filter, Limiter, Phaser, and Reverb, all implemented with a Python wrapper around standard VST effects). The system is based on VGGish neural network architecture with several classification (presence of effects) and regression (parameters of effects) heads. The performance of the algorithm is evaluated on classification and regression accuracy, as well as in informal listening tests.

*xsychr06@stud.fit.vutbr.cz, *Faculty of Information Technology, Brno University of Technology*

1. Introduction

Audio effects are important building blocks of modern music, no matter the genre. The process of choosing effects does not follow rules, and as such this process can be highly unpredictable. Due to this complexity, reverse engineering audio effects affecting an instrument in a mixed track can be quite difficult.

Recreating a specific effect chain based on a sample requires experience in audio engineering, musical expertise, or both. With this knowledge, the process still relies on guessing based on specific features of audio effects, and to make the process worse, the sample usually contains more than one instrument, and thus some features may blend in with other instruments. Even with the correct guess for an effect, the result may be sensitive to the effect's parameters or position in the effect chain, adding further unpredictability.

The coverage of this topic in the literature is quite scarce, and while some papers exist, their scope is quite limited as they focus on either single effects without parameters or heavily controlled combinations of effects.

This work focuses on the detection of guitar effects and their parameters from mixed samples. For the purpose of extracting a guitar track from a mixed sample, this work uses the state-of-the-art Music Source Separator Demucs [1] to isolate the guitar track, which is then processed into a batch of Mel-spectrograms. This serves as input of the backbone model, which extracts the features of the samples and feeds them into a multi-head model containing classification heads and regression heads. Each classification head and regression head focuses on one specific effect, and thus all components of the model work individually, and are trained to be able to classify tracks containing any number of effects.

The final system performs fairly accurate classification of effects in samples containing numerous effects and even classifies effects outside of the controlled dataset, where the implementations of effects differ from the ones used in training. Parameter tuning is the critical point of the system, as no solution that could handle all of the parameter types of included effects was found, therefore the system handles only specific effects well, and as such this negatively affects the final reconstruction of the audio.

2. Audio Effects

The main purpose of audio effects is to modify the sound characteristic of the input signal. This modification is usually achieved by either manipulating the electrical signal directly using analog devices such as guitar pedals, or by utilizing software tools to alter

the digital signal that are often aimed at digitally replicating the effects of analog devices. These effects work in chains, and as some of them are non-linear systems, the order of effects is crucial to the final sound. In addition to the order, each effect has a certain amount of parameters that can significantly alter the sound of the effect.

The audio effects chosen for this work are: **BitCrush**, **Chorus**, **Clipping**, **Compressor**, **Delay**, **Distortion**, **High-pass filter**, **Ladder filter**, **Low-pass filter**, **Limiter**, **Phaser**, and **Reverb**.

These effects were chosen on the basis of being either popular or offering an interesting substitution that may fill space for unimplemented effects in reconstruction.

3. Data

For the purpose of training, a dataset containing known effects and their parameters was needed. For this purpose, a dataset was created and tailored for the needs of this work.

Based on the IDMT-SMT-Audio-Effects Dataset [2], from which clean guitar recordings were used as a basis for generating the effects dataset, using the Pedalboard [3] library that implements basic effects and allows the use of VST plugins in Python environment.

Data were created with a random number of effects with random parameter values while keeping track of these configurations to avoid duplicate samples.

A total of 110k audio samples were used in total duration of 61 hours, affected by 0-12 audio effects with varying number of parameters.

4. Proposed System

4.1 Preprocessing pipeline

As the system aims to detect guitar effects from mixed tracks, an important part of the system comes within the first steps of the pipeline with Source Separation. This module splits the incoming track into instruments and sends the isolated guitar further into the system. Such system is not flawless, and as such, the outgoing instruments might be affected by frequency cutoffs and artifacts, for example, faint echoes of vocals.

In the next step, the guitar track is processed into Mel-spectrograms within 0.960s windows that are fed into a backbone of the model, implemented using pre-trained VGGish [4] architecture.

4.2 Detection and Parameter estimation pipeline

Embeddings from the backbone model (VGGish) are processed using classification and regression heads. Each classification head has corresponding regression heads, and together, each group represents an abstract effect head. These effect heads are each trained on one specific effect with a set amount of parameters to detect and afterwards individually trained on data containing varying amounts of effects.

Classification heads are implemented using 3 linear layers followed by ReLU Activation.

Regression heads contain 4 linear layers followed by the ReLU Activation function and Batch normalization.

4.3 Reconstruction pipeline

From the detection result a text is compiled containing the predicted effects and their parameters as well as live audio feed with these effects active.

The position of these effects is not within the scope of this work.

Due to the implementation of the detected and reconstructed effects and the issues of reverse engineering the used library, the approach of altering audio stream directly was chosen over implementing a VST plugin.

5. Results

For measuring the detection accuracy, a separate test-set was defined. The obtained average effect classification accuracy was 75 %, while parameter estimation had an average mean error of 0.23 from target parameters. Parameters are scaled to the 0.00-1.00 range for consistency.

6. Conclusion and future work

The system implements a working pipeline that can currently estimate effects from tracks, although the reconstruction falls short on the inaccuracy of parameter detection.

Nevertheless, this work lays a solid foundation for further research in the field and offers potential avenues for improving the reconstruction system.

References

[1] Simon Rouard, Francisco Massa, and Alexandre Défossez. Hybrid transformers for music source separation. In *ICASSP 23*, 2023.

[2] Michael Stein. IDMT-SMT-audio-effects dataset, January 2023.

[3] Peter Sobot. Pedalboard, July 2021.

[4] S Hershey. CNN architectures for large-scale audio classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 131–135. IEEE, March 2017.