MUSIC INSTRUMENT DETECTION USING LSTMS AND THE NSYNTH DATASET

Brandi Frisbie
Center for Computer Research in Music and Acoustics, Stanford University, USA
brandi.l.frisbie@gmail.com

ABSTRACT

Music instrument recognition is an important part of Music Information Retrieval. Instrument detection could lead to better music tagging in recommendation systems, create better scoring for automatic music transcription tools, or even predict instrumentation of a piece of music based on a small sample. Most music instrument recognition research to date uses sound separation or other classification techniques. The recent release of the NSynth dataset, however, potentially constitutes a major breakthrough that can help mature the instrument recognition field. In this paper, I propose a framework for music instrument detection using LSTMs and the NSynth dataset. The NSynth dataset (over 280,000 samples from eleven different instruments) will be used as training data and IRMAS (over 2,800 excerpts) will be used for testing. After careful evaluation of the model, future work will involve expanding the datasets to include more instruments and further tuning the model to best fit the data.

1. INTRODUCTION

With recent advancements in machine learning and music and the release of new projects like Magenta, [1] we are closer than ever to solving Music Information Retrieval (MIR) tasks such as music instrument detection. The Google Brain team recently released the NSynth dataset, [2] a large-scale dataset of annotated musical notes. While many studies use NSynth for music generation, I believe that the dataset additionally adds value to the MIR community for a variety of topics such as music instrument recognition.

In previous applications of music instrument detection, other models are used such as SVM, K-NN, logistic regression [7], HMM, GMM [3], ICA, or non-negative matrix factorization for sound separation [5]. My process is unique in that it uses the NSynth dataset and an LSTM. A dataset comprising all possible notes of an instrument might allow a model to learn timbre. The model could be applied to a piece of music to identify the instruments present at a specific point in time. To my knowledge this approach has not been feasible before the release of NSynth and thus has not been attempted previously. I believe that NSynth makes this possible and I will describe a novel approach for the first step in determining the power of NSynth for music instrument detection.

2. DATASETS AND PREPROCESSING

The NSynth dataset will be used for training and the Instrument Recognition in Musical Audio Signals (IRMAS) [1] dataset for testing. Before implementing the model, both datasets require some data cleaning and preprocessing, as described below.

2.1 NSynth

The NSynth dataset contains prerecorded notes for a set of instruments in the range of a standard MIDI piano with different velocities and sample rates. This is significant because it simulates the variations of what a note can sound like in a live performance. As a result, it is well formatted as the foundation for instrument recognition. NSynth provides many unique features including labeling for instrumentation, note quality, and separation by notes, sample rate, and velocity.

Each .wav file contains four seconds of an individual instrument playing a single note and velocity. The dataset also provides tfrecord files for Tensorflow with predetermined features for easy input into a Tensorflow model. The training set contains 289,205 examples with the following instruments: bass, brass, flute, guitar, keyboard, mallet, organ, reed, string, synth_lead, and vocal.

2.1.1 Preprocessing

In order to setup NSynth for this project, the instrumentation samples of both NSynth and IRMAS need to be directly related. Bass, mallet, and synth_lead samples will be removed as there are no samples with these instruments in the IRMAS dataset. For the sake of the present experiment, only acoustic samples will be considered.

[1] https://magenta.tensorflow.org/welcome-to-magenta

[2] https://magenta.tensorflow.org/datasets/nsynth

Attribution: Brandi Frisbie. "Music Instrument Detection Using LSTMs and the NSynth Dataset", Extended abstracts for the Late-Breaking Demo Session of the 18th International Society for Music Information Retrieval Conference, Suzhou, China, 2017.

2.2 IRMAS

While NSynth is a great base for building models, it would need to be tested against real music to see if the model correctly learned to detect instruments. This brought me to IRMAS. IRMAS was created using raw .wav files with the goal of identifying predominant instruments in music [4]. The training data contains three-second clips of a musical excerpt with a predominant instrument. IRMAS samples most likely include multiple instruments with multiple notes being played at a time. This is one reason in which NSynth is distinct, as it only contains one instrument with one note per sample.

Ideally, any dataset could be used or created for the testing data. However, IRMAS was picked to start because it has a variety of genres and instruments. In addition, the testing dataset already divides files by stereo, length between five and twenty seconds, and excerpts with the same annotated instruments the entire excerpt. This allows IRMAS to provide a good basis for checking the error rate of this implementation before moving on to a larger scope.

2.2.1 *Preprocessing*

The IRMAS testing dataset comes with .wav files and corresponding text files with the predominant instrument labels. First, any samples with electric guitar will be removed because the guitar_electronic in NSynth is not acoustic. The IRMAS text file labeling needs to match with the labeling in NSynth. It is important to note that NSynth does not distinguish between string, brass, and reed instruments. For now, individual instruments will stay in their respective instrument groups. By listening to the NSynth samples, it is possible to distinguish some of the specific instruments, but relabeling will take careful evaluation and I plan on doing this in the next step of the project.

In order to have a testing dataset in the same format as the NSynth training data, the raw .wav files from the IRMAS dataset need to be converted into the tfrecord format with similar features. To do this, the same implementation as the NSynth setup [2] will be performed using WaveNet [9]. Once the data is in the correct format, we can begin with the model implementation.

3. LSTM APPROACH

For the model implementation, an LSTM will be used because one goal for this study would be to identify instruments in a piece of music in sequence or by a specific time in the song. LSTMs have outperformed other models in pattern recognition applications for speech [6], have been used to model polyphonic music with expressive timing and dynamics [8], and they have high potential for identifying the sequence of instruments in music. My implementation will be similar to the Magenta Polyphony RNN model, [3] where Adam Optimizer will be used for hyperparameter optimization and other factors will be tuned as the model is further evaluated.

4. CONCLUSION AND FUTURE WORK

In this paper, I have proposed a framework for instrument recognition in acoustical music. Recent advancements in MIR have brought music instrument detection in closer reach. My approach is a first step in determining the effectiveness of NSynth with the end goal of live instrument detection for an entire piece of music or for a point in time.

As next steps, I will implement my idea by converting the IRMAS .wav files to tfrecord files, train and test an LSTM model, evaluate the model, and iteratively tune the model. I may also create my own dataset and expand upon the instruments included in NSynth. More broadly, future research can include instrument recognition in acoustical or even live performances. There is potential to detect new instruments in music as well as non-traditional instruments based on sound combinations and we might even be able to identify and predict notes in music from a recording.

5. REFERENCES

[1] Juan J Bosch, Jordi Janer, Ferdinand Fuhrmann, and Perfecto Herrera. A comparison of sound segregation techniques for predominant instrument recognition in musical audio signals. In *ISMIR*, pages 559–564, 2012.

[2] Jesse Engel, Cinjon Resnick, Adam Roberts, Sander Dieleman, Douglas Eck, Karen Simonyan, and Mohammad Norouzi. Neural audio synthesis of musical notes with wavenet autoencoders. *arXiv preprint arXiv:1704.01279*, 2017.

[3] Antti Eronen et al. Automatic musical instrument recognition. Master's thesis, Mémoire de DEA, Tempere University of Technology, 2001.

[4] Ferdinand Fuhrmann et al. *Automatic musical instrument recognition from polyphonic music audio signals*. PhD thesis, Universitat Pompeu Fabra, 2012.

[5] Toni Heittola, Anssi Klapuri, and Tuomas Virtanen. Musical instrument recognition in polyphonic audio using source-filter model for sound separation. In *ISMIR*, pages 327–332, 2009.

[6] Haşim Sak, Andrew Senior, and Françoise Beaufays. Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition. *arXiv preprint arXiv:1402.1128*, 2014.

[7] Greg Sell, Gautham J Mysore, and Song Hui Chon. Musical instrument detection. Technical report, Center for Computer Research in Music and Acoustics, 2006.

[8] Ian Simon and Sageev Oore. Performance RNN: Generating music with expressive timing and dynamics, 2017.

[9] Aäron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew W. Senior, and Koray Kavukcuoglu. WaveNet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*, 2016.

[3] `https://github.com/tensorflow/magenta/tree/master/magenta/models/polyphony_rnn`