COMPARISON OF NETWORK ARCHITECTURES FOR POLYPHONIC GUITAR TRANSCRIPTION

Andrew Wiggins, Youngmoo Kim
Music and Entertainment Technology Laboratory, Drexel University, USA
{awiggins, ykim}@drexel.edu

ABSTRACT

In our previous work [1], we introduced a framework for the unsupervised transcription of solo acoustic guitar performances. The approach extends the technique used in DrummerNet [2], in which a transcription network is fed into a fixed synthesis network and is trained via reconstruction loss. Our initial tests to apply this technique to the problem of guitar transcription performed poorly, so in this work, we focus on improving the transcription part of the previously proposed framework. Here we compare the capabilities and limitations of two different transcription network structures for the task of polyphonic guitar transcription. To verify the plausibility the network structure in the unsupervised case, we investigate the task in the supervised setting, utilizing the limited labeled guitar data available in the GuitarSet dataset [3]. We find that the 2D CNN (Convolutional Neural Network) operating on input spectrograms from [4] is better suited to the guitar transcription task than the U-Net architecture based on 1D convolutions on raw audio used in [2]. In future work, we will leverage our insights regarding transcription network structure to improve upon our original unsupervised model.

1. INTRODUCTION

The guitar is a popular instrument among both professional and amateur musicians. While an experienced musician may be able to learn to perform a piece by simply listening to it, many guitarists seek out sheet music to learn via music notation. Because the process of creating sheet music from a recording can be time-consuming and requires expertise, we are interested in systems that can automate this task. Existing approaches to automatic guitar transcription require access to labeled training data, but unfortunately, existing datasets are limited in size and diversity. [5–9] Thus, we are interested in pursuing an unsupervised guitar transcription framework.

Our previous work introduced a scheme for unsupervised guitar transcription inspired by DrummerNet a recent, promising unsupervised approach to drum transcription by Choi et al [2]. Their network consisted of a trainable transcription module and a fixed synthesis module. The transcription module takes in audio and predicts a note transcription, and then the synthesis module resynthesizes audio from the predicted transcription. The network is trained to minimize the difference between the original audio and the audio reconstructed by the synthesizer. As a result, the network learns to produce accurate transcriptions.

Our initial attempts to extend this technique to the task of automatic guitar transcription performed poorly for the task of transcribing real-world guitar performances. In this work, we address the transcription part of the network, performing preliminary experiments to compare different possible network architectures in a supervised setting.

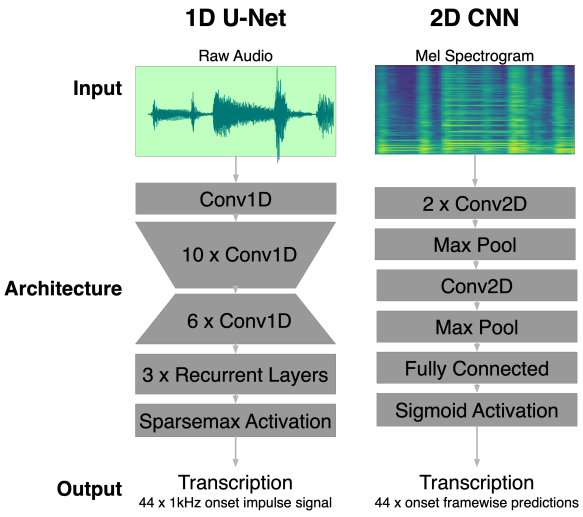

Figure 1. An overview of the two network architectures compared for the task of polyphonic guitar transcription. The 1D U-Net proved useful for drum transcription in [2], and the 2D CNN was used to predict note onsets in [4].

2. NETWORK ARCHITECTURES

Figure 1 provides an overview of the two network variations we explore for the task of polyphonic guitar transcription: a 1D U-Net used for drum transcription in [2] (which we used in our initial prototype) and a 2D CNN used for piano transcription in [4].

2.1 1D U-Net

The input to the U-Net is a 2-second clip of audio with a sampling rate of 16kHz. After an initial convolutional layer with 128 channels, the encoder consists of ten 50-channel convolutional layers interleaved with max-pooling layers of size 2. The decoder consists of six 50-channel convolutional layers interleaved with bi-linear interpolation layers of size 2. All convolutions have a kernel size of 3x3. Since we have 10 downsampling layers in the encoder and only 6 upsampling layers in the decoder, the output has one-sixteenth the sampling rate of the input.

Next, there is a set of 3 Gated-Recurrent Unit layers (GRUs), for sequence modelling. The first recurrent layer uses 100 channels and operates bi-directionally along the time axis to model temporal relations both forward and backward in time. The second recurrent layer uses 50-channels and is uni-directional, for modelling temporal dependency. Finally, the third recurrent layer is uni-directional, uses 44 channels (one for each pitch playable on the acoustic guitar), and operates along the pitch axis to model the dependencies between pitches on the guitar.

Finally, there is a Sparsemax activation along both axes. Sparsemax is an activation function that produces the sparsity in time and pitch that we expect given the physical limitations of a guitar performance (limited playing speed, and a maximum of 6 notes at a time). The output is a 1kHz signal with impulses representing onsets for each of the 44 guitar pitches.

2.2 2D CNN

The 2D CNN takes as input mel-scaled spectrograms with 229 frequency bins a hop size of 512 samples, and a sample rate of 16kHz. These mel spectrograms are compressed using a logarithm function.

The inputs are processed by a series of 2 3x3 convolutional layers each with 12 filters. Each of these is followed by a Rectified Linear Unit activation (ReLU). Next there is a max pooling of dimension 2 along the frequency bin axis only–the number of frames in each representation remains constant. This is followed by a 3x3 convolutional layer with size 24 and ending in a ReLU. Next, there is a fully connected layer with an output size of 44 for each frame. This represents the 44 playable pitches on a standard acoustic guitar. This model terminates with a sigmoid activation.

The output representation is a framewise onset prediction for each of the 44 guitar pitches. The frame rate matches that of the input, approximately 32 frames per second.

3. EXPERIMENT

We utilize the GuitarSet dataset [3] to train and test the two transcription architectures. GuitarSet includes performances from 6 different guitarists, so we build a train set using 5 of the guitarists and hold out the 6th for testing, as in [10]. All training and testing audio is segmented into 2-second clips. The training set consists of 4730 clips, and the test set contains 946 clips. We use a sample rate of 16kHz.

Using the note onset times and pitch labels included in the GuitarSet annotations, we create training labels in the appropriate formats for the two network variations. We train with a batch size of 32 using a binary cross entropy loss, and the 2D CNN makes use of batch normalization and dropout during training.

We compute precision, recall and f-measure for note transcription using mir-eval [11]. As per the mir-eval standard, predicted notes are considered correct if the onsets are within 50ms and the pitches are with 1 quarter tone of the target. These metrics are computed for each 2-second clip and then averaged over the entire testing dataset.

4. RESULTS AND DISCUSSION

Model	Precision	Recall	F-measure
1D U-Net	0.05 ± 0.02	0.56 ± 0.23	0.08 ± 0.04
2D CNN	0.84 ± 0.16	0.76 ± 0.19	0.78 ± 0.16

Table 1. The results from our experiments comparing the 1D U-Net and 2D CNN models. The means and standard deviations across the testing dataset are reported for all metrics.

The results from our preliminary experiments are shown in table 1. We found that the 2D CNN greatly outperformed the 1D U-Net in all metrics. In observing the U-Net's failure, it is important to note that the recall score is much larger than precision, indicating a large number of false positives. In the output transcriptions from this model, we observed excessive "pitch-streaking" where multiple pitches are detected simultaneously at each note onset. The DrummerNet's sparsemax activation layer does not insure single note detections as effectively for guitar signals.

Another reason for the U-Net's overall failure may be the many more parameters it has than the 2D CNN, which can provide training challenges given the limited amount of guitar data available. Additionally, while the U-Net is a more sophisticated model with mechanisms for incorporating details on multiple scales, it has the disadvantage of processing raw audio. The U-Net model may have been successful for the task of drum transcription since there were only 3 classes (kick, snare, hi hat) which are visually distinct in the time domain. However, with the task of guitar transcription there are 44 classes, for the 44 different pitches, and nearby pitches can appear to be very similar to one another in the time domain. The 44 pitch classes are perhaps more visually distinct in spectrogram representations. The mel spectrogram preprocessing pipeline in the 2D CNN is an advantage of this model.

In the future, we will leverage the insights from these preliminary comparison experiments to improve the transcription module within our unsupervised guitar transcription framework.

5. REFERENCES

[1] A. Wiggins and Y. Kim, "Towards unsupervised acoustic guitar transcription," *Late-Breaking Demo from International Society for Music Information Retrieval Conference, ISMIR*, 2020.

[2] K. Choi and K. Cho, "Deep unsupervised drum transcription," in *20th International Society for Music Information Retrieval Conference, ISMIR 2019*. International Society for Music Information Retrieval, 2019, pp. 183–191.

[3] Q. Xi, R. M. Bittner, J. Pauwels, X. Ye, and J. P. Bello, "Guitarset: A dataset for guitar transcription," in *Proceedings of the 19th International Conference on Music Information Retrieval (ISMIR), Paris, France*, 2018.

[4] C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck, "Onsets and frames: Dual-objective piano transcription," in *Proceedings of the 19th International Society for Music Information Retrieval Conference, ISMIR 2018, Paris, France, 2018*, 2018. [Online]. Available: https://arxiv.org/abs/1710.11153

[5] G. Burlet and I. Fujinaga, "Robotaba guitar tablature transcription framework," in *Proceedings of the 14th International Conference on Music Information Retrieval (ISMIR), Curitiba, Brazil*, 2013.

[6] G. Burlet and A. Hindle, "Isolated guitar transcription using a deep belief network," *PeerJ Computer Science*, vol. 3, p. e109, 2017.

[7] T.-W. Su, Y.-P. Chen, L. Su, and Y.-H. Yang, "Tent: Technique-embedded note tracking for real-world guitar solo recordings," *Transactions of the International Society for Music Information Retrieval*, vol. 2, no. 1, 2019.

[8] K. Yazawa, D. Sakaue, K. Nagira, K. Itoyama, and H. G. Okuno, "Audio-based guitar tablature transcription using multipitch analysis and playability constraints," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2013, pp. 196–200.

[9] K. Yazawa, K. Itoyama, and H. G. Okuno, "Automatic transcription of guitar tablature from audio signals in accordance with player's proficiency," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 3122–3126.

[10] A. Wiggins and Y. Kim, "Guitar tablature estimation with a convolutional neural network." in *20th International Society for Music Information Retrieval Conference, ISMIR 2019*. International Society for Music Information Retrieval, 2019.

[11] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, D. P. Ellis, and C. C. Raffel, "mir_eval: A transparent implementation of common mir metrics," in *In Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR*. Citeseer, 2014.