nnAUDIO: A PYTORCH AUDIO PROCESSING TOOL USING 1D CONVOLUTION NEURAL NETWORKS

Kin Wai Cheuk[1,2] **Kat Agres**[2] **Dorien Herremans**[1]

[1] Information Systems Technology and Design,
Singapore University of Technology and Design (SUTD), Singapore
[2] Institute of High Performance Computing,
Agency for Science, Technology and Research (A*STAR), Singapore
`kinwai_cheuk@mymail.sutd.edu.sg`

EXTENDED ABSTRACT

With the advent of Graphics Processing Units (GPUs), many computational methods are now taking advantages of this technology to dramatically reduce computation time. Theano [2], Tensorflow [1], Keras [3], and PyTorch [6] are well-known computational frameworks that leverage the power GPUs. Tensorflow has a `tf.signal` package that performs Fast Fourier Transform (FFT) and Short-Time Fourier Transform on GPUs. Since Tensorflow is notorious for its intricacy, there is a high-level API, called Keras, for people who want to quickly build a neural network without writing the low-level graphs for it. Kapre [4] is the Keras version for GPU audio processing. Along similar lines, PyTorch has recently developed torchaudio[1], but this tool has not been fully integrated into PyTorch at the time of writing. Further, torchaudio requires extra dependencies, and the installation often requires significant trouble-shooting [7]; for example, torchaudio is not currently compatible with Windows 10 [5]. Among the three tools, only Kapre and torchaudio support audio to Mel-spectrogram conversions, and Kapre is the only implementation that supports log-frequency spectrogram and trainable kernels for audio transformations. These, however, cannot be integrated with the popular machine learning library PyTorch. We present a GPU based audio processing tool, nnAudio, that supports the calculation of linear-frequency spectrogram, log-frequency spectrogram, Mel-spectrogram, and Constant Q Transform (CQT). nnAudio uses mainly one-dimensional (1D) convolution using PyTorch to do most of the audio processing and transformations, which implies that the transformation kernels can be integrated into a larger neural network and further trained/optimized to specific tasks. Our proposed library, nnAudio, is available online.[2]

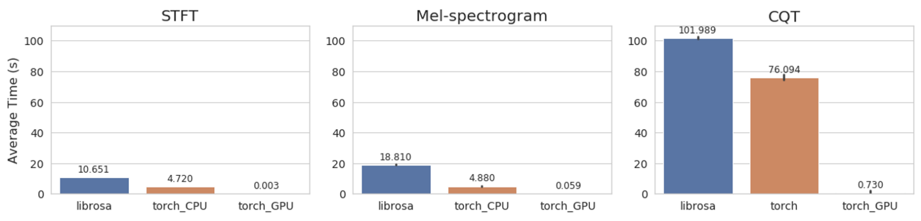

Figure 1. Computation time in seconds required to process 1,770 audio excerpts for different implementation techniques using a DGX with Intel(R) Xeon(R) CPU E5-2698, and 1 Tesla V100 DGXS 32GB GPU.

[1]https://github.com/PyTorch/audio
[2]http://dorienherremans.com/nnAudio

We use the MAPS dataset[3] to benchmark our audio processing tool. A total of 1,770 wav files from the AkPnBcht/UCHO/ folder were used for the benchmark. We cut out the first 20,000 samples from each audio excerpt to remove the silence. Each of the audio excerpts are kept the same length (80,000 samples) by removing the excessive samples in the end. Their final length is equivalent to 1.8 seconds. Fig 1 shows the time taken to extract spectrograms from 1,770 audio excerpts. It is clear from the figure that our newly proposed library is at least 100 times faster than traditional calculation.

The mathematical formula for Discrete Fourier Transform (DFT) at frequency bin k is given in Fig 2 (right bottom corner), which is very similar to the mathematical formula for 1D convolution $\sum_{s=0}^{N-1}$ input $*$ kernel. We can therefore implement DFT with two 1D convolutional neural networks (Conv1d), with $\cos(2\pi k\frac{s}{N})$ as the kernel for one Conv1d and $\sin(2\pi k\frac{s}{N})$ as the kernel for the other. Because Conv1d features a sliding window across the input signal, the mathematical operation automatically becomes a short-time Fourier Transform (STFT). Fig 2 shows how different types of spectrograms are obtained by STFT using convolution.

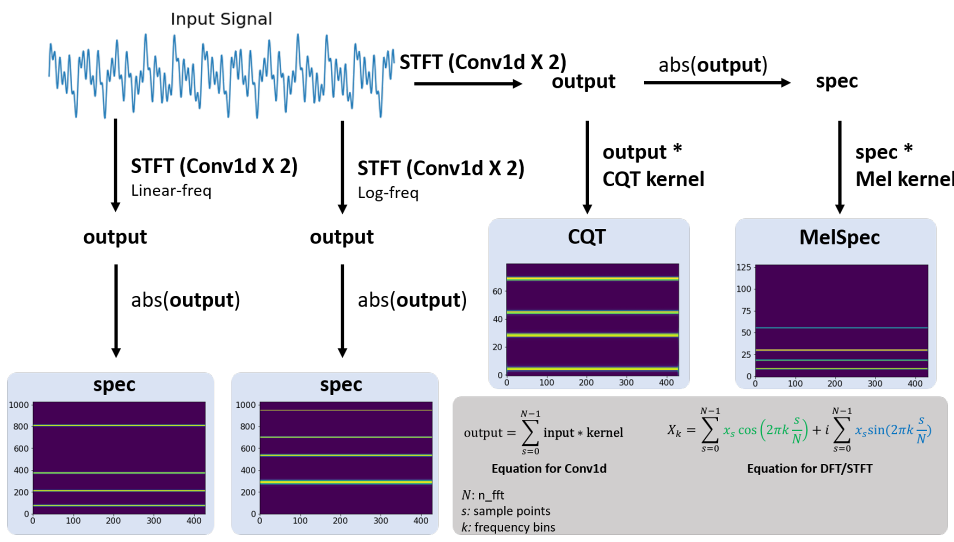

Figure 2. Workflow of calculating different spectrograms with nnAudio.

ACKNOWLEDGMENTS

This work is supported by IDC Grant no. IDIN18003, MOE2018-T2-2-161, and SING-2018-02-0204.

REFERENCES

[1] M. Abadi, Ashish Agarwal, Martín Abadi, et al. TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

[2] J. Bergstra and et. at. Theano: a cpu and gpu math expression compiler.

[3] P.W.D. Charles. Project title. `https://github.com/charlespwd/project-title`, 2013.

[4] K. Choi, D. Joo, and J. Kim. Kapre: On-gpu audio preprocessing layers for a quick implementation of deep neural network models with keras. *arXiv preprint arXiv:1706.05781*, 2017.

[5] L. Ericson. How to install torch audio on windows 10 conda? `https://stackoverflow.com/q/54872876`, 2019. Accessed: 2019-09-10.

[6] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in pytorch. 2017.

[7] K. Qu. some problems when installing torchaudio on mac. `https://stackoverflow.com/q/56659166/`, 2019. Accessed: 2019-09-10.

[3]https://www.tsi.telecom-paristech.fr/aao/en/2010/07/08/maps-database-a-piano-database-for-multipitch-estimation-and-automatic-transcription-of-music/