SPLEETER: A FAST AND STATE-OF-THE ART MUSIC SOURCE SEPARATION TOOL WITH PRE-TRAINED MODELS

Romain Hennequin, Anis Khlif, Felix Voituret, Manuel Moussallam
Deezer R&D, Paris
research@deezer.com

EXTENDED ABSTRACT

We present and release a new tool for music source separation with pre-trained models called Spleeter. Spleeter was designed with ease of use, separation performance and speed in mind. Spleeter is based on Tensorflow [1] and makes it possible to:

- separate audio files into 2, 4 or 5 stems with a single command line using pre-trained models.
- train source separation models or fine-tune pre-trained ones with Tensorflow (provided you have a dataset of isolated sources).

The performance of the pre-trained models are very close to the published state of the art and is, to the authors knowledge, the best performing 4 stems separation model on the common musdb18 benchmark [6] to be publicly released. Spleeter is also very fast as it can separate a mix audio file into 4 stems 100 times faster than real-time[1] on a single Graphics Processing Unit (GPU) using the pre-trained 4-stems model. Spleeter is packaged within Docker which makes it usable as is on various platforms.

Purpose: We release Spleeter with pre-trained state-of-the-art models in order to help the Music Information Retrieval (MIR) community leverage the power of source separation in various MIR tasks, such as vocal lyrics analysis from audio (audio/lyrics alignement, lyrics transcription...), music transcription (chord transcription, drums transcription, bass transcription, chord estimation, beat tracking), singer identification, any type of multilabel classification (mood/genre...) or vocal melody extraction. We believe that source separation has reached a level of maturity that makes it worth of consideration for these tasks and that specific features computed from isolated vocals, drums or bass may help increase performances, especially in low data availability scenarios (small datasets, limited annotation availability) for which supervised learning might be difficult. Spleeter also makes it possible to fine tune the provided state-of-the-art models in order to adapt the system to a specific use-case. Finally, having an available source separation tool such as Spleeter will allow researchers to compare performances of their new models to a state-of-the-art one on their own private datasets instead of musdb18, which is usually the only used dataset for reporting separation performances for unreleased models. Note that we cannot release the training data for copyright reasons, and thus, sharing pre-trained models were the only way to make these results available to the community.

Implementation details: Spleeter contains pre-trained models for:

- vocals/accompaniment separation.
- 4 stems separation as in SiSec [7] (vocals, bass, drums and other).
- 5 stems separation with an extra piano stem (vocals, bass, drums, piano and other). It is, to the authors knowledge, the first released model to perform such a separation.

The pre-trained models are U-nets [2] and follows similar specifications as in [5]. The U-net is a encoder/decoder Convolutional Neural Network (CNN) architecture with skip connections. We used 12-layer U-nets (6 layers for the encoder and 6 for the decoder). A U-net is used for estimating a soft mask for each source (stem). Training loss is a L_1-norm between masked input mix spectrograms and source target spectrograms. The models were trained on Deezer internal datasets (noteworthily the Bean dataset that was used

[1] We note, though, that the model cannot be applied in real-time as it needs buffering

in [5]) using Adam [3]. Training time took approximately a full week on a single GPU. Separation is then done from estimated source spectrograms using soft masking or multi-channel Wiener filtering.

Training and inference is implemented in Tensorflow which makes it possible to run the code on Central Processing Unit (CPU) or GPU.

Speed: As the whole separation pipeline can be run on a GPU and the model is based on a CNN (which makes computation parallelization very efficient), model inference is very fast. For instance, Spleeter is able to separate the whole musdb18 test dataset (about 3 hours and 27 minutes of audio) into 4 stems in less than 2 minutes, including model loading time (about 15 seconds), and audio wav files export, using a single GeForce RTX 2080 GPU, and a double Intel Xeon Gold 6134 CPU @ 3.20GHz (CPU is used for mix files loading and stem files export only). Spleeter is then able to separate into 4 stems 100 seconds of stereo audio in less than 1 second, which makes it very useful for efficiently processing large datasets.

Separation performances: The models compete with the state of the art on the standard musdb18 dataset [6] while it was not trained, validated or optimized in any way with musdb18 data. We report results in terms of standard source separation metrics [9], namely Signal to Distorsion Ratio (SDR), Signal to Artifacts Ratio (SAR), Signal to Interference Ratio (SIR) and source Image to Spatial distortion Ratio (ISR), are presented in Table 1 compared to Open-Unmix [8] which is, to the authors knowledge, the only released system that performs near state-of-the-art performances. We present results for soft masking and for multi-channel Wiener filtering (applied using Norbert [4]). As can be seen, for most metrics Spleeter is competitive with Open-Unmix and especially on SDR for all instruments.

	vocals				bass				drums				other			
	SDR	SIR	SAR	ISR	SDR	SIR	SAR	ISR	SDR	SIR	SAR	ISR	SDR	SIR	SAR	ISR
Spleeter Mask	6.55	15.19	6.44	**12.01**	5.10	10.01	5.15	9.18	5.93	12.24	5.78	10.50	4.24	7.86	4.63	9.83
Spleeter MWF	**6.86**	**15.86**	**6.99**	11.95	**5.51**	10.30	5.96	**9.61**	**6.71**	**13.67**	**6.54**	**10.69**	**4.55**	**8.16**	**4.88**	**9.87**
Open-Unmix	6.32	13.33	6.52	11.93	5.23	**10.93**	**6.34**	9.23	5.73	11.12	6.02	10.51	4.02	6.59	4.74	9.31

Table 1. 4 stems separation results. Bold font indicates highest metric values.

Spleeter is available on github[2] with a permissive license. This repository will be possibly used for releasing other models with improved performances or models separating into more than 5 stems in the future.

REFERENCES

[1] Martín Abadi et al. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.

[2] Andreas Jansson, Eric J. Humphrey, Nicola Montecchio, Rachel Bittner, Aparna Kumar, and Tillman Weyde. Singing voice separation with deep u-net convolutional networks. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 323–332, 2017.

[3] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. *arXiv e-prints*, page arXiv:1412.6980, Dec 2014.

[4] Antoine Liutkus and Fabian-Robert Stöter. sigsep/norbert: First official norbert release, July 2019.

[5] Laure Prétet, Romain Hennequin, Jimena Royo-Letelier, and Andrea Vaglio. Singing voice separation: A study on training data. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 506–510, May 2019.

[6] Zafar Rafii, Antoine Liutkus, Fabian-Robert Stöter, Stylianos Ioannis Mimilakis, and Rachel Bittner. The MUSDB18 corpus for music separation, December 2017.

[7] Fabian-Robert Stöter, Antoine Liutkus, and Nobutaka Ito. The 2018 Signal Separation Evaluation Campaign. *arXiv e-prints*, page arXiv:1804.06267, Apr 2018.

[8] Fabian-Robert Stöter, Stefan Uhlich, Antoine Liutkus, and Yuki Mitsufuji. Open-unmix - a reference implementation for music source separation. *Journal of Open Source Software*, 2019.

[9] Emmanuel Vincent, Rémi Gribonval, and Cédric Fevotte. Performance measurement in blind audio source separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1462–1469, July 2006.

[2] `https://github.com/deezer/spleeter`