Github Repo:
https://github.com/deezer/spleeter

Google Colab Demo:
https://colab.research.google.com/github/deezer/spleeter/blob/master/spleeter.ipynb

Dataset used for training and testing

The models were trained on Deezer's internal datasets (noteworthily the Bean dataset that was used in (Prétet et al., 2019)) using Adam (Kingma & Ba, 2014). Training time took approximately a full week on a single GPU.

Preprocessing of inputs

Module for building data preprocessing pipeline using the tensorflow data API. Data preprocessing such as audio loading, spectrogram computation, cropping, feature caching or data augmentation is done using a TensorFlow dataset object that output a tuple (input_, output) where:

- input is a dictionary with a single key that contains the (batched) mix spectrogram of audio samples
- output is a dictionary of a spectrogram of the isolated tracks (ground truth)

Evaluation metric performance

The models were tested on the standard musdb18 dataset (Rafii et al., 2017).

Standard source separation metrics are used for evaluation purposes(Vincent, Gribonval, & Fevotte, 2006)

- Signal to Distortion Ratio (SDR)
- Signal to Artifacts Ratio (SAR)
- Signal to Interference Ratio (SIR)
- Source Image to Spatial distortion Ratio (ISR)

Open-Unmix (Stöter, Uhlich, Liutkus, & Mitsufuji, 2019) and Demucs (Défossez, Usunier, Bottou, & Bach, 2019) are the only released system that performs near state-of-the-art performances. As can be seen, for most metrics Spleeter is competitive with Open-Unmix and especially on SDR for all instruments, and is almost on par with Demucs.
The Spleeter results were presented with soft masking and with multi-channel Wiener filtering (applied using Norbert (Liutkus & Stöter, 2019))

	Spleeter Mask	Spleeter MWF	Open-Unmix	Demucs
Vocals SAR	6.44	6.99	6.52	7.00
Vocals ISR	12.01	11.95	11.93	12.04
Bass SDR	5.10	5.51	5.23	6.70
Bass SIR	10.01	10.30	10.93	13.03
Bass SAR	5.15	5.96	6.34	6.68
Bass ISR	9.18	9.61	9.23	9.99
Drums SDR	5.93	6.71	5.73	7.08
Drums SIR	12.24	13.67	11.12	13.74
Drums SAR	5.78	6.54	6.02	7.04
Drums ISR	10.50	10.69	10.51	11.96
Other SDR	4.24	4.55	4.02	4.47
Other SIR	7.86	8.16	6.59	7.11
Other SAR	4.63	4.88	4.74	5.26
Other ISR	9.83	9.87	9.31	10.86

Model used/ Libraries used

It comes in the form of a Python Library based on Tensorflow, with the following pre-trained models.

Spleeter pre-trained models:

- vocals/accompaniment separation
- 4 stems separation as in SiSec (Stöter, Liutkus, & Ito, 2018) (vocals, bass, drums and other).
- 5 stems separation with an extra piano stem (vocals, bass, drums, piano, and other)

The pre-trained models are U-nets (Jansson et al., 2017) and follow similar specifications as in (Prétet, Hennequin, Royo-Letelier, & Vaglio, 2019).

https://ejhumphrey.com/assets/pdf/jansson2017singing.pdf
https://arxiv.org/abs/1906.02618