Spleeter by deezer

The journey to open-sourcing
the code and models

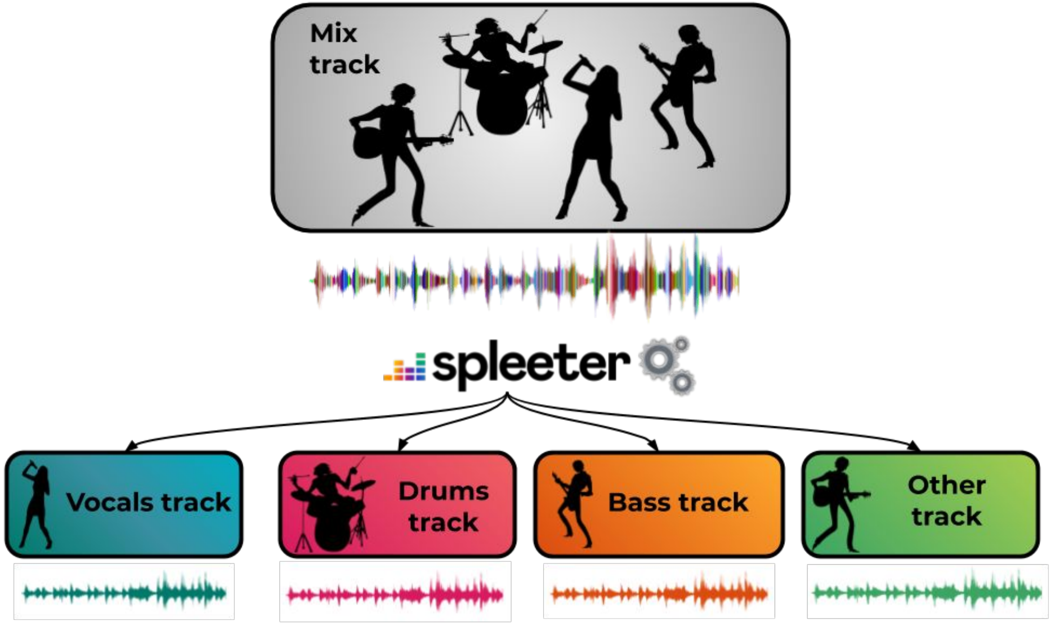
FOSDEM 2020

Anis Khlif, Félix Voituret

Who's been involved

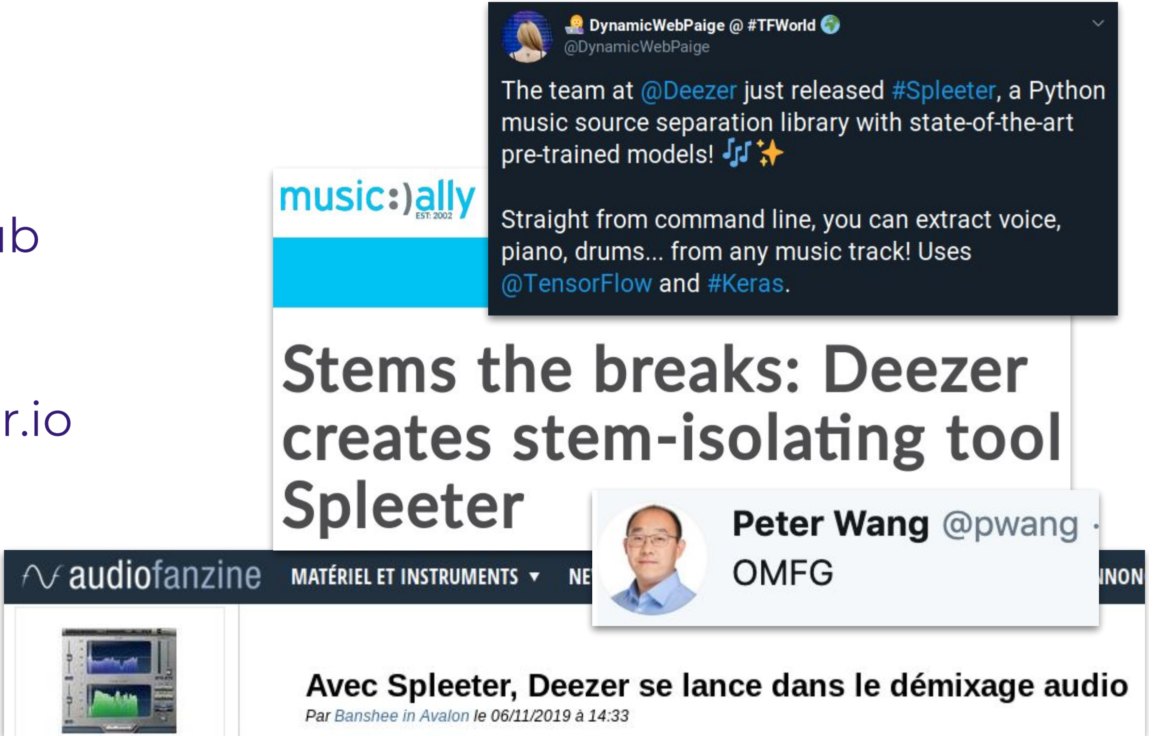
- Romain Hennequin - *Lead research scientist*
- Laure Prétet - *Former intern*
- Anis Khlif - *Research Engineer*
- Félix Voituret - *Research Engineer*
- Manuel Moussallam - *Head of Deezer Research*

What is it all about ?



Large impact on tech audience

- **9500+** stars on Github
- **200k+** views
- **100k+** read on deezer.io



DynamicWebPaige @ #TFWorld
@DynamicWebPaige

The team at @Deezer just released #Spleeter, a Python music source separation library with state-of-the-art pre-trained models! 🎶✨

Straight from command line, you can extract voice, piano, drums... from any music track! Uses @TensorFlow and #Keras.

Stems the breaks: Deezer creates stem-isolating tool Spleeter

Peter Wang @pwang ·
OMFG

Avec Spleeter, Deezer se lance dans le démixage audio
Par Banshee in Avalon le 06/11/2019 à 14:33

Myth busting

- ☒ Deezer solved source separation
- ☒ Spleeter performs better than all other solutions

What did we bring ?

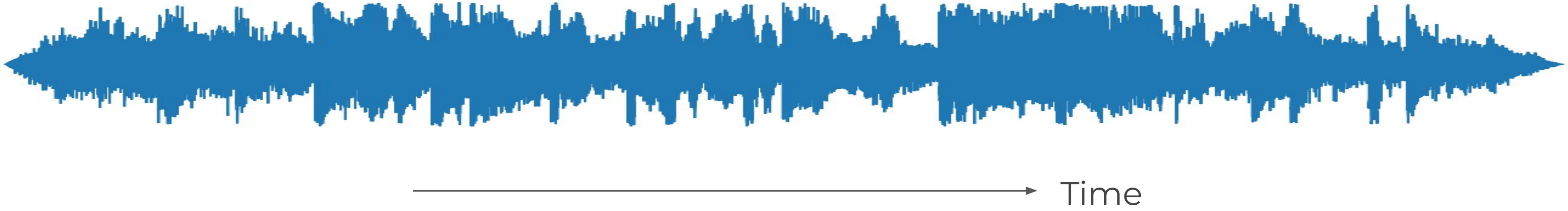
State of the art

Fast

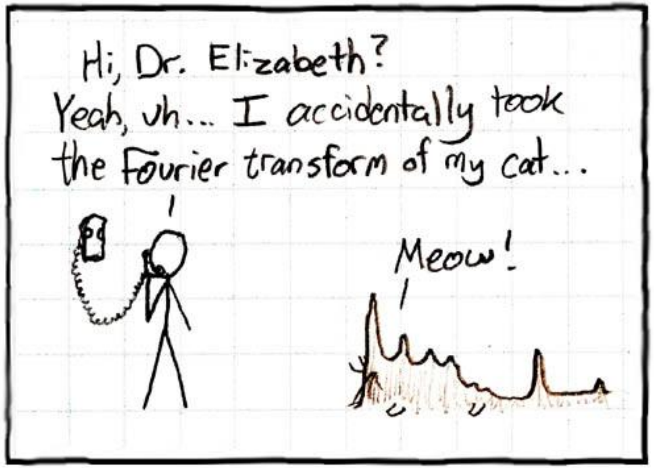
MIT Licensed

Primer on source separation

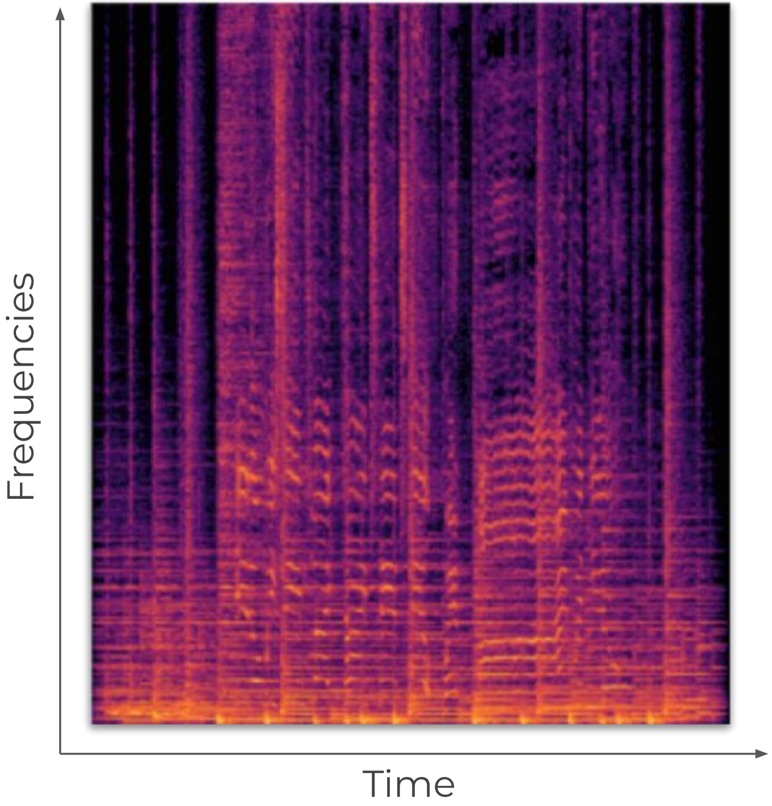
Waveform



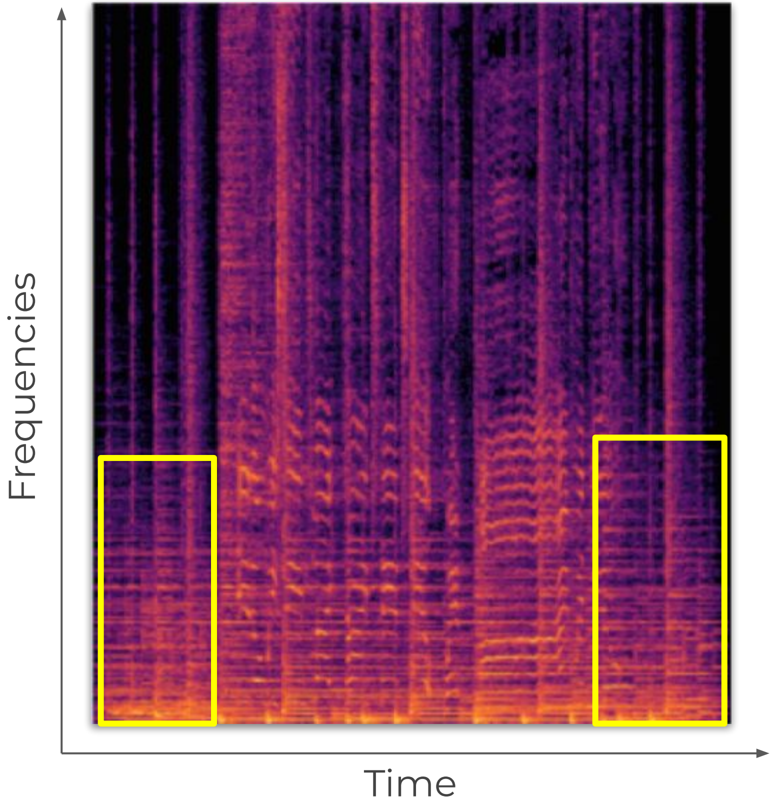
Time-frequency representation


Hi, Dr. Elizabeth?
Yeah, uh... I accidentally took the Fourier transform of my cat...
Meow!

Magnitude spectrogram

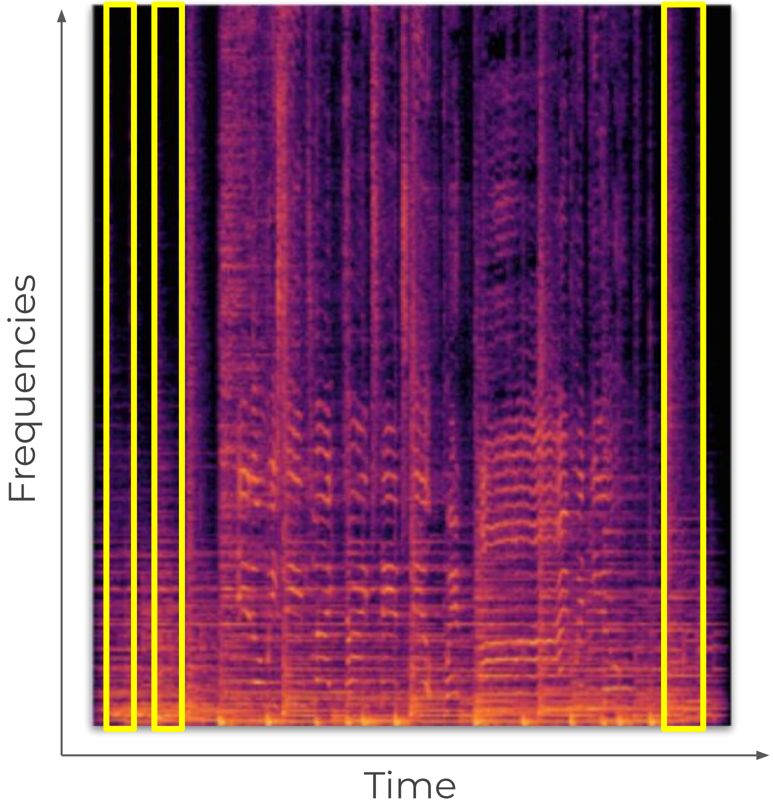


Magnitude spectrogram



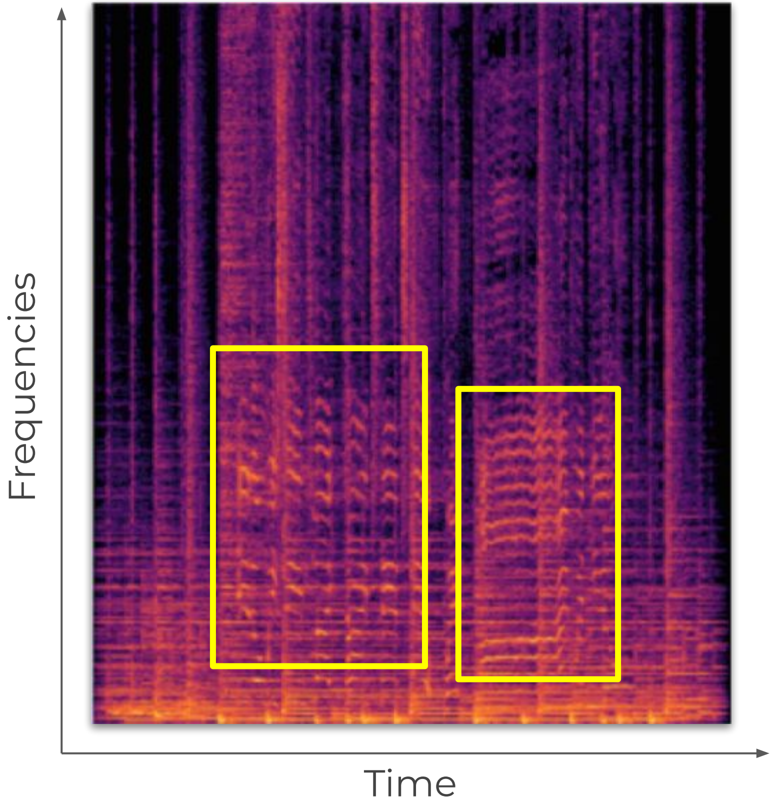
Harmonic content

Magnitude spectrogram



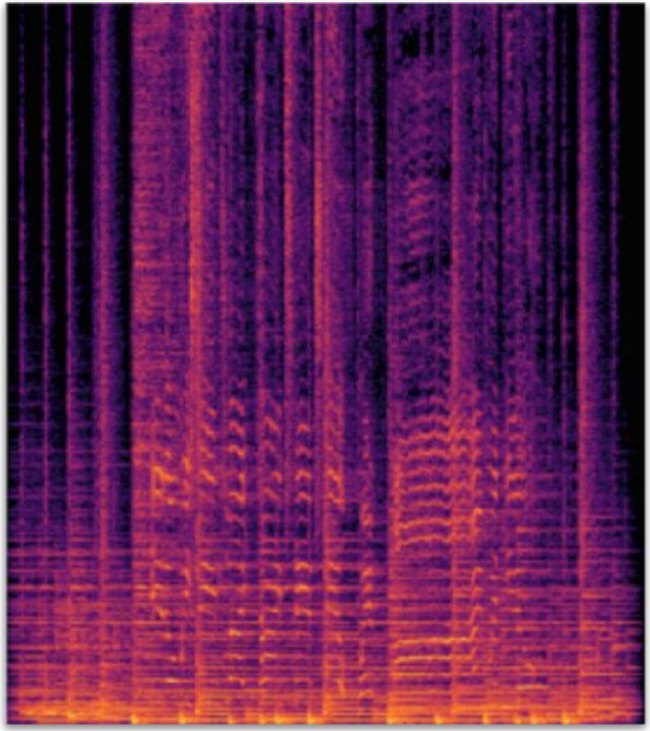
In-harmonic
Percussive content

Magnitude spectrogram



Vocal content

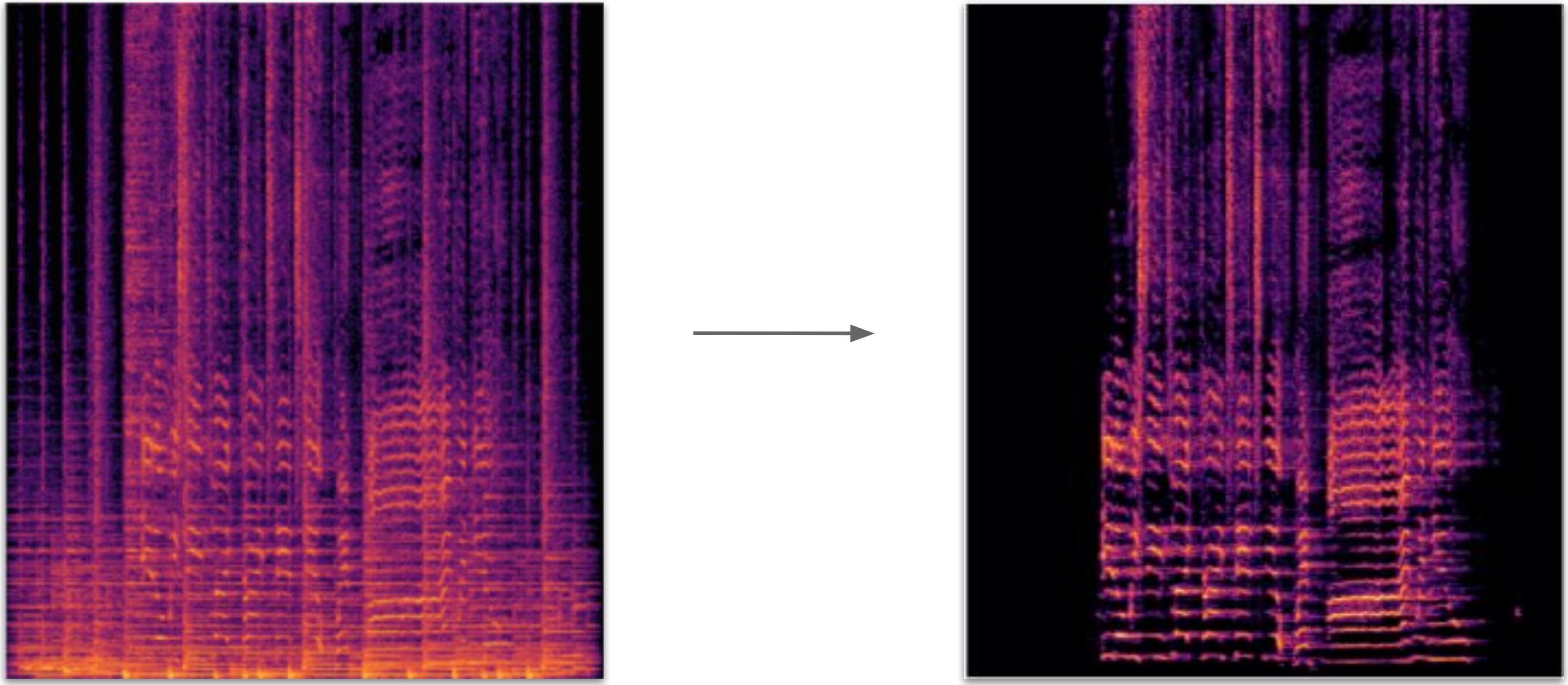
Magnitude spectrogram



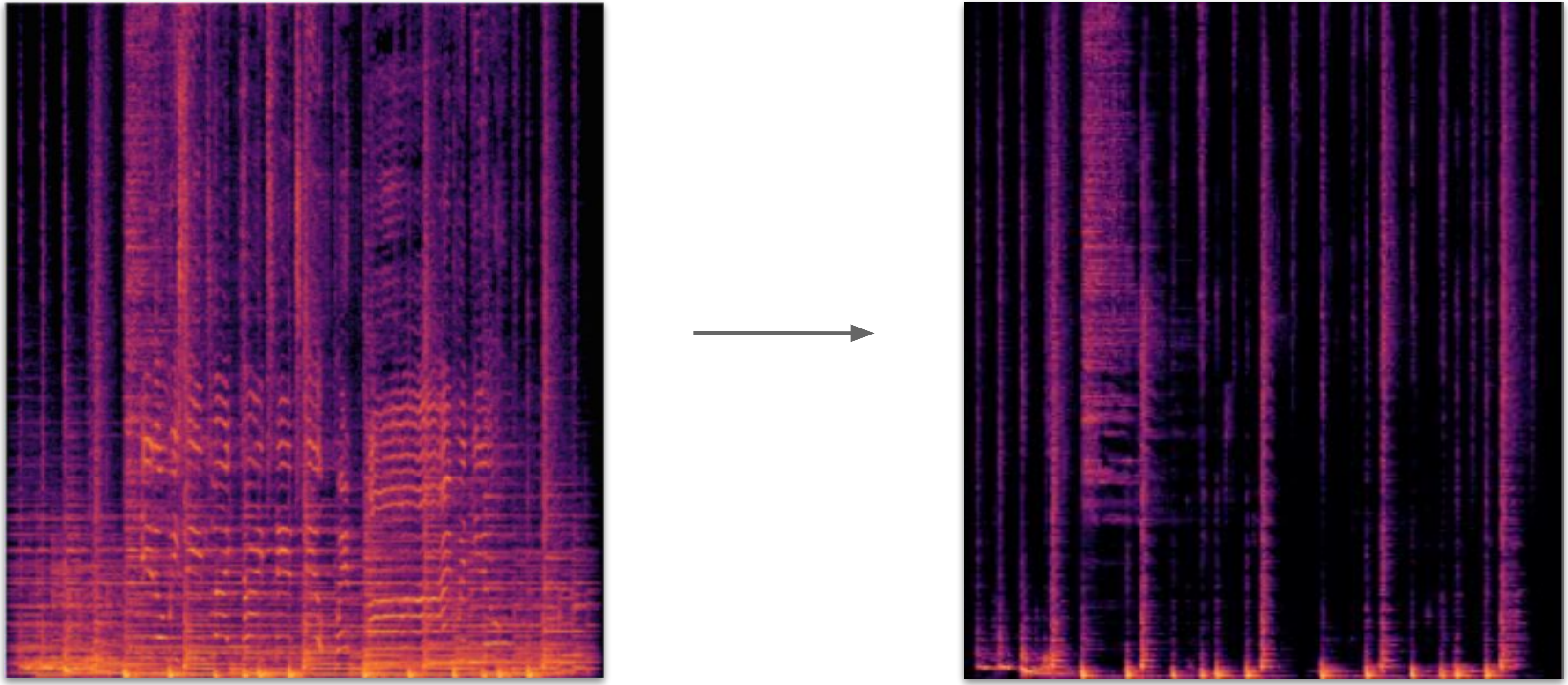
Learn a **mask** for each instrument !

What fraction of the energy at each time and each frequency bin should be assigned to this instrument.

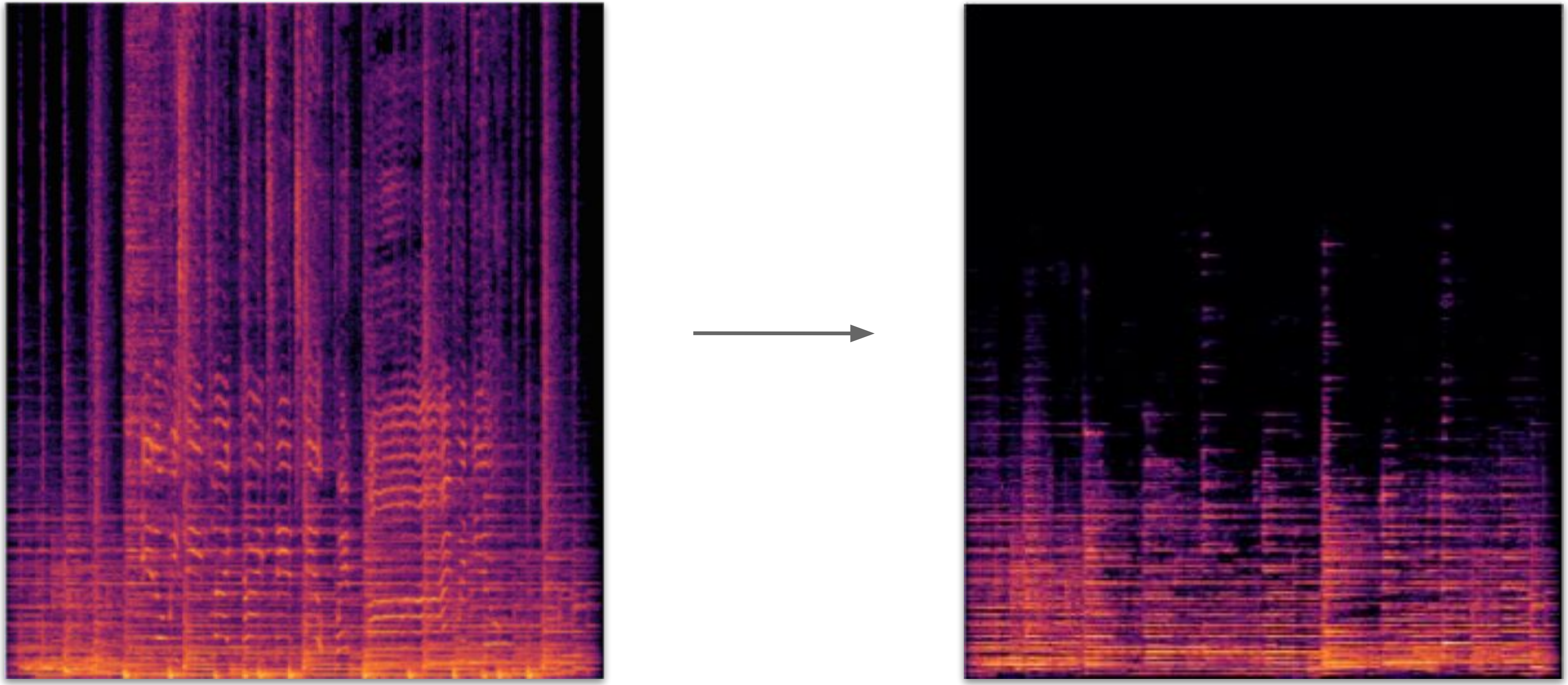
Magnitude spectrogram



Magnitude spectrogram

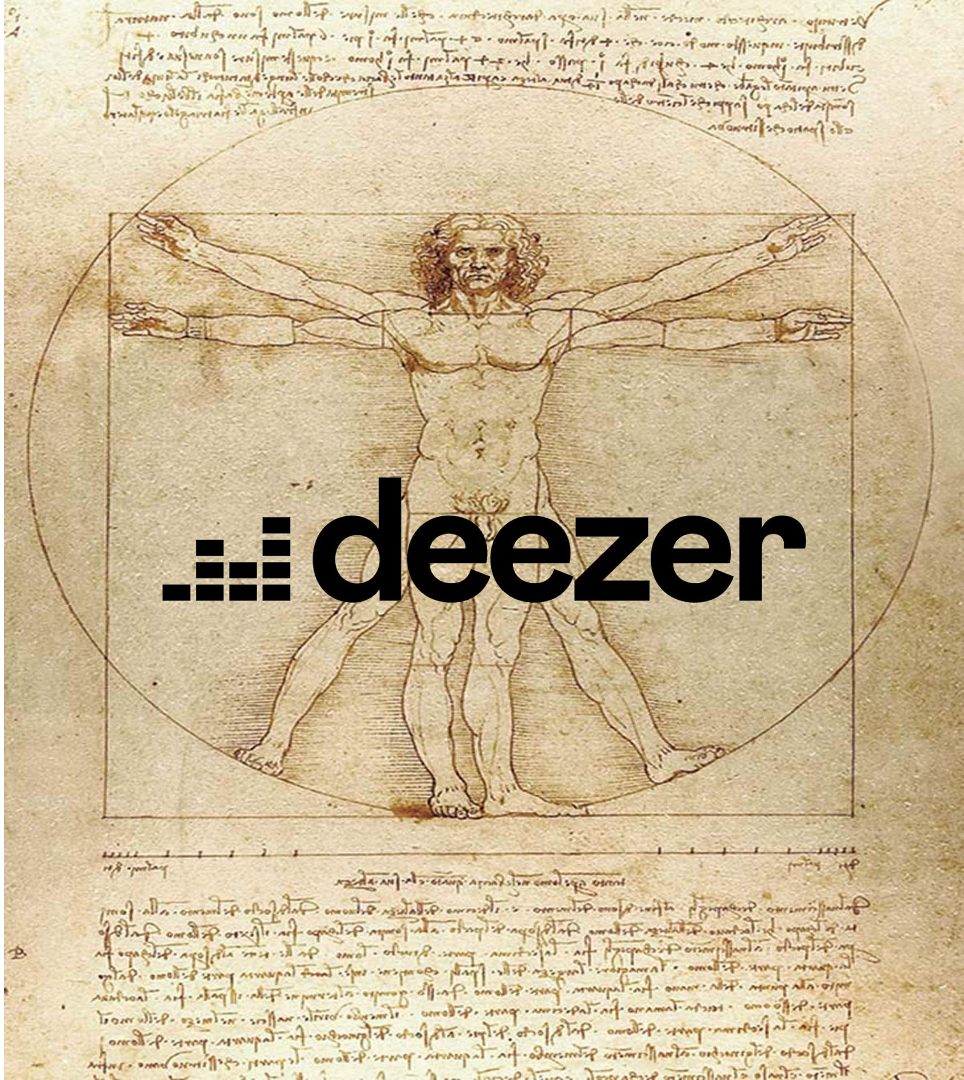


Magnitude spectrogram

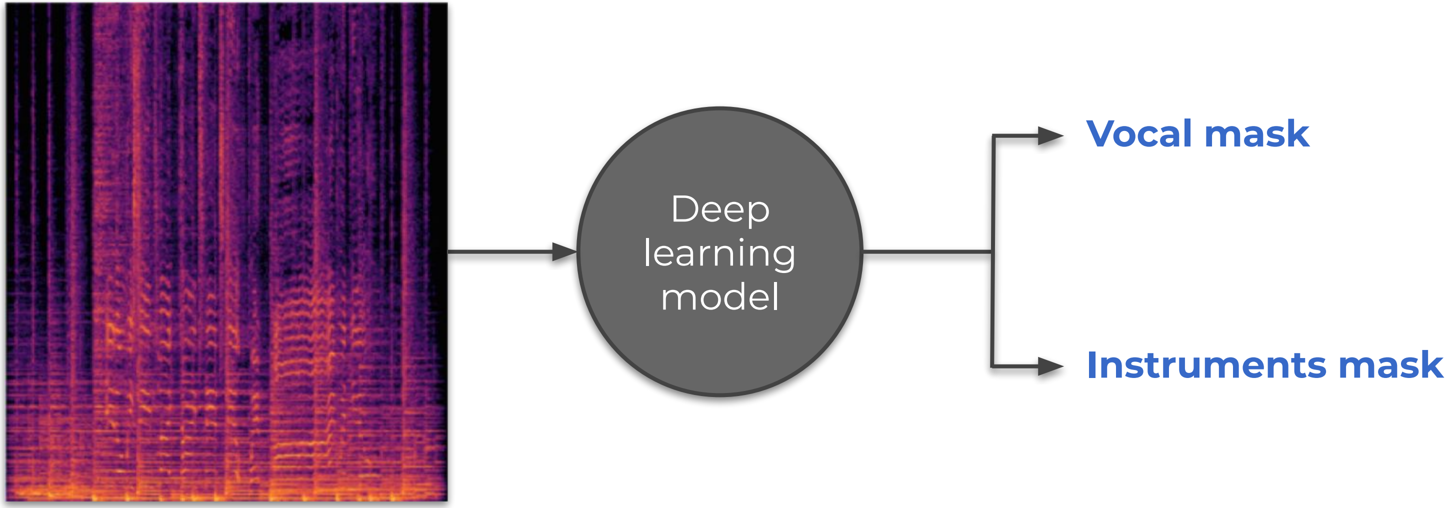


Spleeter models

2, 4 & 5 stems



A deep learning approach to mask prediction

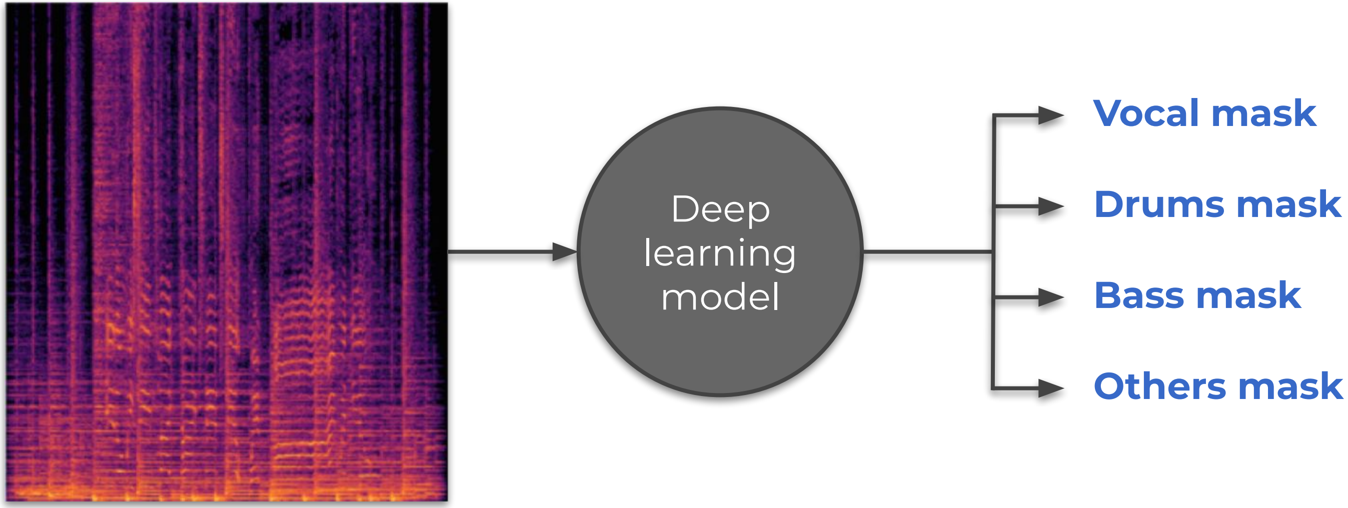


Deep learning model

Vocal mask

Instruments mask

4-stems



Deep learning model

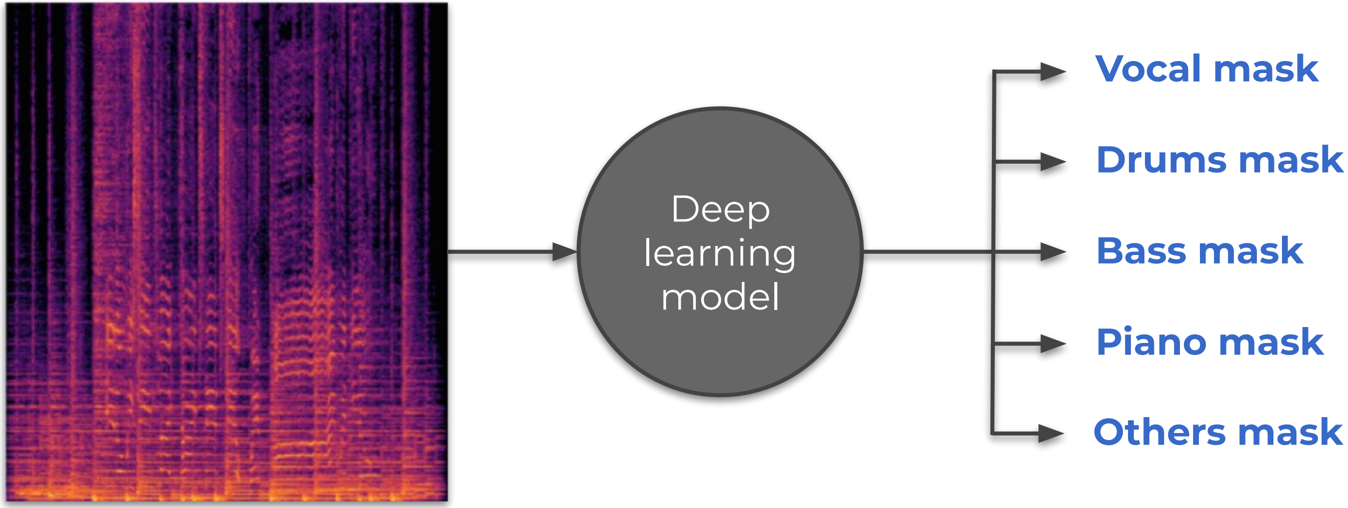
Vocal mask

Drums mask

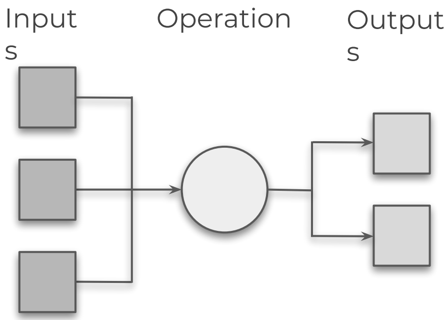
Bass mask

Others mask

5-stems

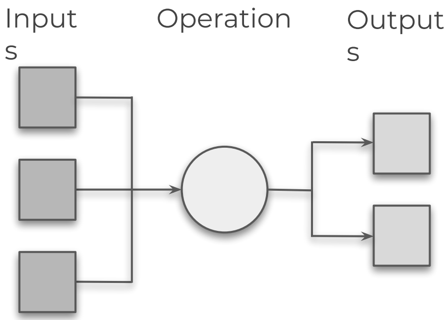

Deep learning model
Vocal mask
Drums mask
Bass mask
Piano mask
Others mask

Quick introduction to TensorFlow



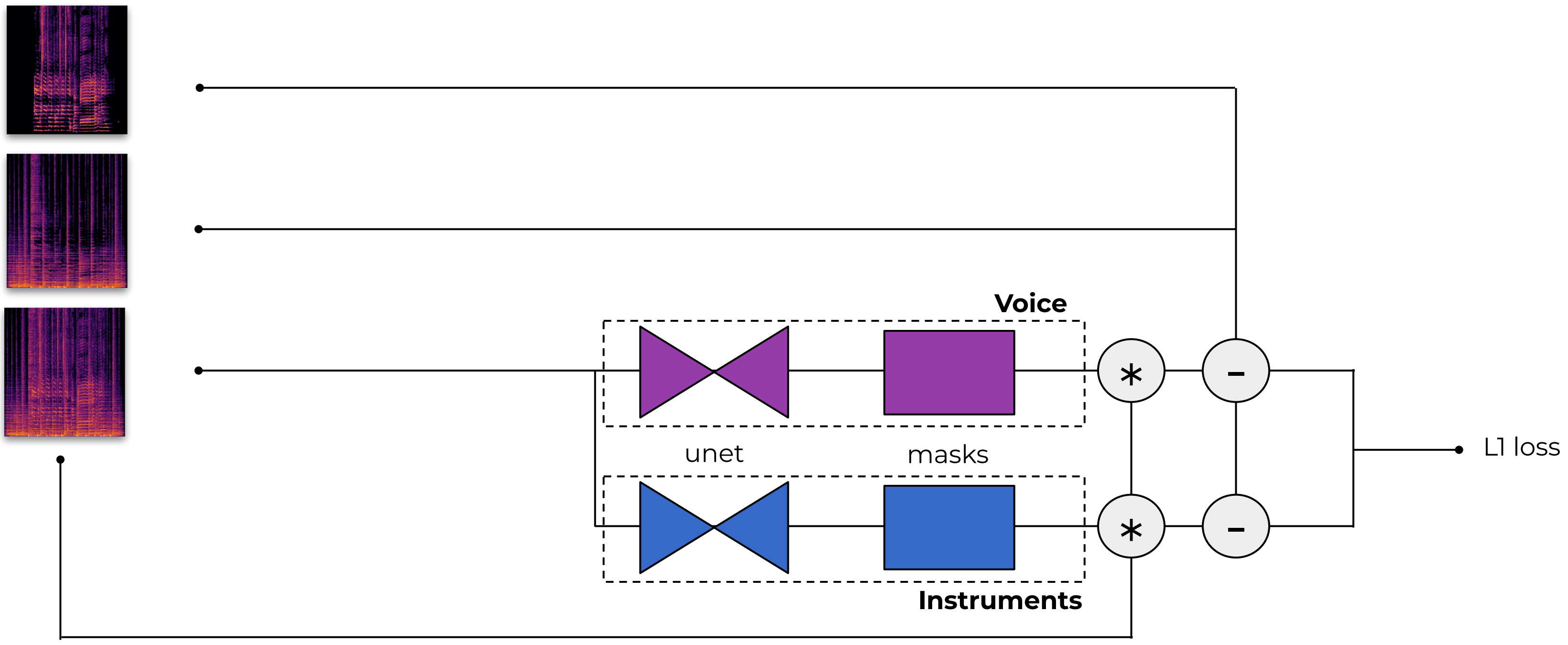
- Build **computation graph** that represent a parametrized function
- Parameters (or **weights**) can be modified (trained) to fit an optimization function
- A **model** can be run in any tensorflow environment
- Some graph operations can be run very efficiently on GPU

Quick introduction to TensorFlow



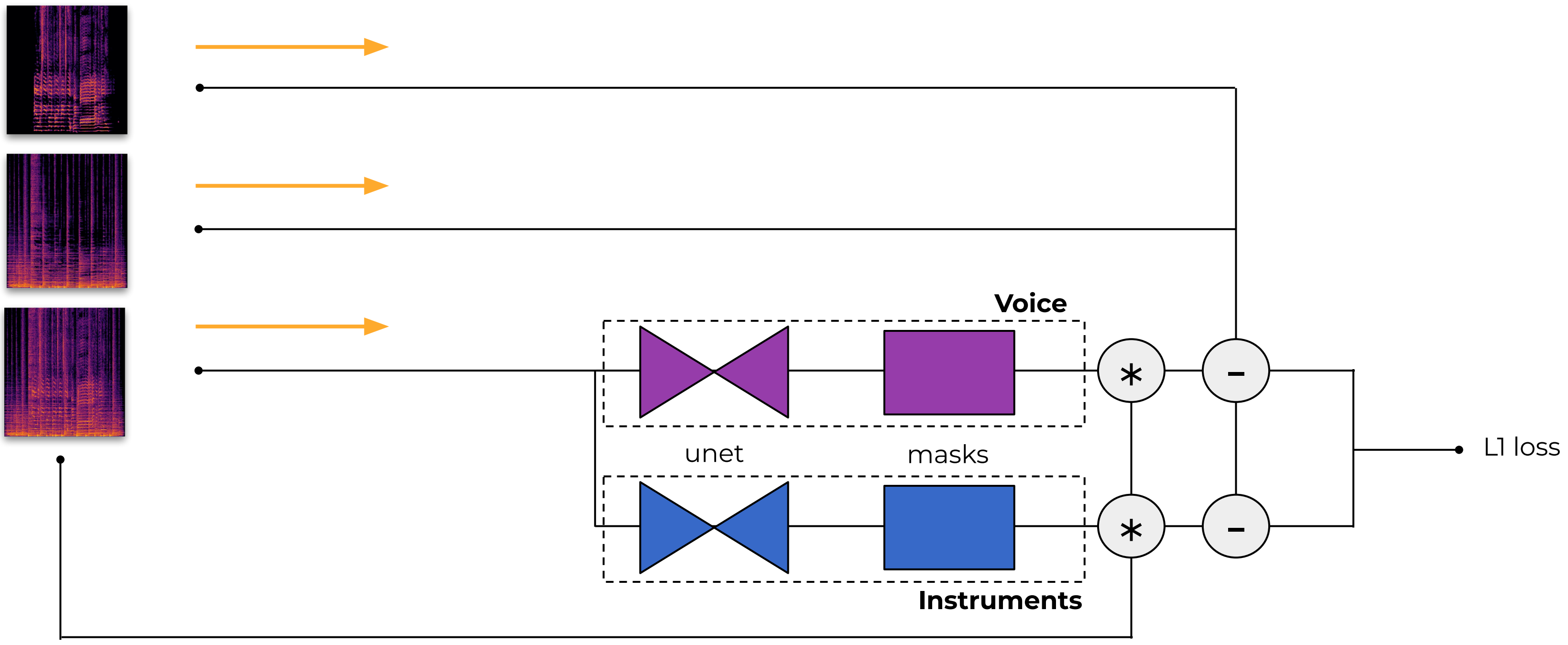
model = **computation graph** (network architecture) + **weights** (parameters)

Overview



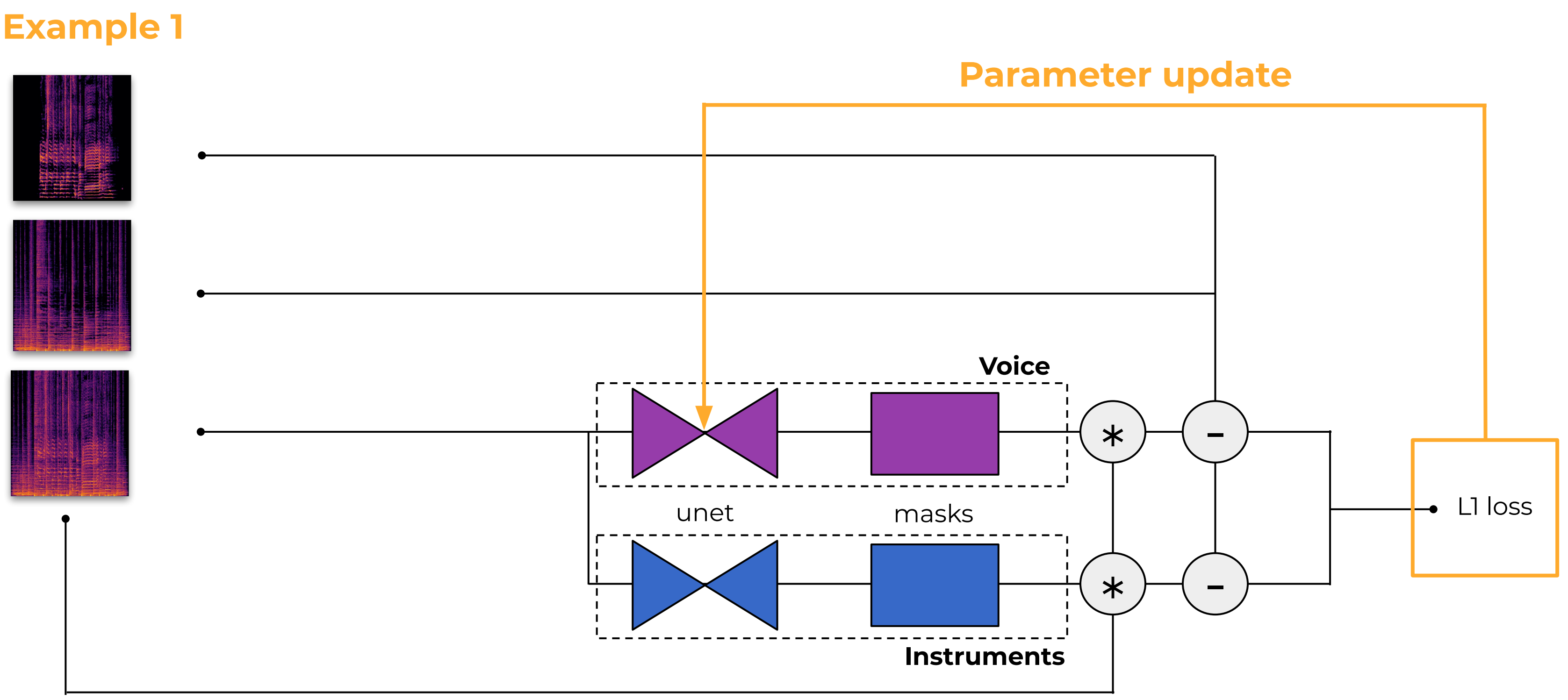
Overview

Example 1



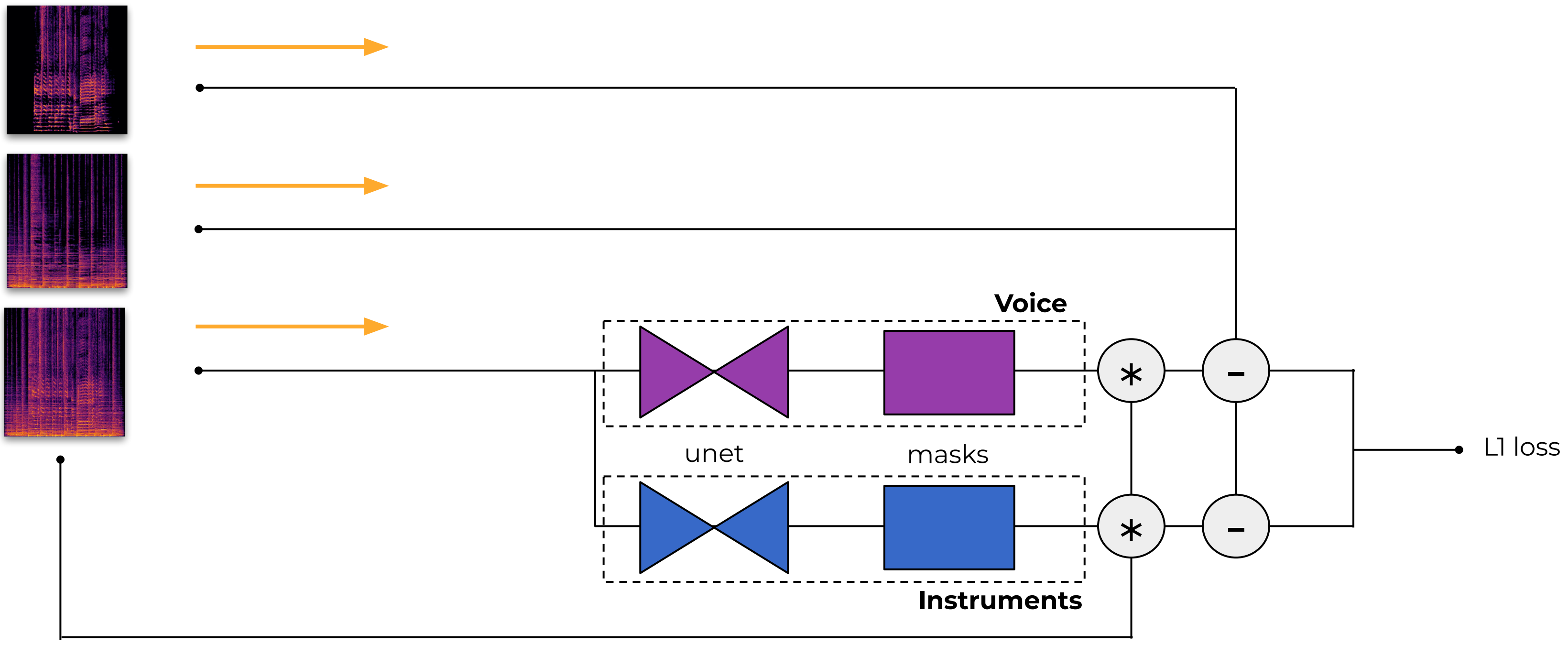
Spleeter models

Overview

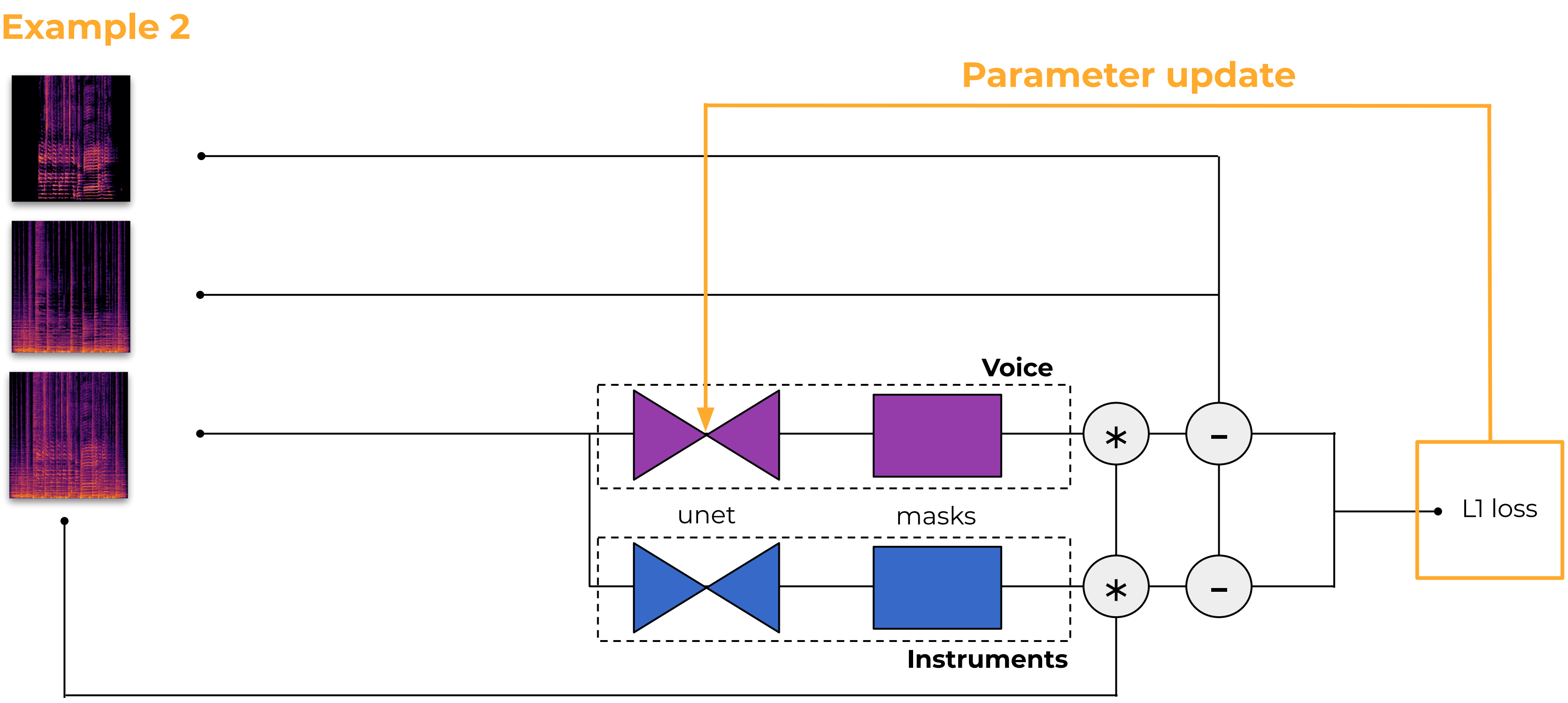

Example 1
Parameter update
Voice
unet
masks
Instruments
L1 loss
*
*
-
-

Overview

Example 2

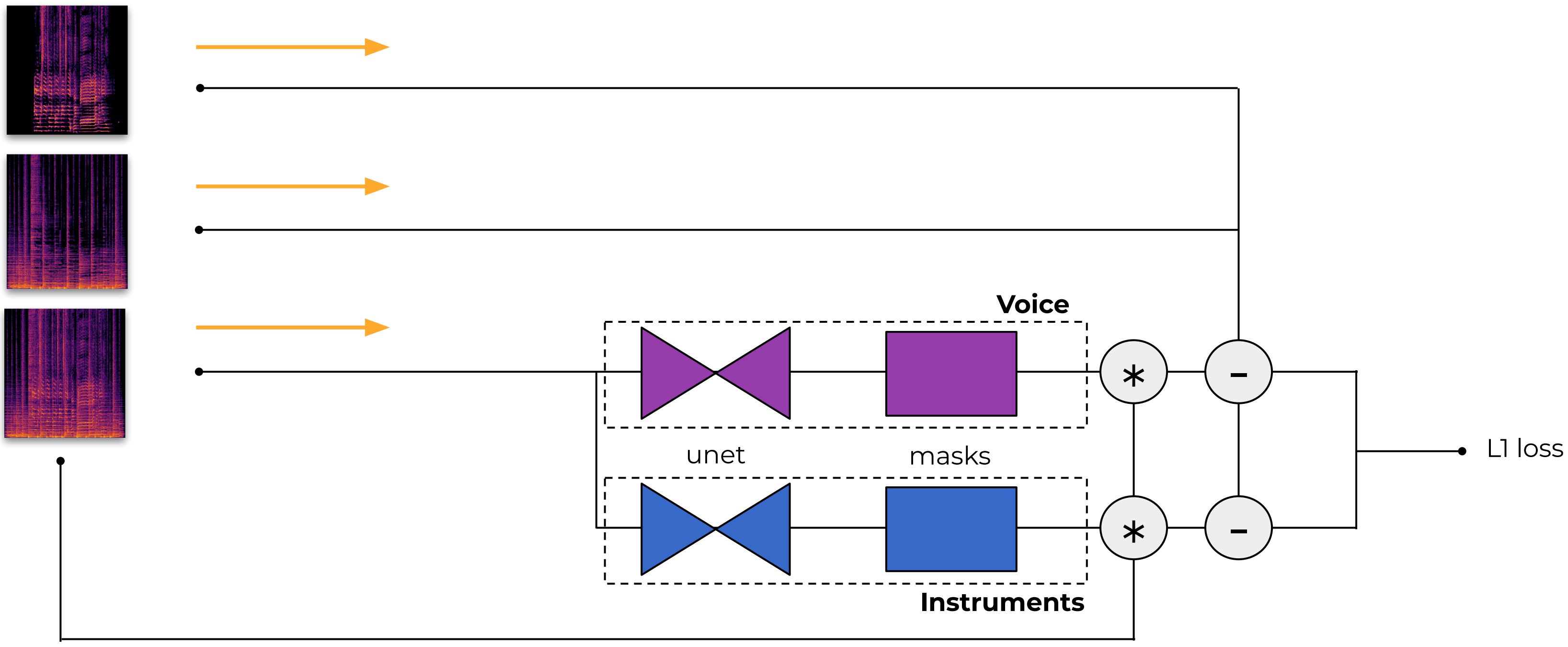


Overview

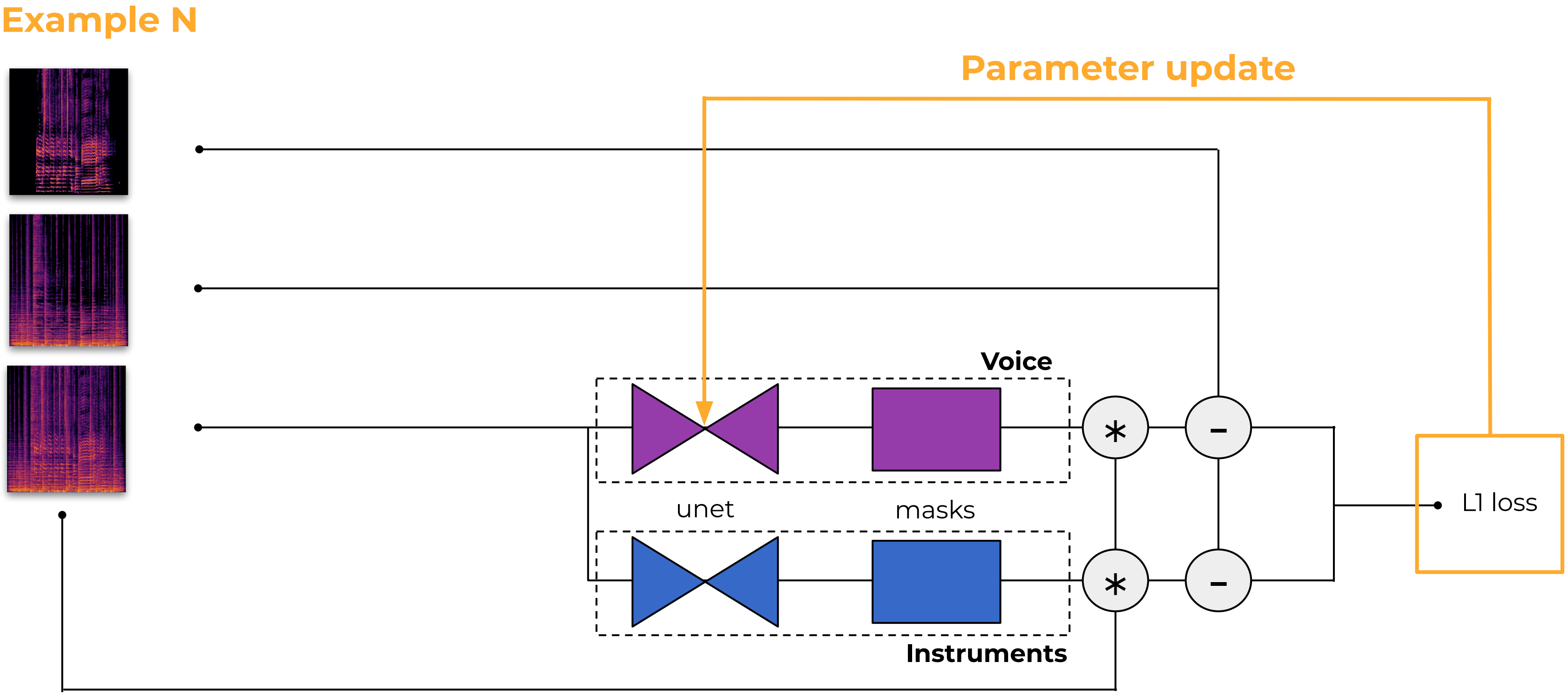

Example 2
Parameter update
Voice
unet
masks
Instruments
*
*
–
–
L1 loss

Overview

Example N



Overview


Example N
Parameter update
Voice
unet
masks
Instruments
*
*
-
-
L1 loss

Dataset

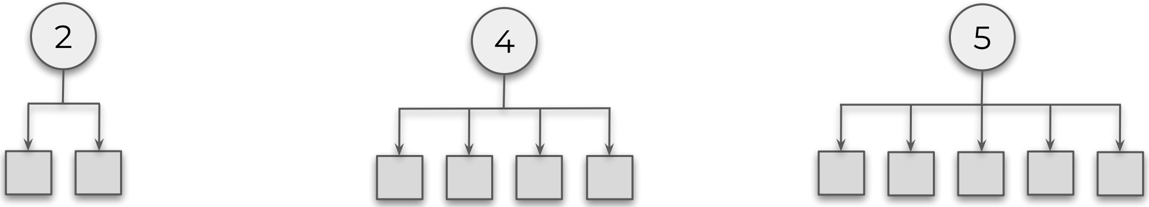
- In-house dataset of tracks
- ~24k tracks with stems
- ~80 hours of recording

… that we are not allowed to release!

BUT…

Training

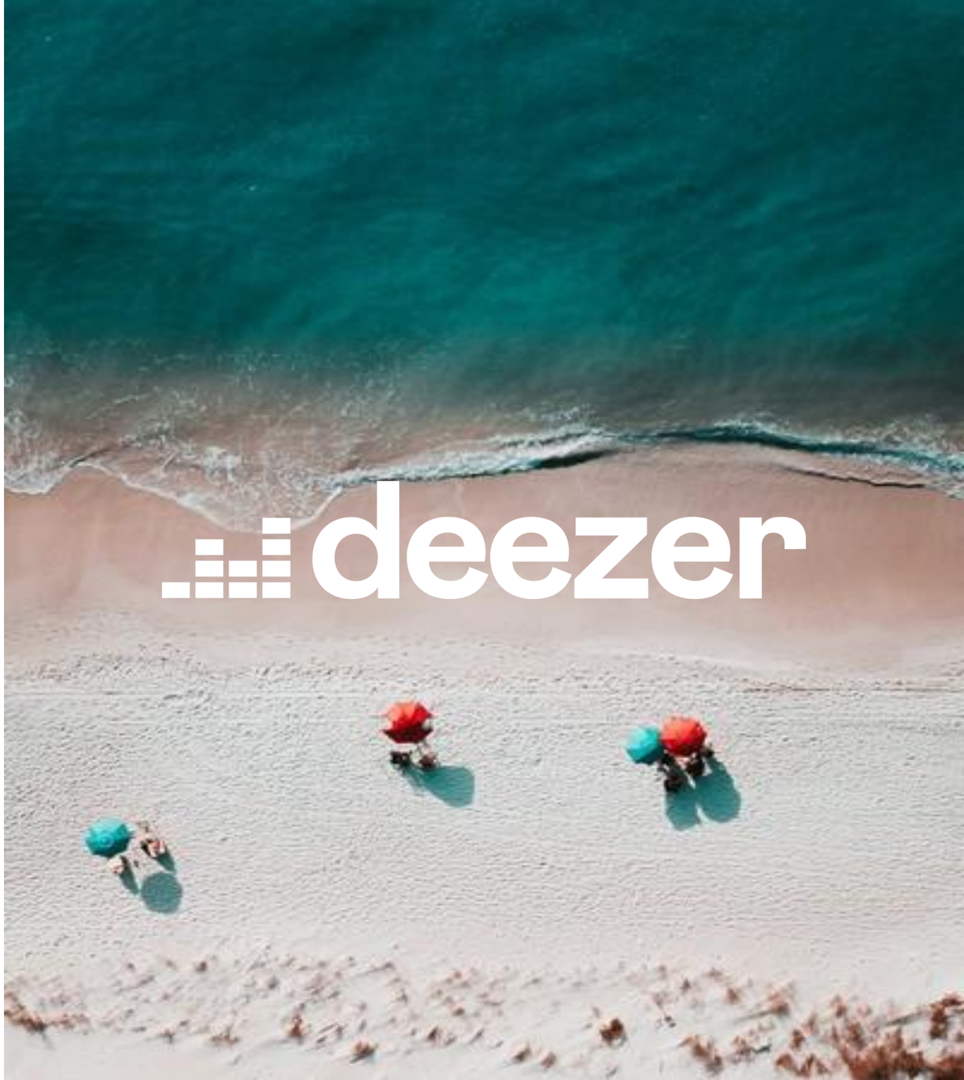
We can release learned **weights**



- One mask per channel per source.
- 1 branch predicts masks for 1 source, with 2 channels
- ~10M parameters per branch

Open-sourcing Spleeter

Packaging & distribution



Packaging constraints

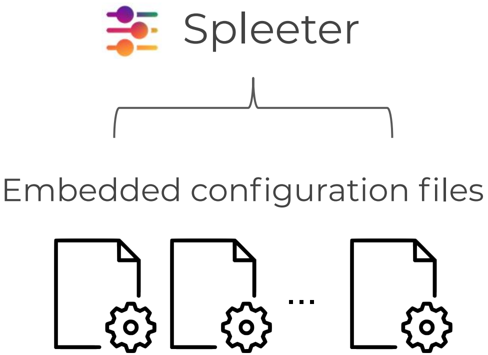
⚙ Predefined configurations

📥 On demand model downloading

>_ Oneliner command

Predefined configurations

- JSON formatted file
- Mostly model related parameters
- Provided as
    - File path
    - Configuration name

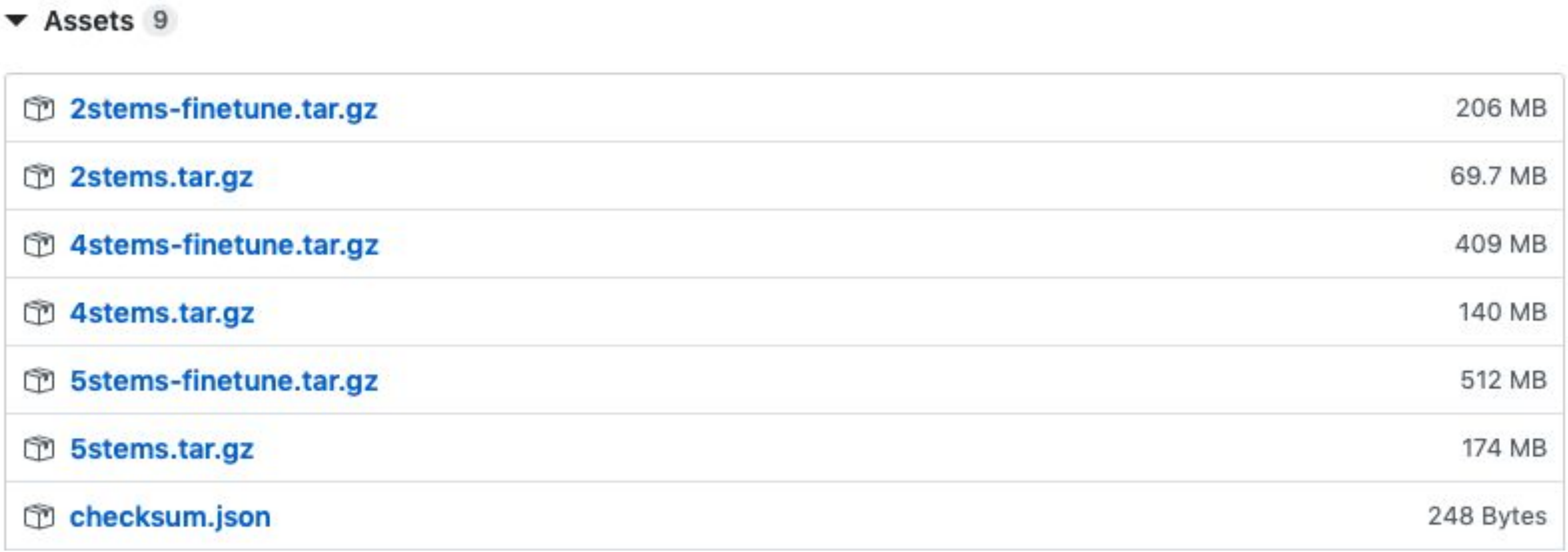


Spleeter

Embedded configuration files

Using GitHub releases as model hub

Spleeter → deezer/spleeter



▼ Assets 9

2stems-finetune.tar.gz	206 MB
2stems.tar.gz	69.7 MB
4stems-finetune.tar.gz	409 MB
4stems.tar.gz	140 MB
5stems-finetune.tar.gz	512 MB
5stems.tar.gz	174 MB
checksum.json	248 Bytes

Separate source from command line

Separate with default 2stems configuration

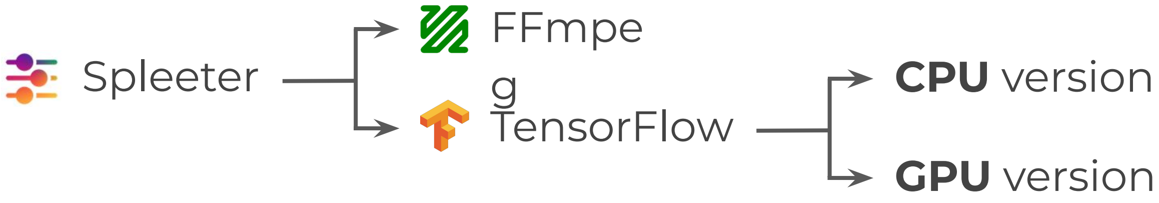
```
$ spleeter separate -i input_file.mp3 -o output_dir
```

Separate with specific embedded configuration

```
$ spleeter separate -i input_file.mp3 -o output_dir -p spleeter:4stems
```

Distribution constraints

Critical dependencies to manage :



Spleeter

FFmpe
g

TensorFlow

CPU version

GPU version

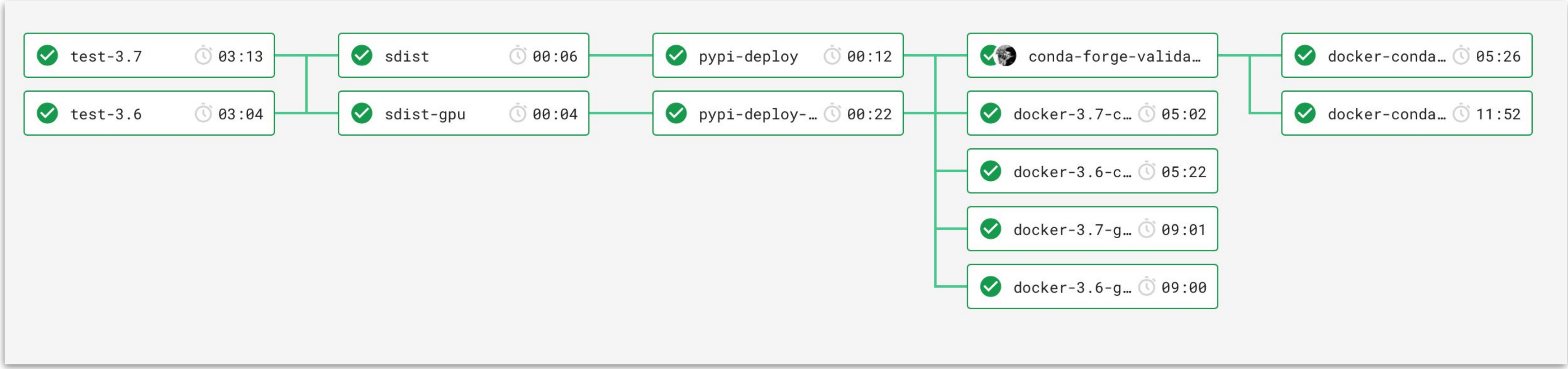
- Cross platform
- Cross hardware
- User friendly

Distribution channels

python™
Package
Index

CONDA-FORGE

Continuous integration and delivery



test-3.7 03:13

test-3.6 03:04

sdist 00:06

sdist-gpu 00:04

pypi-deploy 00:12

pypi-deploy-… 00:22

conda-forge-valida…

docker-3.7-c… 05:02

docker-3.6-c… 05:22

docker-3.7-g… 09:01

docker-3.6-g… 09:00

docker-conda… 05:26

docker-conda… 11:52

Legal considerations

- No Intellectual Property consensus on **weights**
- We decided to open-source **model**

Bibliography and references

Industrial integrations :

- **AconDigital** plugins
- Various public web applications

~30 projects referenced as using Spleeter on **GitHub**

Research publications :

- https://ieeexplore.ieee.org/document/8683555
- http://archives.ismir.net/ismir2019/latebreaking/000036.pdf

Demo

Spleeter Live

Thank you

research@deezer.com