Music Source Separation

FREQUENCY DOMAIN: D3NET, TAKAHASHI+, CVPR2021

WAVEFORM DOMAIN: DEMUCS, DEFOSSEZ+, ARXIV2021

Nabarun Goswami
Department of Advanced Interdisciplinary Studies, The University of Tokyo, D1

Music Production or Mixing

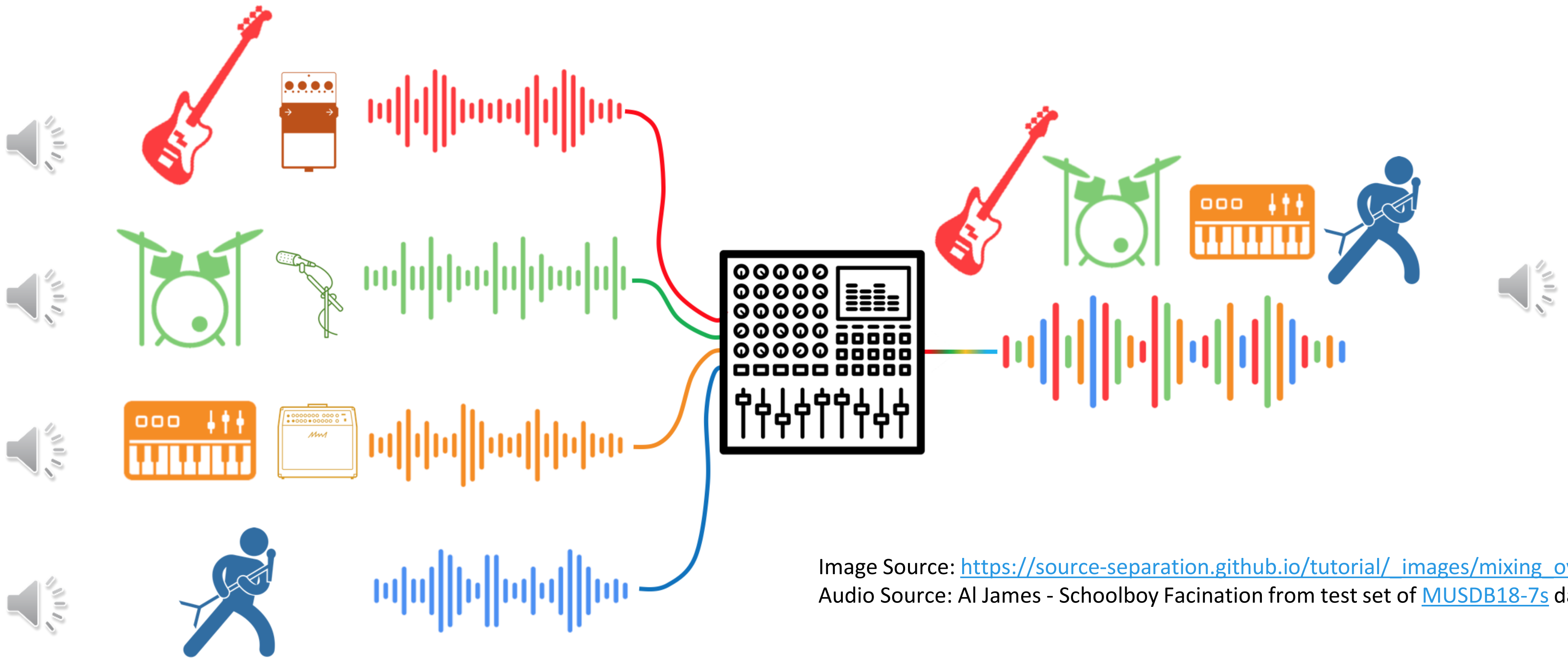


Image Source: https://source-separation.github.io/tutorial/_images/mixing_overview.png
Audio Source: Al James - Schoolboy Facination from test set of MUSDB18-7s dataset

Music Source Separation or De-mixing

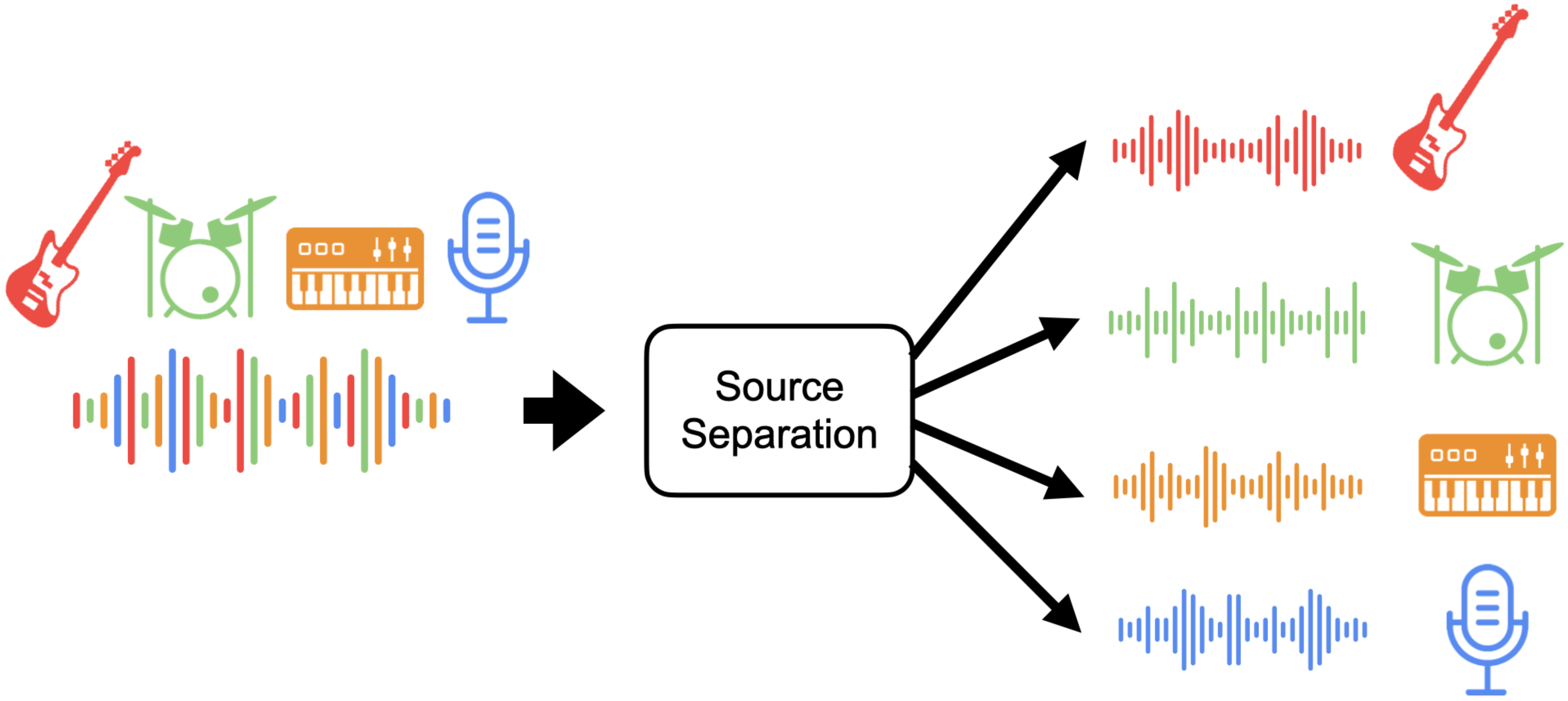
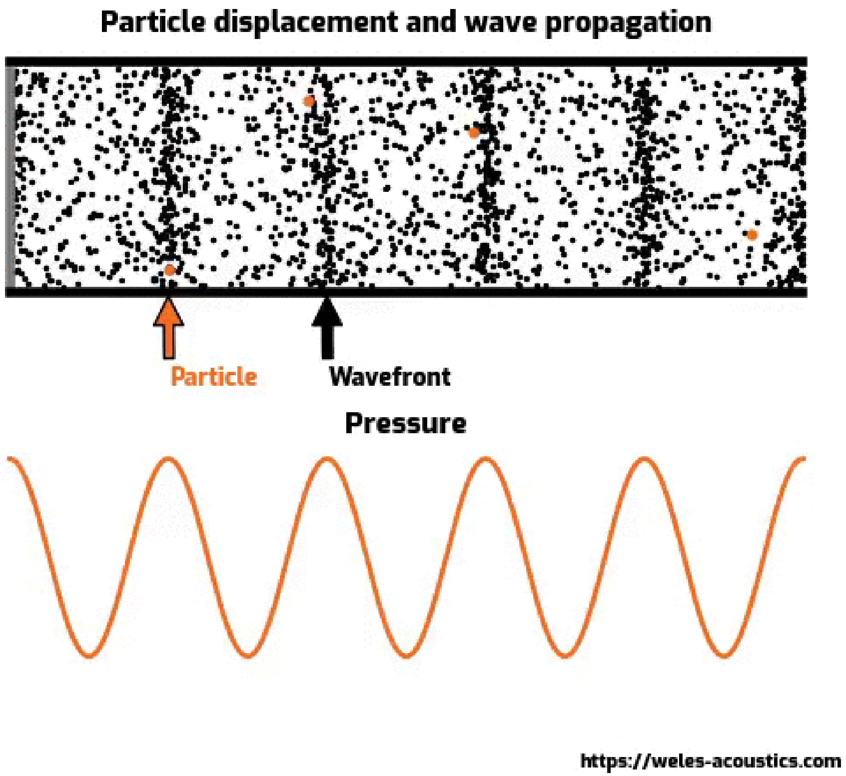
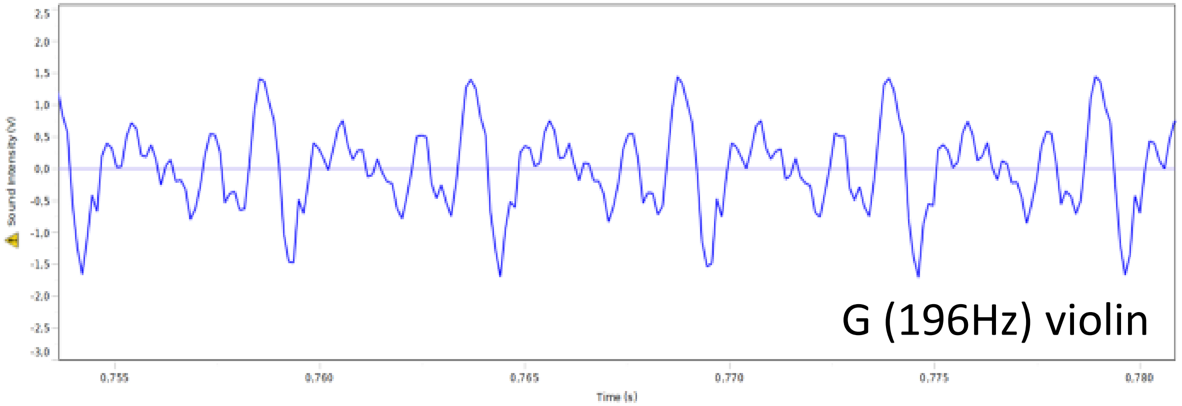


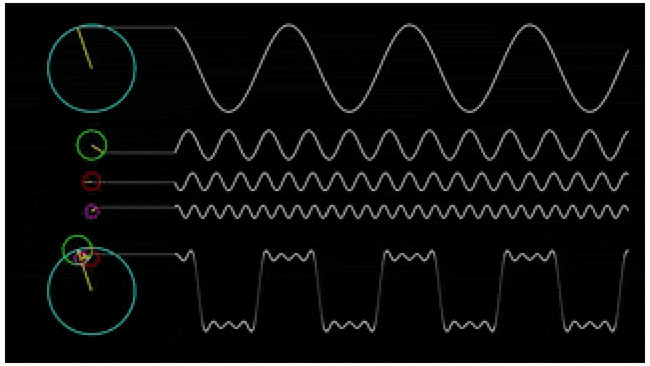
Image Source: https://source-separation.github.io/tutorial/_images/source_separation_io.png

Sounds are just a bunch of Sine Waves





G (196Hz) violin



Images Source: https://medium.com/age-of-awareness/the-science-behind-the-sound-10bdc94ad70

Time-Domain Representation of Audio

Digital audio is just a sequence of amplitude values sampled at a certain rate.

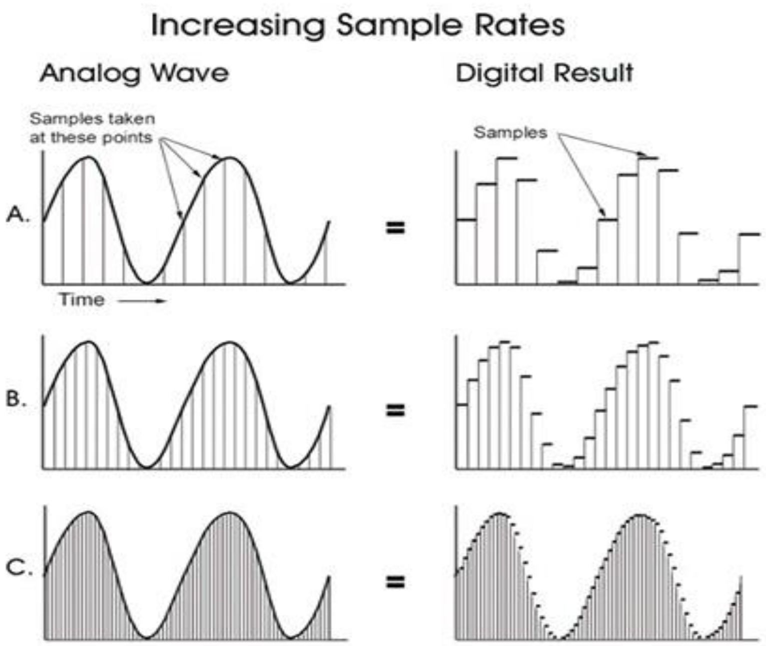


Image source: https://mp4gain.com/mp4gain/wp-content/uploads/2019/09/sample-rate-0.jpg

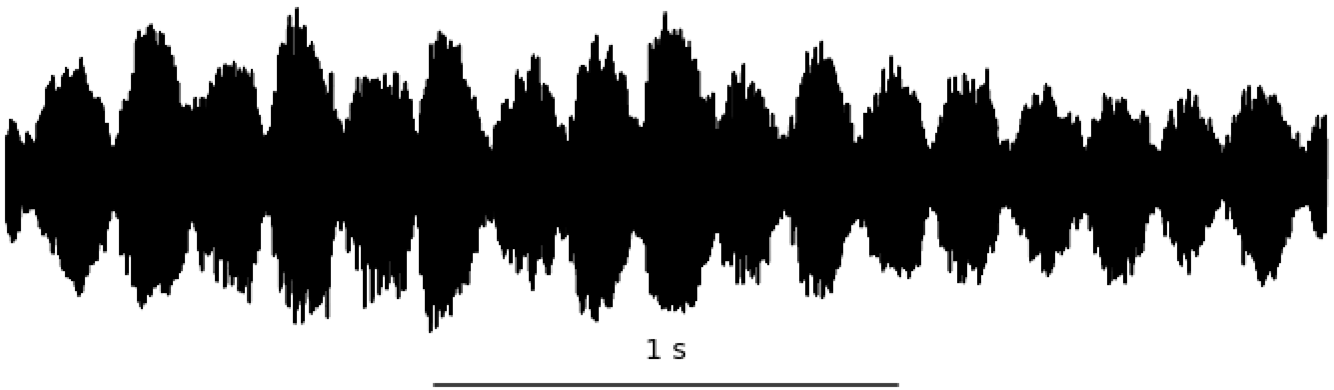


Image Source: https://jvbalen.github.io/notes/waveform.html

Typical Sampling Rates:

- 8kHz : walkie talkie/telephone
- 16kHz : VoIP
- 44.1kHz : CD Music
- 96kHz : DVD/BluRay

Time-Frequency Representation of Audio

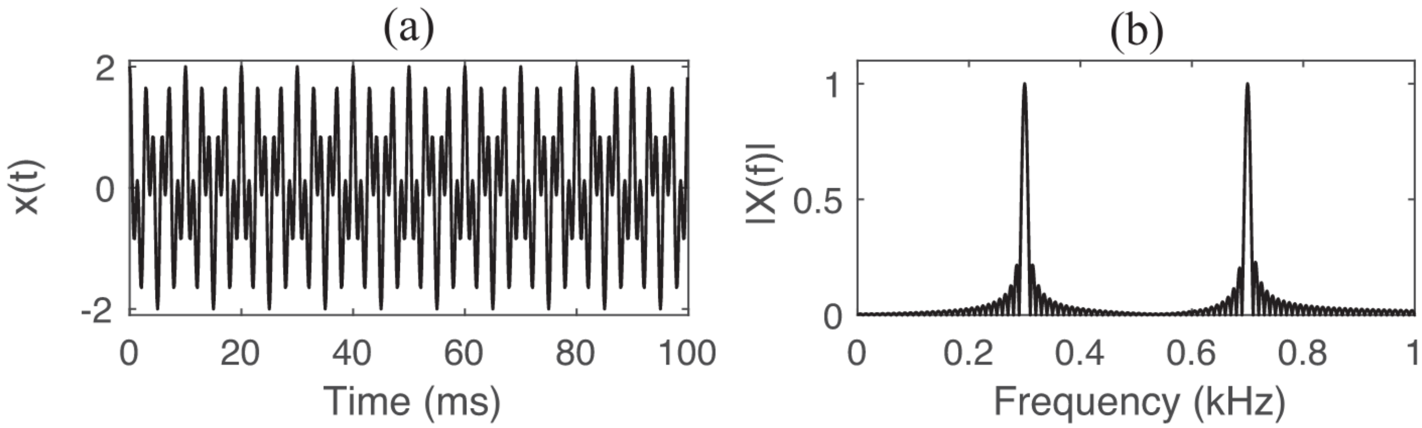
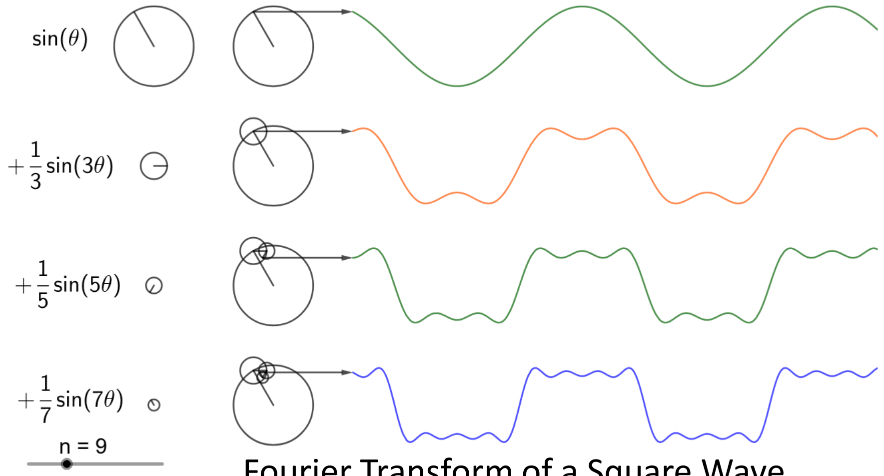


Image Source: https://source-separation.github.io/tutorial/_images/right_representation.png



Fourier Transform of a Square Wave

Image Source: https://www.geogebra.org/resource/hwtd5jkk/dvdK7uhml3NL7tKh/material-hwtd5jkk.png

Short-time Fourier Transform

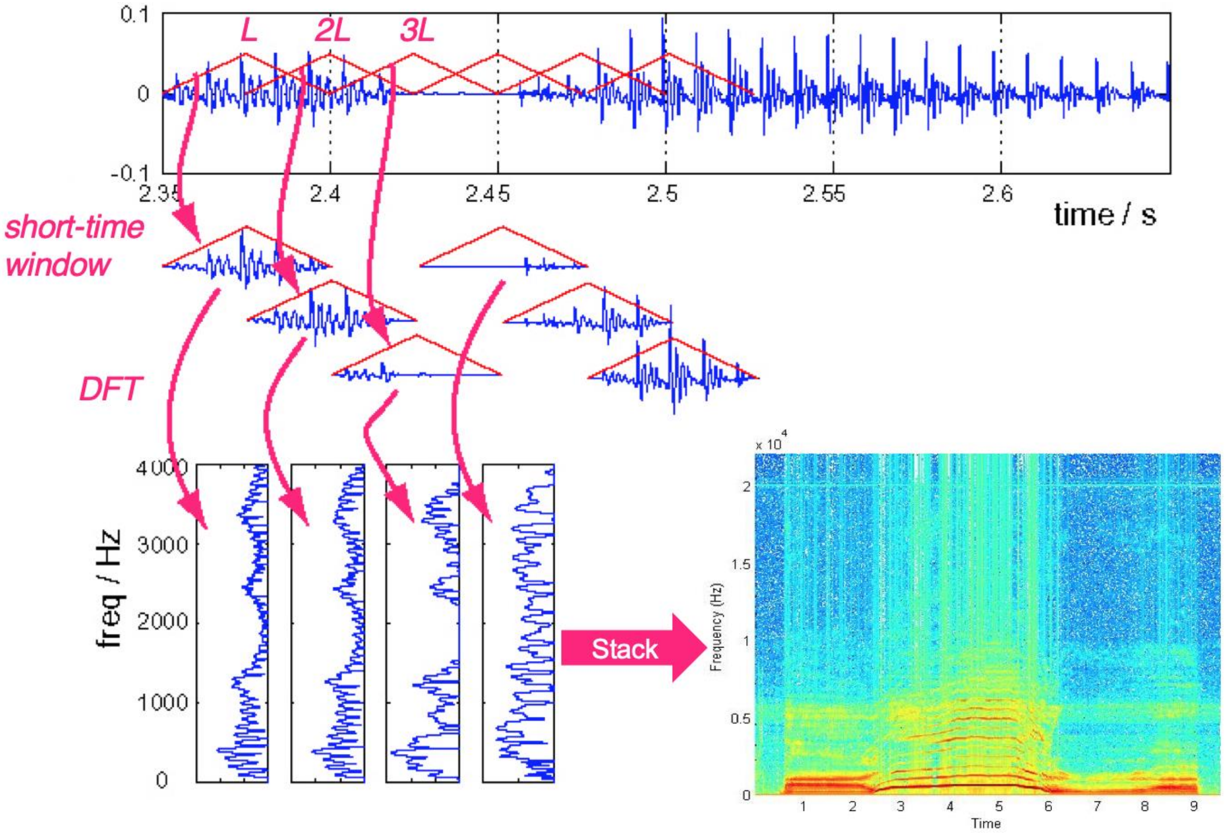
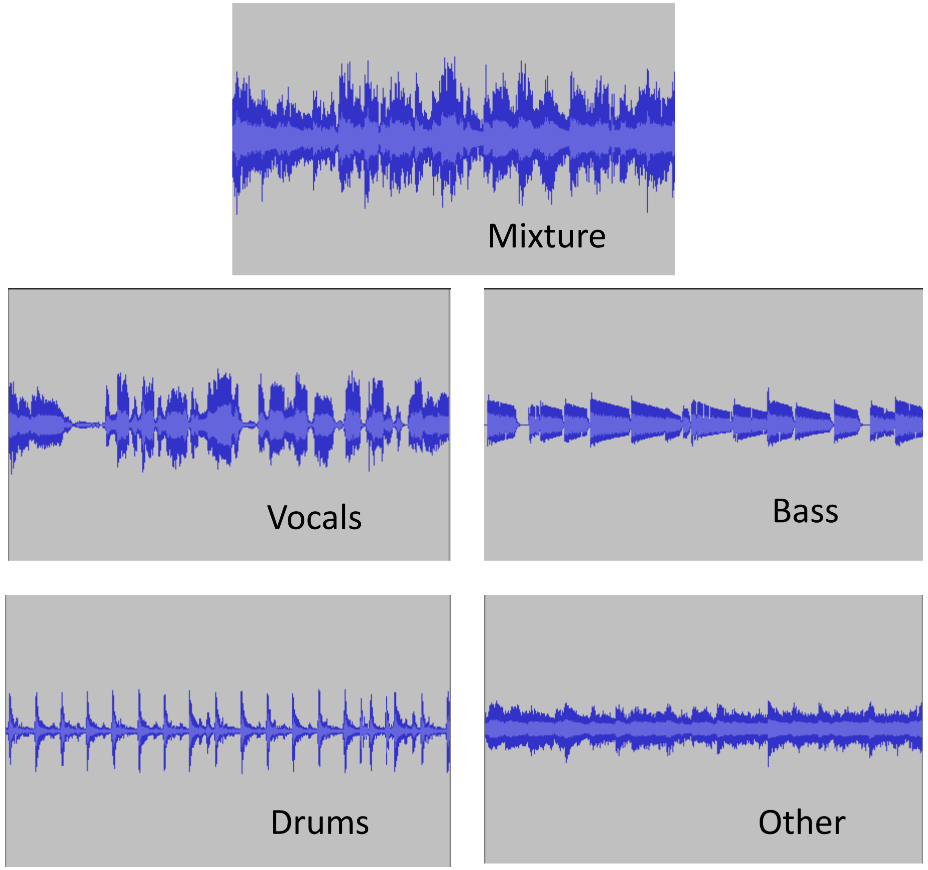


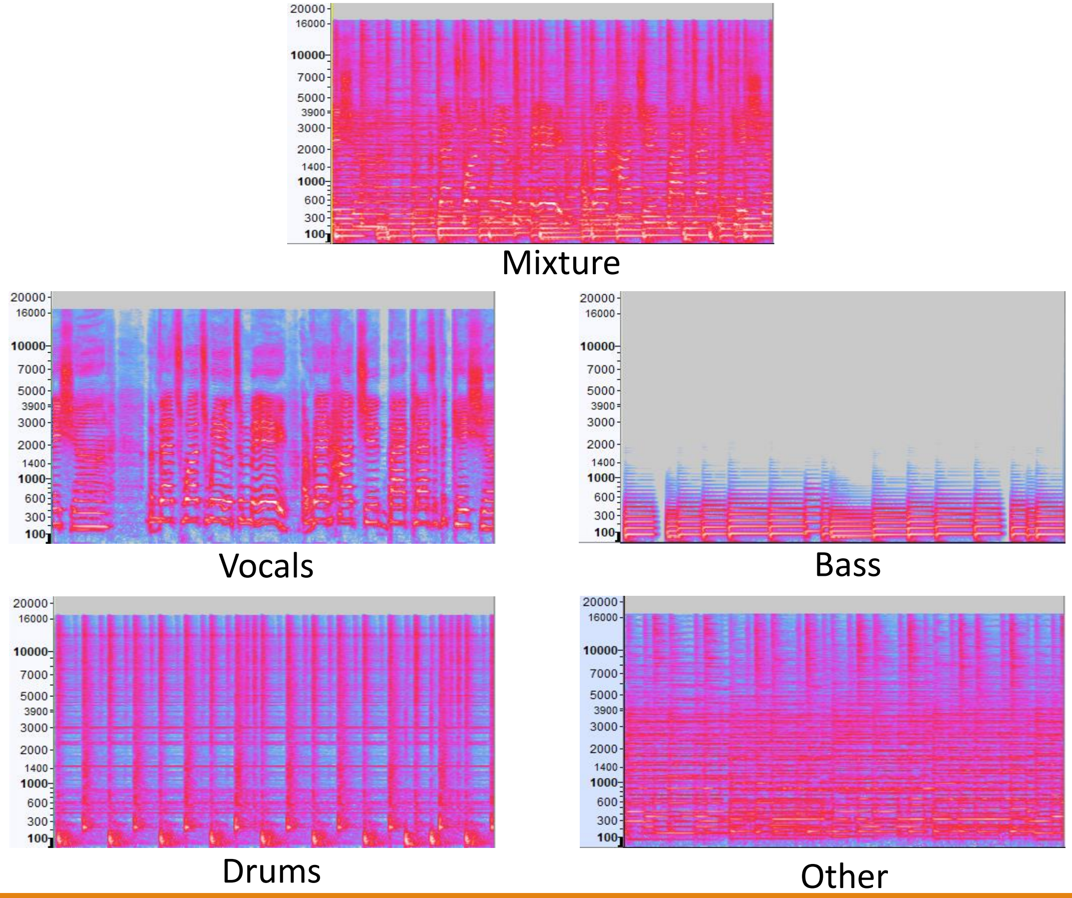
Image source: https://source-separation.github.io/tutorial/_images/stft_process.png

Two ways of Representing Sounds
Two ways of Tackling Source Separation

Time Domain or Waveform Domain

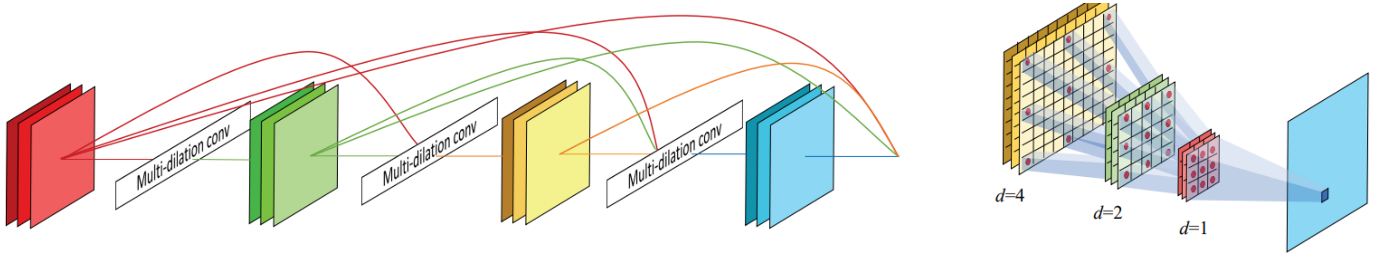


Frequency Domain



D3Net

Densely connected multidilated convolutional networks for dense prediction tasks (CVPR 2021, Takahashi+)



(a) Dilated dense block

(b) Multi-dilated convolution

Figure 1. Illustration of D2 block. (a) The connectivity pattern is the same as that in DenseNet except that the D2 block involves the multidilated convolution. (b) Illustration of the multidilated convolution at the third layer. The production of a single feature map involves multiple dilation factors depending on the input channel. For clarity, we omit the normalization and nonlinearity from the illustration.

Table 4. SDRs for MUSDB18 dataset. '*' denotes the method operating in the time domain.

Method	SDR in dB					
	Vocals	Drums	Bass	Other	Acco.	Avg.
TAK1 (MMDenseLSTM) [41]	6.60	6.43	5.16	4.15	12.83	5.59
UHL2 (BLSTM ensemble) [45]	5.93	5.92	5.03	4.19	12.23	5.27
GRU dilation 1 [22]	6.85	5.86	4.86	**4.65**	13.40	5.56
UMX [37]	6.32	5.73	5.23	4.02	-	5.33
demucs* [7]	6.29	6.08	5.83	4.12	-	5.58
Meta-TasNet* [31]	6.40	5.91	5.58	4.19	-	5.52
Nachmani *et al.* * [25]	6.92	6.15	**5.88**	4.32	-	5.82
D3Net without dilation	6.86	6.37	4.97	4.21	13.19	5.60
D3Net standard dilation	7.12	6.61	5.19	4.53	13.39	5.86
D3Net (proposed)	**7.24**	**7.01**	5.25	4.53	**13.52**	**6.01**

(a) Image (b) Ground truth (c) D3Net

Figure 5. Qualitative examples of Cityscapes results on *val* set.

Table 3. Results on Cityscapes *test* set. Baseline results are from original papers. All models are trained on the *train* set without using coarse data.

	Backbone	mIoU
PSPNet [55]	D-ResNet-101	78.4
PSANet [56]	D-ResNet-101	78.6
PAN [20]	D-ResNet-101	78.6
AAF [17]	D-ResNet-101	79.1
HRNetV2 [47]	HRNetV2-W48	80.4
D3Net (FCN)	D3Net-L	**80.8**

Image Source: https://openaccess.thecvf.com/content/CVPR2021/papers/Takahashi_Densely_Connected_Multi-Dilated_Convolutional_Networks_for_Dense_Prediction_Tasks_CVPR_2021_paper.pdf

D3Net

Music Source Separation in Frequency Domain

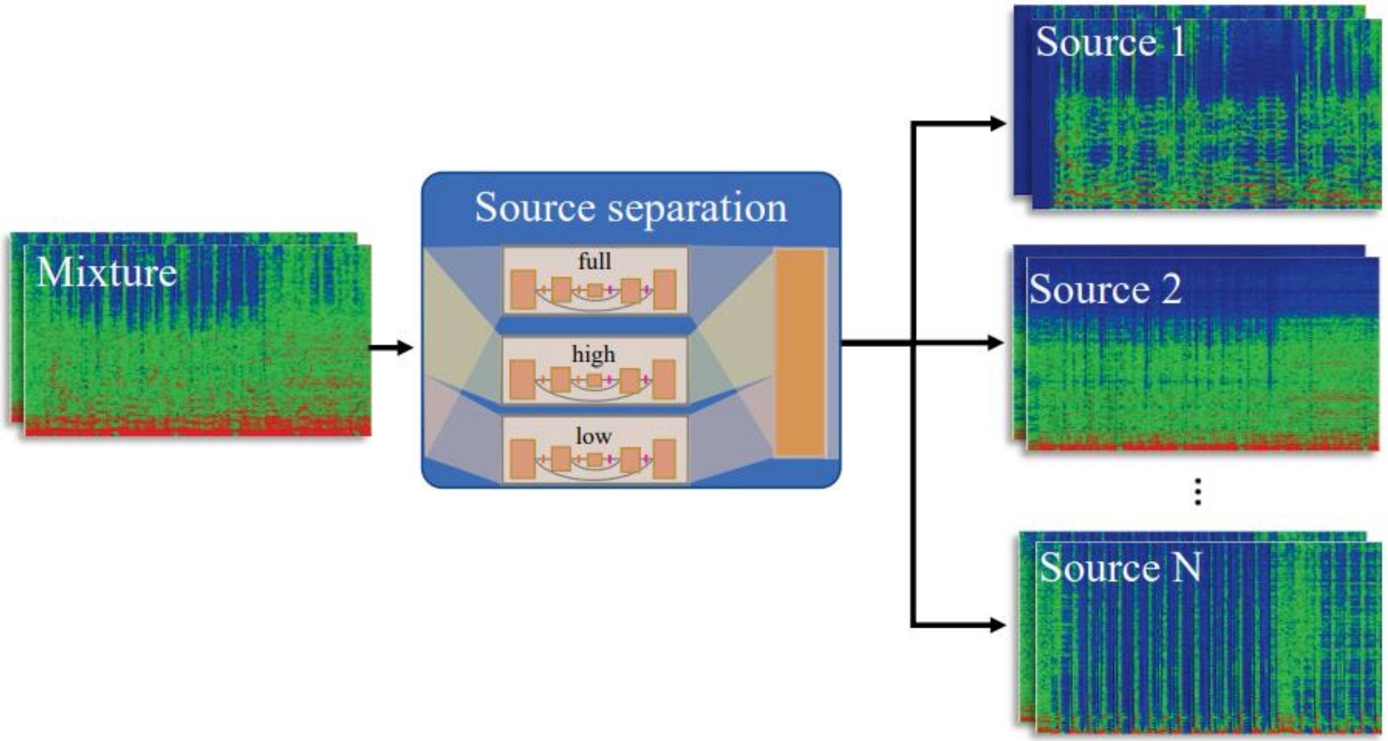


Figure 6. Illustration of audio source separation in STFT domain.

Multi-dilated Convolutions has anti aliasing property

Aliasing

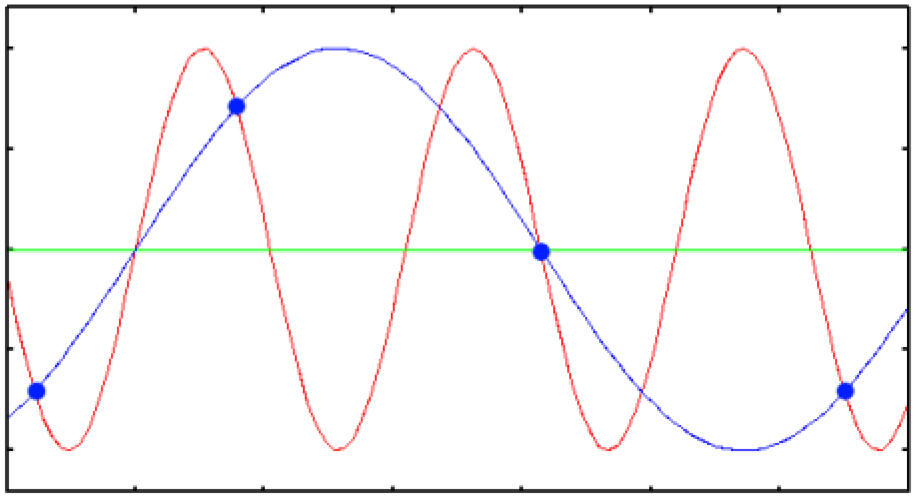
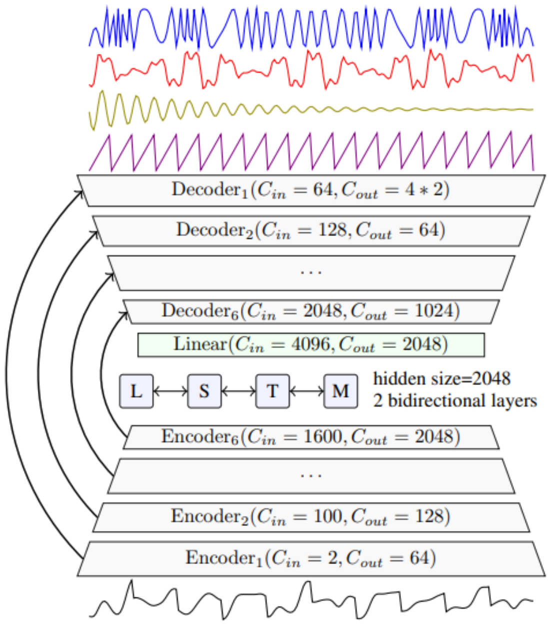


Image Source: https://support.ircam.fr/docs/AudioSculpt/3.0/res/aliasing.png

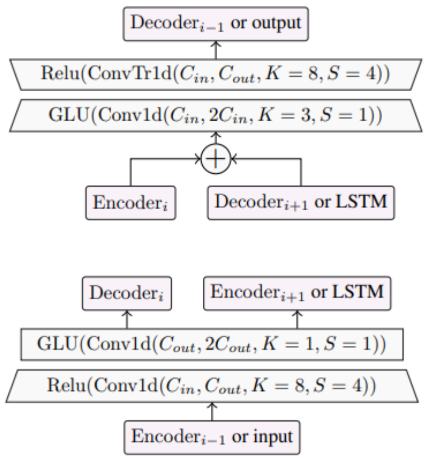
Image Source: https://openaccess.thecvf.com/content/CVPR2021/papers/Takahashi_Densely_Connected_Multi-Dilated_Convolutional_Networks_for_Dense_Prediction_Tasks_CVPR_2021_paper.pdf

Demucs

Music Source Separation in the Waveform Domain (Defossez+, Preprint, 2021)



(a) Demucs architecture with the mixture waveform as input and the four sources estimates as output. Arrows represents U-Net connections.



(b) Detailed view of the layers Decoder$_i$ on the top and Encoder$_i$ on the bottom. Arrows represent connections to other parts of the model. For convolutions, C_in (resp C_out) is the number of input channels (resp output), K the kernel size and S the stride.

Figure 2: Demucs complete architecture on the left, with detailed representation of the encoder and decoder layers on the right.

			Test SDR in dB				
Architecture	**Wav?**	**Extra?**	All	Drums	Bass	Other	Vocals
IRM oracle	✗	N/A	8.22	8.45	7.12	7.85	9.43
Wave-U-Net	✓	✗	3.23	4.22	3.21	2.25	3.25
Open-Unmix	✗	✗	5.33	5.73	5.23	4.02	6.32
Meta-Tasnet	✓	✗	5.52	5.91	5.58	4.19	6.40
Conv-Tasnet†	✓	✗	5.73 ±.10	6.02 ±.08	6.20 ±.15	4.27 ±.03	6.43 ±.16
DPRNN	✓	✗	5.82	6.15	5.88	4.32	6.92
D3Net	✗	✗	6.01	**7.01**	5.25	**4.53**	**7.24**
Demucs†	✓	✗	6.28 ±.03	6.86 ±.05	**7.01** ±.19	4.42 ±.06	6.84 ±.10
Spleeter	✗	~ 25k*	5.91	6.71	5.51	4.55	6.86
TasNet	✓	~ 2.5k	6.01	7.01	5.25	4.53	7.24
MMDenseLSTM	✗	804	6.04	6.81	5.40	4.80	7.16
Conv-Tasnet††	✓	150	6.32 ±.04	7.11 ±.13	7.00 ±.05	4.44±.03	6.74 ±.06
D3Net	✗	1.5k	6.68	7.36	6.20	**5.37**	**7.80**
Demucs†	✓	150	**6.79** ±.02	**7.58** ±.02	**7.60** ±.13	4.69 ±.04	7.29 ±.06

Image Source: https://hal.archives-ouvertes.fr/hal-02379796/document

iSeparate

Implementation and Reproduction of Music Source Separation Methods

https://github.com/media-comp/2022-iSeparate

iSeparate

This repository consists of an attempt to reimplement, reproduce and unify the various deep learning based methods for Music Source Separation.

This project was started as part of the requirement for the course Media Computing in Practice at the University of Tokyo, under the guidance of Yusuke Matsui sensei.

This is a work in progress, current results are decent but not as good as reported in the papers, please use with a pinch of salt. Will continue to try and improve the quality of separation.

Currently implemented methods:

Model	Paper	Official code
D3Net	Densely connected multidilated convolutional networks for dense prediction tasks (CVPR 2021, Takahashi et al., Sony)	link
Demucs v2	Music Source Separation in the Waveform Domain (Arxiv 2021, Defossez et al., Facebook, INRIA)	link

Separate using pre-trained model

Create your own Karaoke tracks!

Currently the D3Net vocals model has been uploaded to Huggingface and you can run vocals-accompaniment separation using that model with the `separate.py` script. Invoke the separation as follows:

```
python separate.py \
                -c configs/d3net/eval.yaml \
                -i path/to/song.wav
```

Currently only `.wav` … use the following command to convert `.mp3` file to `.wav` file within the co… …ove:

```
…peg -i song.mp3 song.wav
```

Check the README to create your own Karaoke Tracks!

Implementation Details

- Framework: Pytorch
- Training Dataset: MUSDB18
  - “The *musdb18* is a dataset of 150 full lengths music tracks (~10h duration) of different genres along with their isolated *drums*, *bass*, *vocals* and *others* stems.” (https://sigsep.github.io/datasets/musdb.html)
- Infrastructure:
  - 4x Nvidia A100 GPU with 80GB VRAM (not all 80GB was used)
  - Batch size: 32 per GPU
  - Automatic Mixed Precision
- Training Time:
  - D3Net (vocals): ~0.5 days
  - Demucs (all sources): ~4 days
- Model Size on disk:
  - D3Net (vocals): ~13 MB
  - Demucs (all sources): ~1 GB

Results from Current Implementation

Audio files are too big to embed in the PowerPoint presentation.

Demo in external Audio software.

Pull Requests

- Train D3Net for other sources
- Verify Demucs implementation (results are not as good as reported in the paper or the official code)
- Create a web-app (maybe a huggingface space)
- Create a VST plugin for integration in Digital Audio Workstations (https://audioassemble.com/how-to-make-a-vst/)
- Bugs and Fixes
- Implement other methods
- Anything else…

Thank you