Speaker Embedding Network & Applications

박승원 (Deepest, MINDsLab)

Deep Con 2nd

2019년 10월 6일

About me

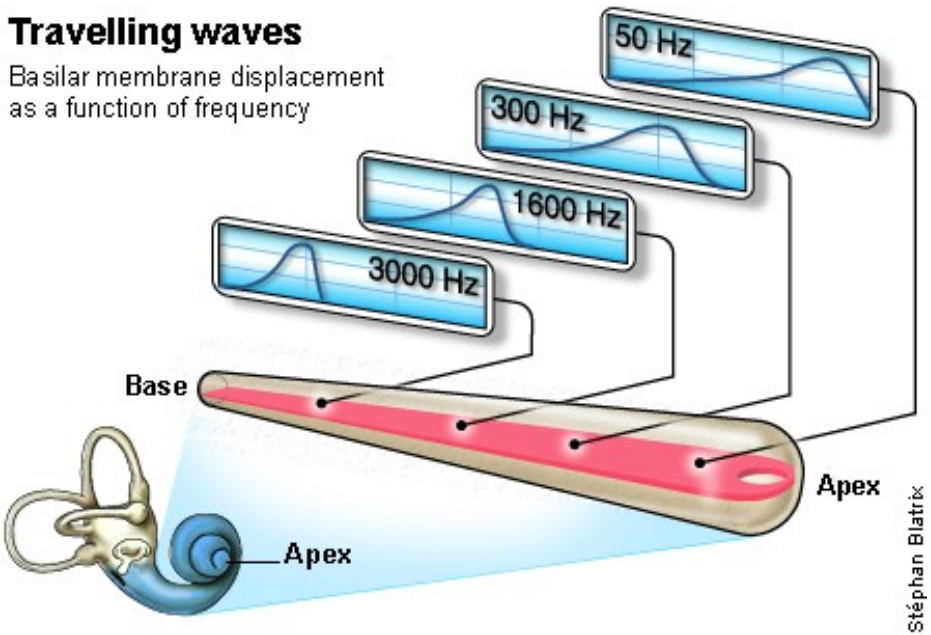
- 학부생 (물리 17)
  - 컴퓨터공학 복수전공 (2018.09 –)
- Deepest (2019.01 –)
- 마인즈랩 학사병특 (2019.02 –)
- `http://swpark.me`

Contents

Introduction

How should we feed audio to neural network?

- Raw audio $\in [-1, 1]^T$
- Mel-spectrogram, MFCC



Introduction – STFT & Mel-spectrogram

- Hamming window: $w[n] = 0.54 - 0.46\cos(2\pi n/L)$
- window 25 ms, stride(hop) 10 ms

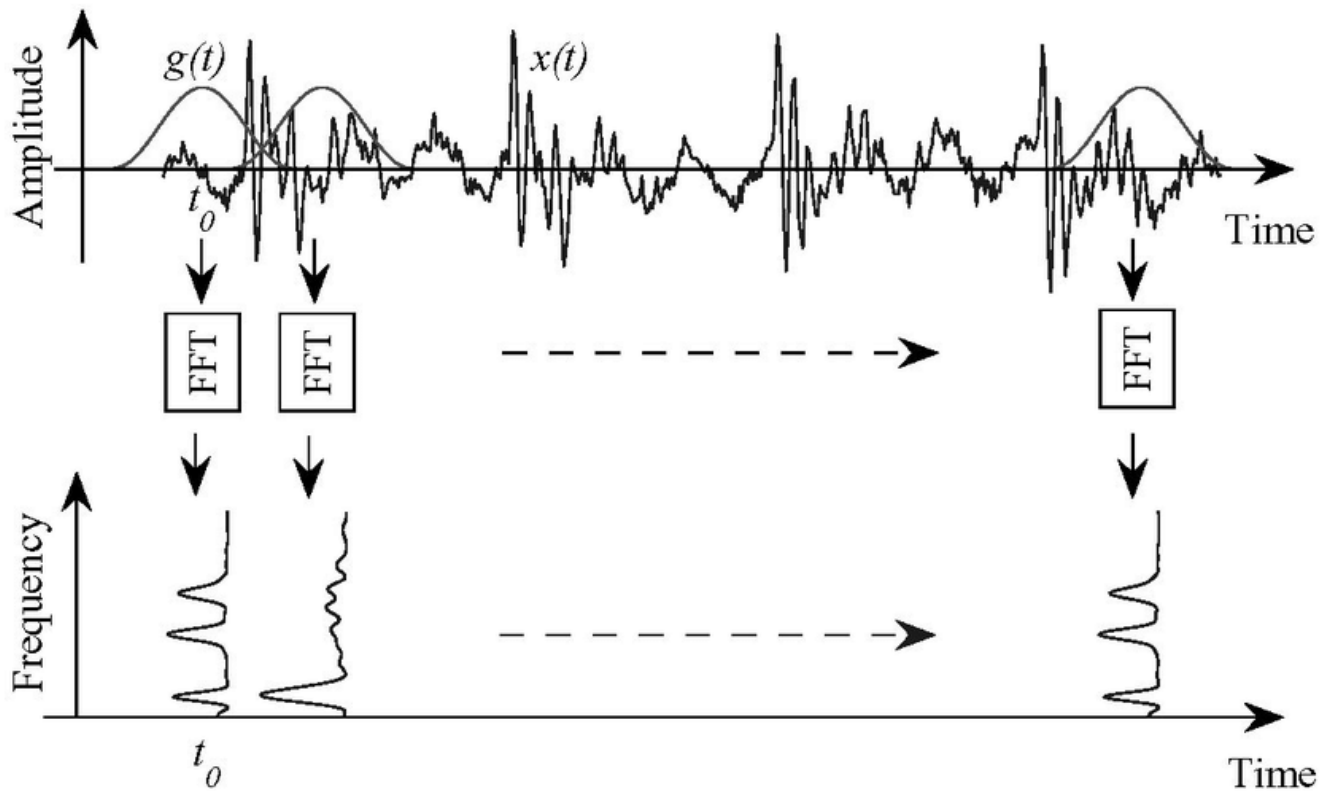
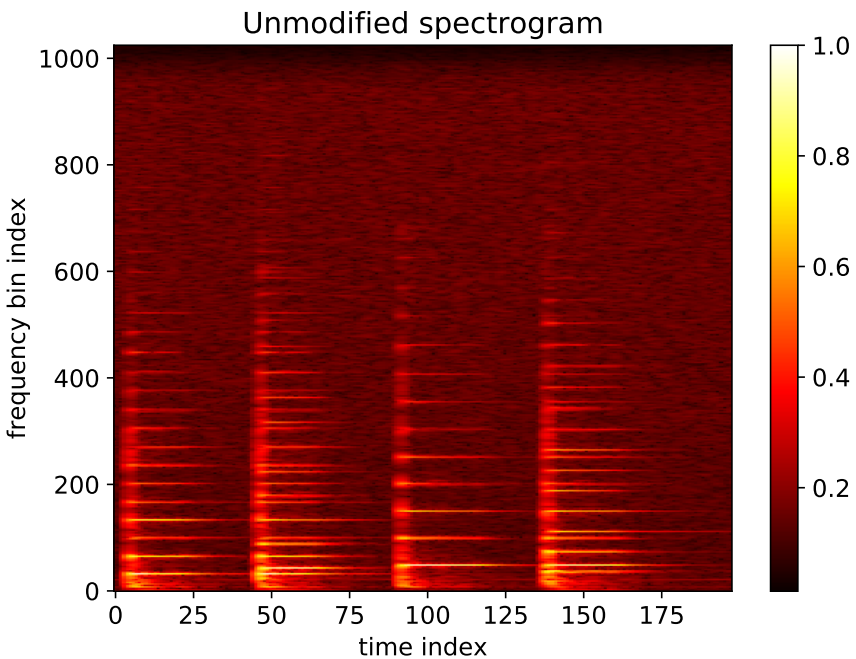
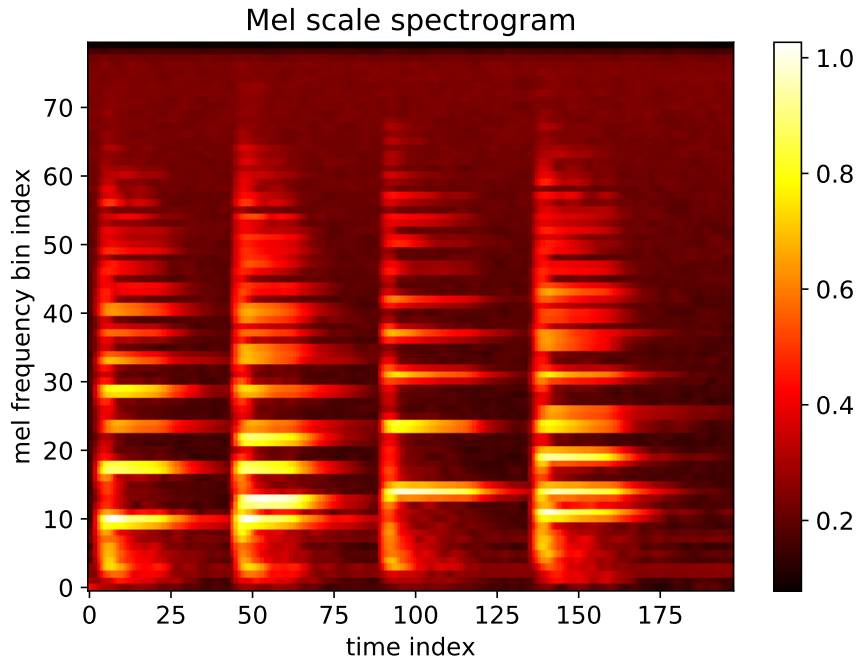


Image by Robert X. Gao, at `https://bit.ly/2Ikbiga`

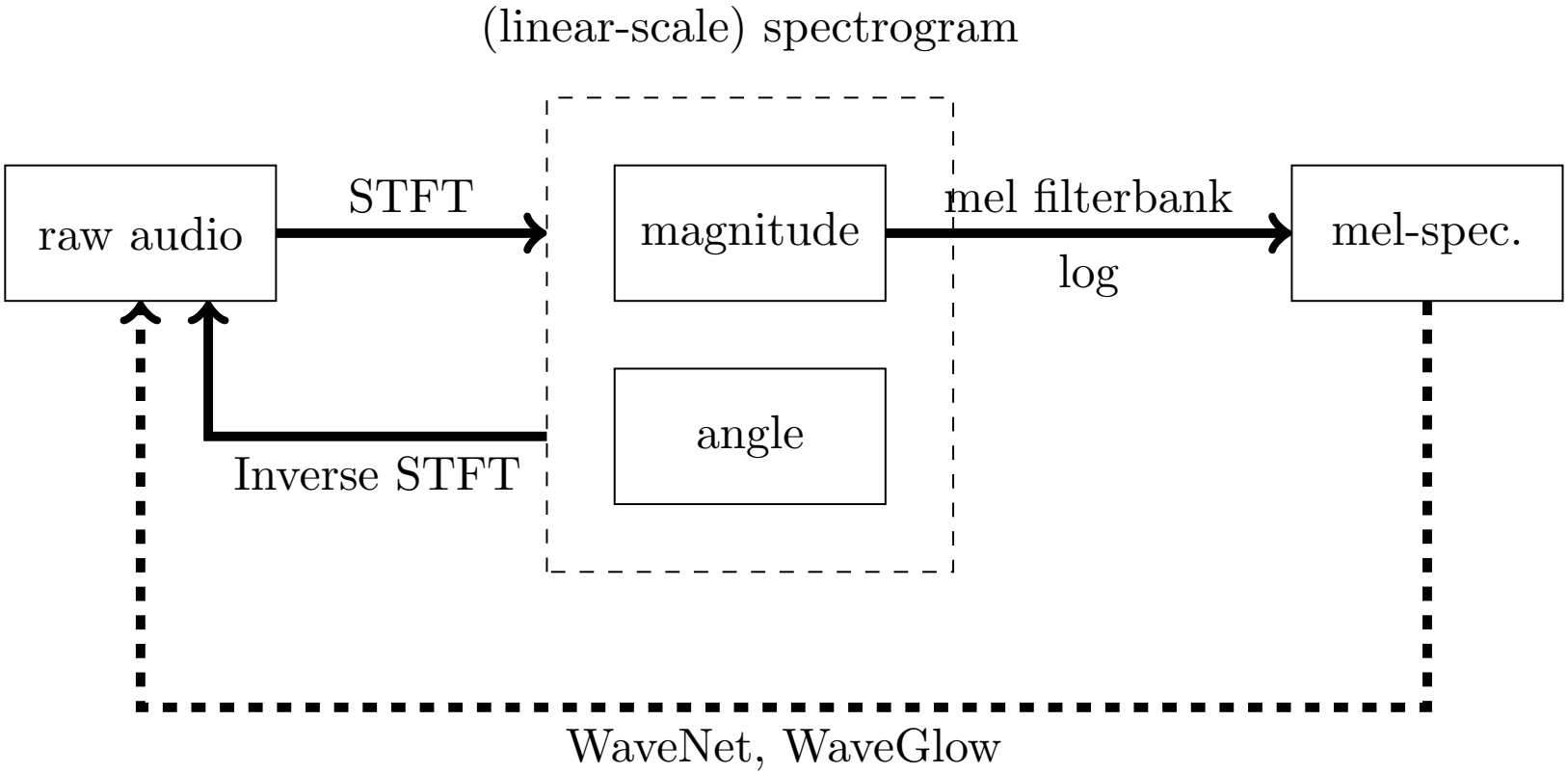
Introduction – STFT & Mel-spectrogram





Images generated with `github.com/bkvogel/griffin_lim`

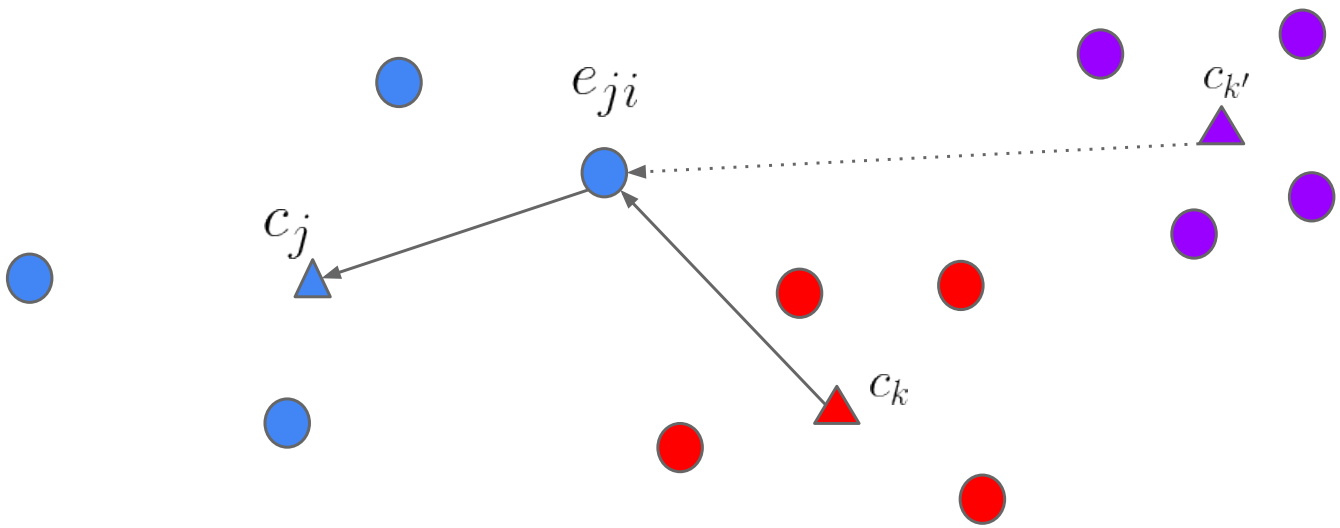
Introduction – STFT & Mel-spectrogram



Speaker Verification: d-vector

Overview

- Utterance $\xrightarrow{STFT}$ mel-spec. $\xrightarrow{LSTM+proj.}$ embedding $\in \mathbb{R}^{256}$
- Text independent, Zero-shot



from 'Generalized End-to-End Loss for Speaker Verification' by L. Wan *et al.*

Speaker Verification: d-vector

Loss function

$$L(\mathbf{e}_{ji}) = -\mathbf{S}_{ji,j} + \log \sum_{k=1}^{N} \exp(\mathbf{S}_{ji,k}). \tag{6}$$

where

$$\mathbf{c}_j^{(-i)} = \frac{1}{M-1} \sum_{\substack{m=1 \\ m \neq i}}^{M} \mathbf{e}_{jm}, \tag{8}$$

$$\mathbf{S}_{ji,k} = \begin{cases} w \cdot \cos(\mathbf{e}_{ji}, \mathbf{c}_j^{(-i)}) + b & \text{if} \quad k = j; \\ w \cdot \cos(\mathbf{e}_{ji}, \mathbf{c}_k) + b & \text{otherwise.} \end{cases} \tag{9}$$

Speaker Verification: d-vector

- Create random-sized batch: 70 – 90 frames
- Inference: window 80 / hop 40, average pooling
  - 80 frame: 25 ms + 79 * 10 ms = 815 ms

```
1  def forward(self, x):    # (B, T, num_mels)
2      x, _ = self.lstm(x) # (B, T, lstm_hidden)
3      x = x[:, -1, :]      # (B, lstm_hidden)
4      x = self.proj(x)     # (B, emb_dim)
5      x = x / torch.norm(x, p=2, dim=1, keepdim=True)
6      return x
```

Speaker Verification: Data

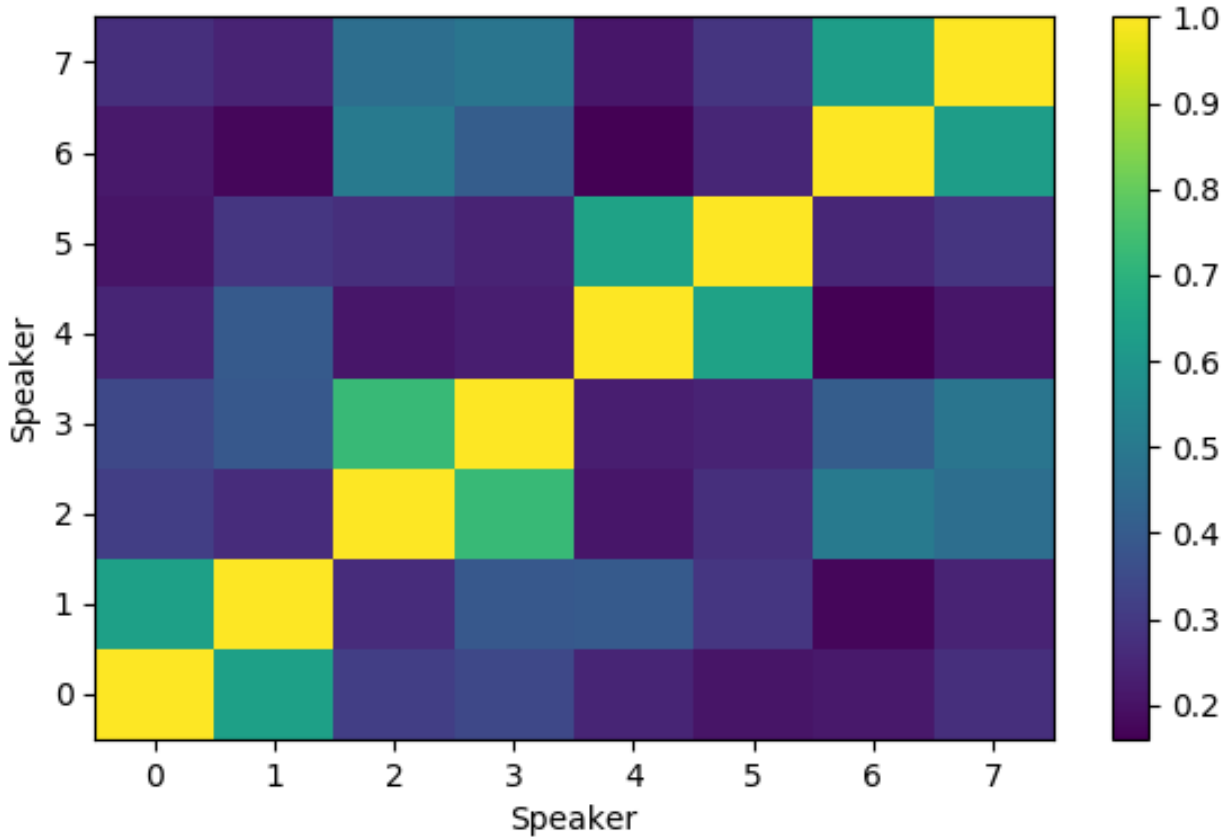
Training data should be:

- large enough,
- contain speakers with various tones,
- utterances recorded from 'the wild'

to prevent overfitting & discard any other info. than speaker's identity from the embedding.

Speaker Verification

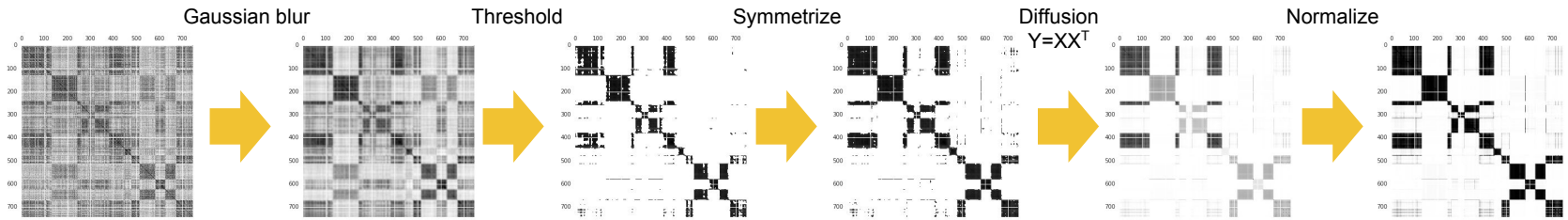
- Training data: VoxCeleb 2 (Multilingual, 5,994 spkr, 10^6 utt.)
- Demo: (이명박 / 문재인 / 박근혜 / 손석희) x 2
  - None of them were seen during training.



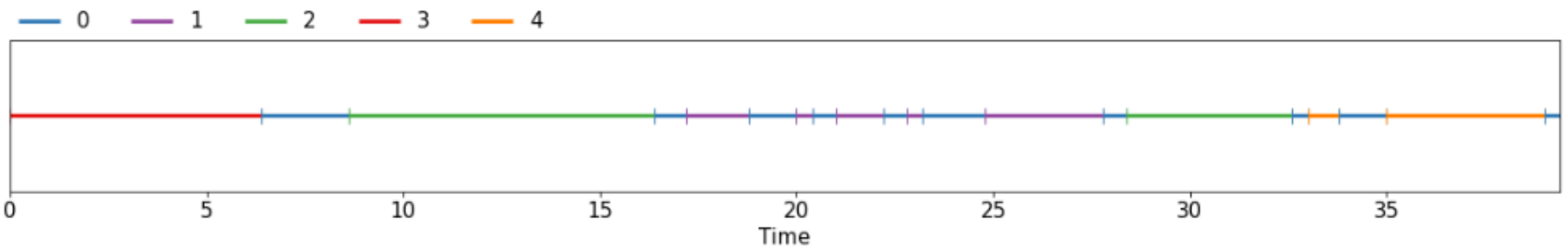
Speaker Diarization (unsupervised)

1710.10468

- d-vectors obtained with window 24 / hop 12 frames



from 'Speaker Diarization with LSTM' by Q. Wang *et al.*

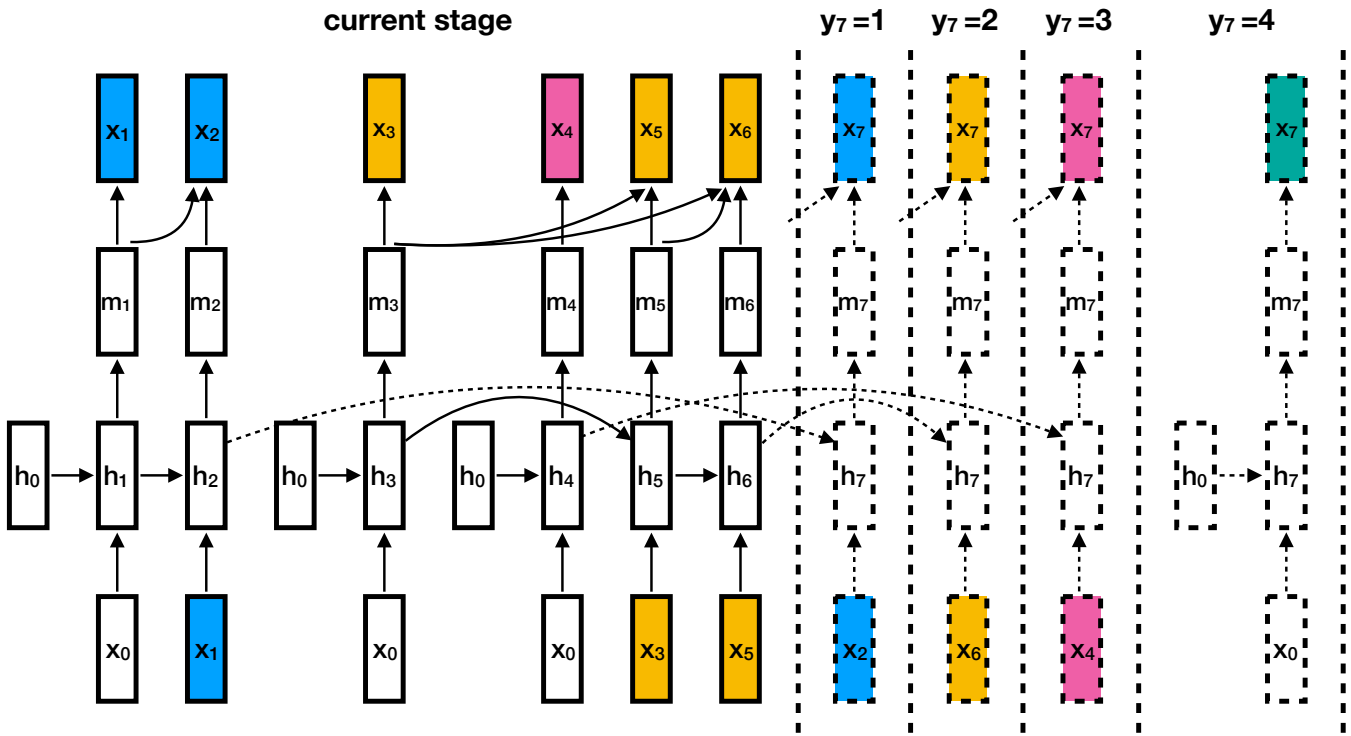


Speaker Diarization (supervised)

1810.04719

Jointly learns to:

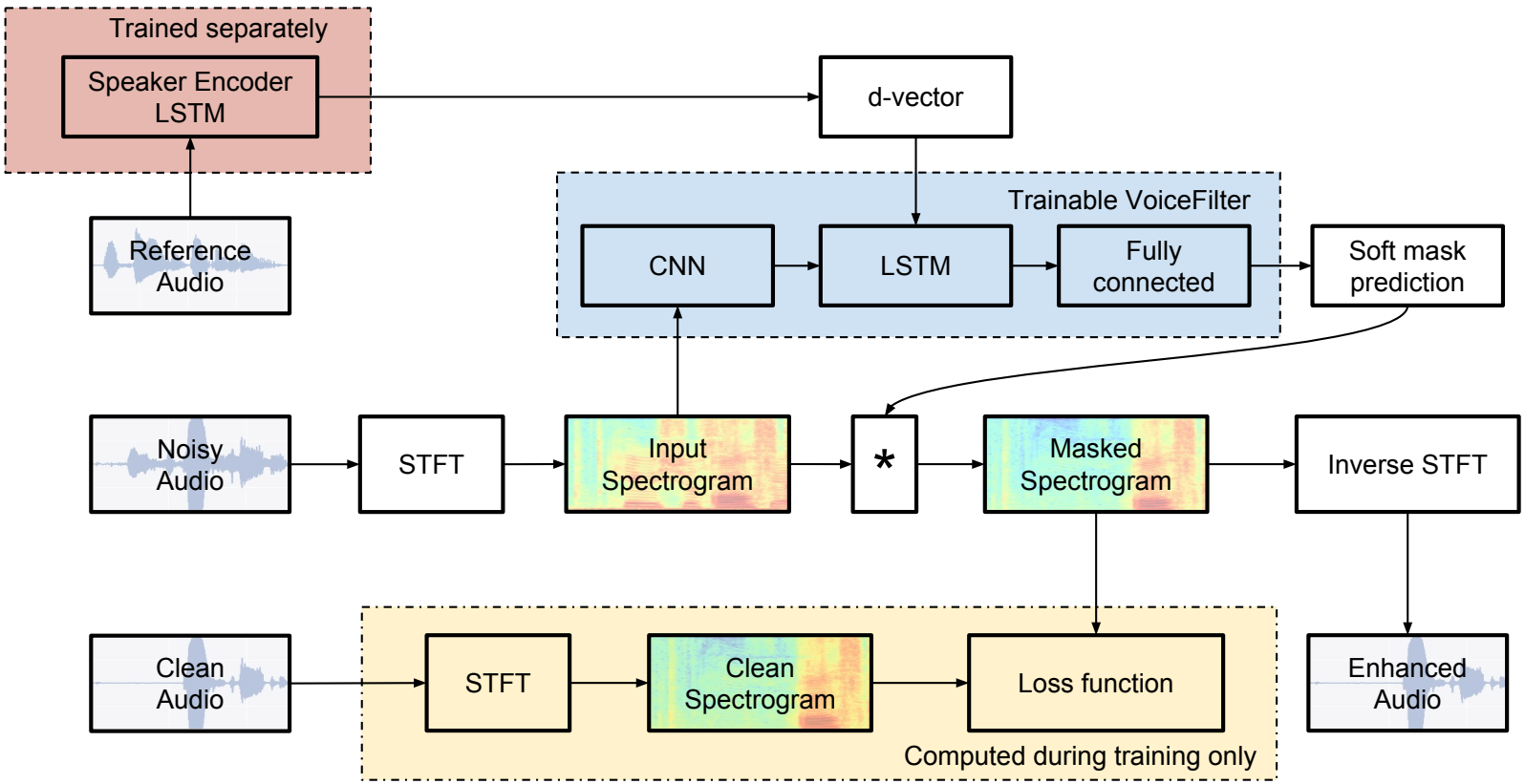
- assign speaker number / detect speaker change



from 'Fully Supervised Speaker Diarization' by A. Zhang *et al.*

Speech Separation: VoiceFilter

1810.04826



from 'VoiceFilter: Targeted Voice Separation by Speaker-Conditioned Spectrogram Masking' by Q. Wang *et al.*

Speech Separation: VoiceFilter

1810.04826

Table 1: *Parameters of the VoiceFilter network.*

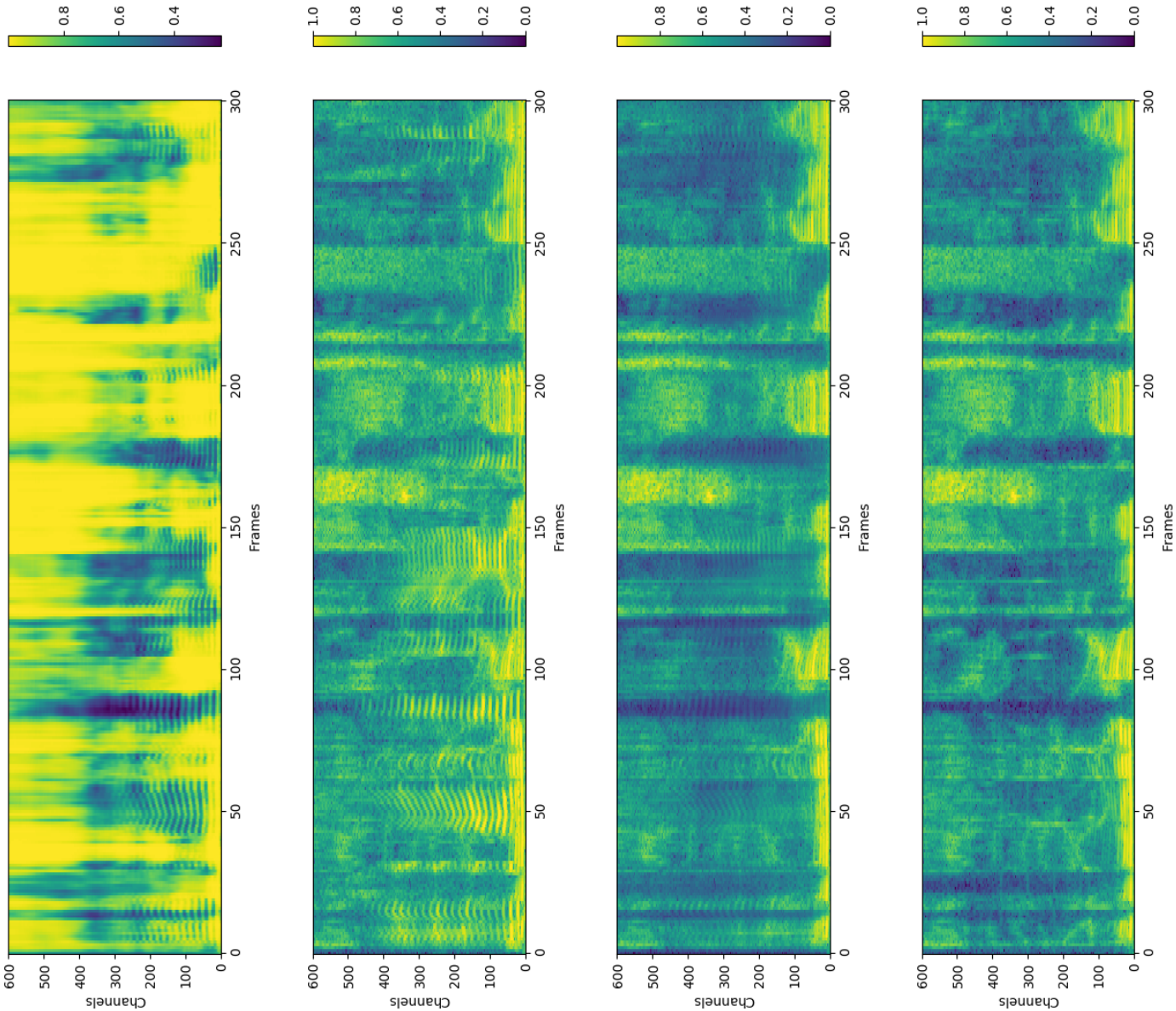
Layer	Width		Dilation		Filters / Nodes
	time	freq	time	freq	
CNN 1	1	7	1	1	64
CNN 2	7	1	1	1	64
CNN 3	5	5	1	1	64
CNN 4	5	5	2	1	64
CNN 5	5	5	4	1	64
CNN 6	5	5	8	1	64
CNN 7	5	5	16	1	64
CNN 8	1	1	1	1	8
LSTM	-	-	-	-	400
FC 1	-	-	-	-	600
FC 2	-	-	-	-	600

from 'VoiceFilter: Targeted Voice Separation by Speaker-Conditioned Spectrogram Masking' by Q. Wang *et al.*

Speech Separation: VoiceFilter

- Griffin-Lim Algorithm(1984): phase reconstruction from mag.
  - Computationally expensive, quality degradation
- Here, we use a phase from the mixed input.

```
dvec, mixed_mag, mixed_phase = batch[0]
mask = model(mixed_mag, dvec)
est_mag = mask * mixed_mag
est_wav = spec2wav(est_mag, mixed_phase)
```



mask / mixed / estimated / target

Implementation of VoiceFilter

Random thoughts on paper implementation

- `github.com/mindslab-ai/voicefilter` ★ **300+**
  - Reddit $>$ Facebook $\gg$ Twitter
- Power of template
- Things that were missing from the paper:
  - BatchNorm is crucial, but was not mentioned in paper
  - What optimizer? What loss function?
  - (arXiv paper reverse-engineering)

and...

6. Acknowledgements

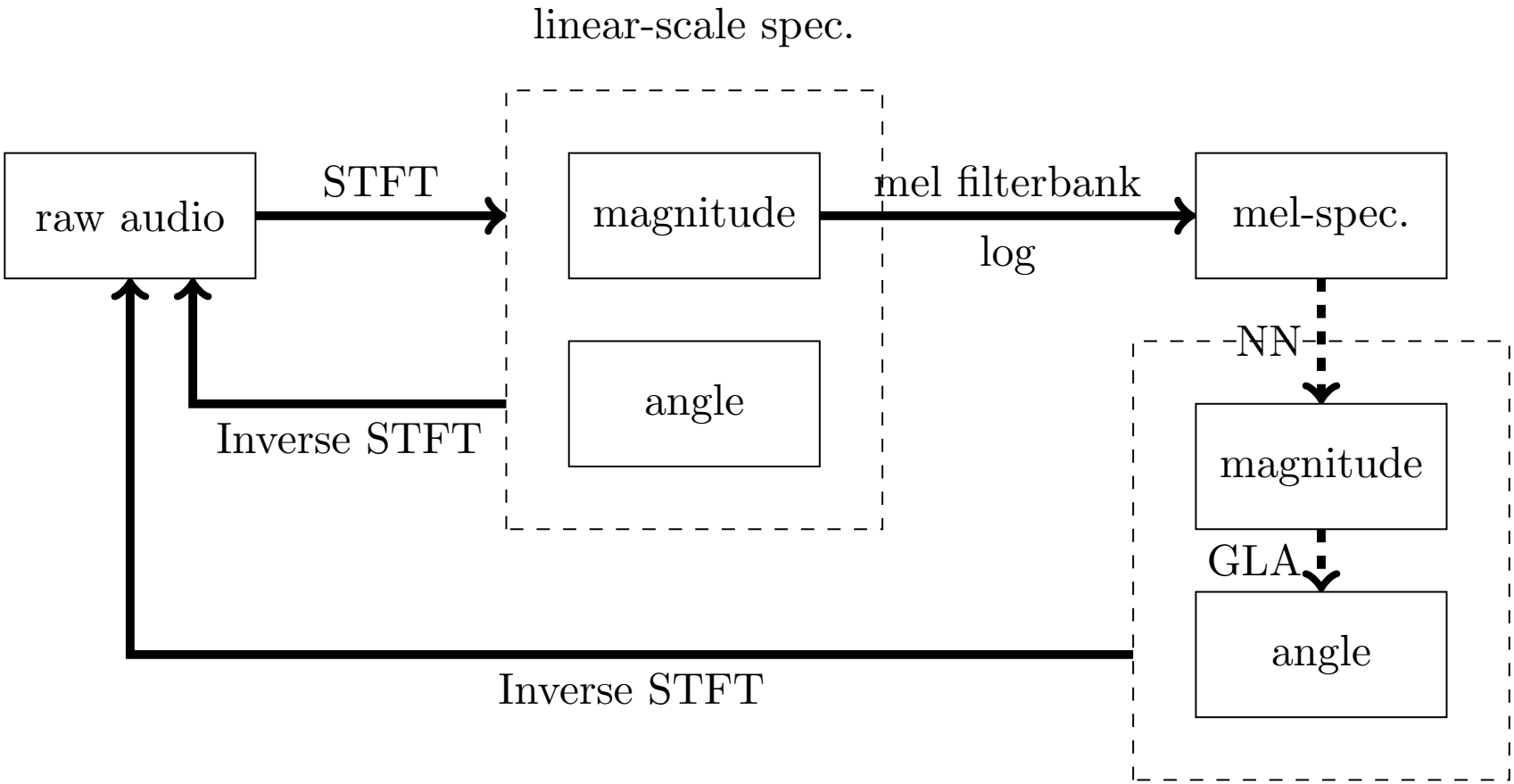
The authors would like to thank Seungwon Park for open sourcing a third-party implementation of this system.[2] We would like to thank Yiteng (Arden) Huang, Jason Pelecanos, and Fadi Biadsy for the helpful discussions.

[2] https://github.com/mindslab-ai/voicefilter

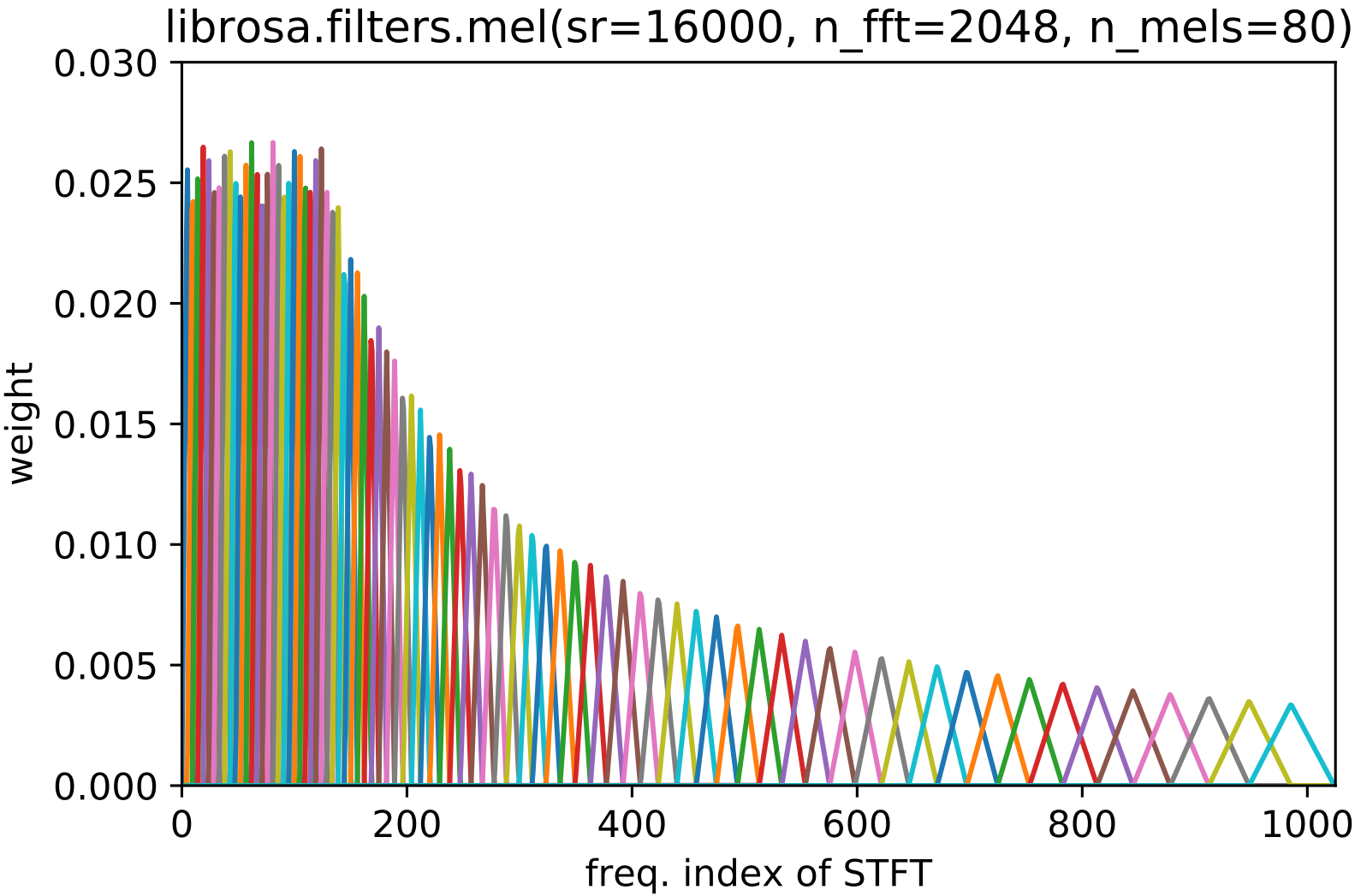
from 'VoiceFilter: Targeted Voice Separation by Speaker-Conditioned Spectrogram Masking' by Q. Wang *et al.*

Thank You

Appendix: Griffin-Lim vocoder

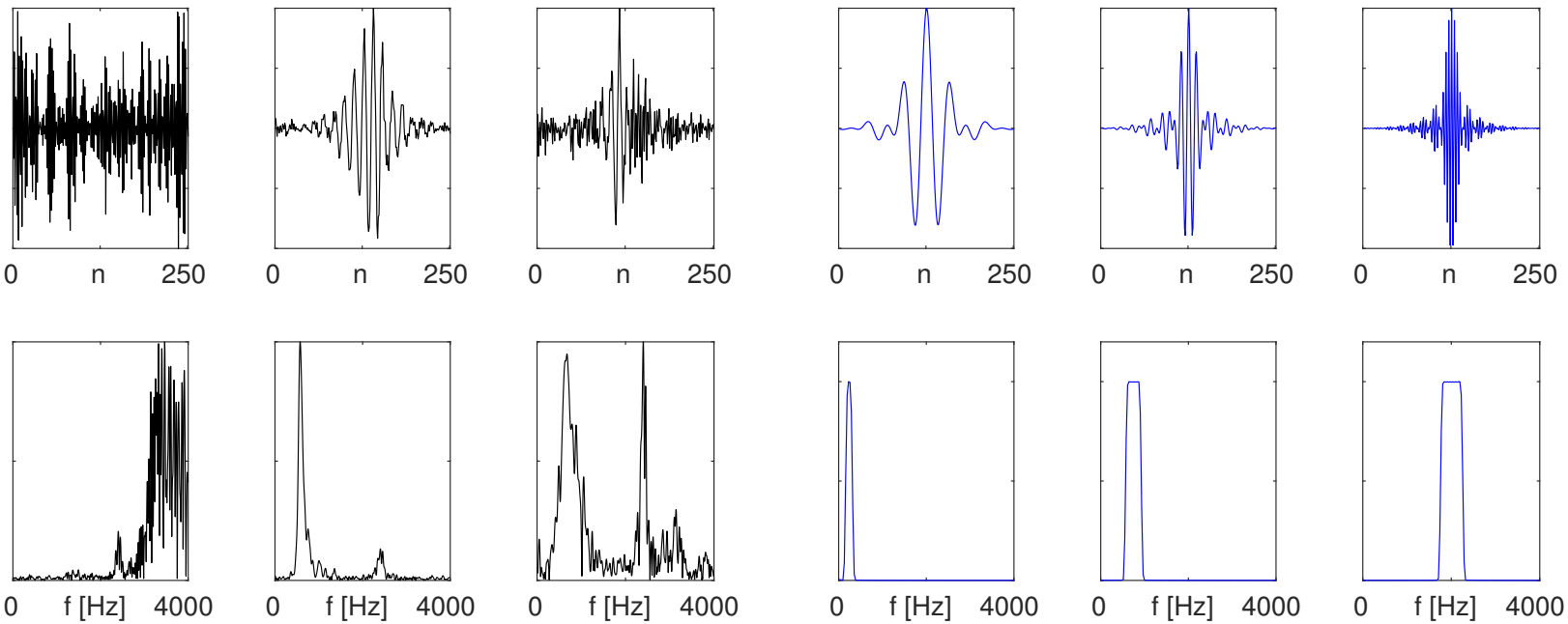


Appendix: Mel-scale filter



Appendix: SincNet

1808.00158

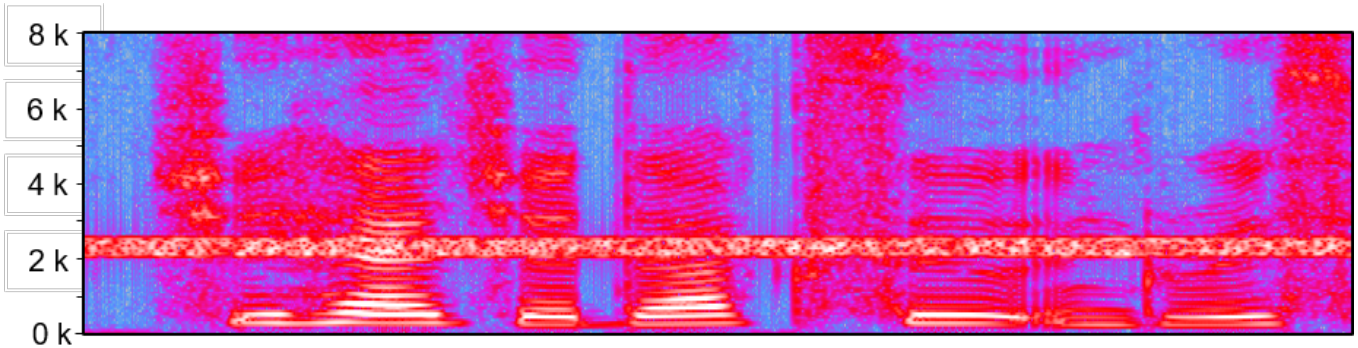


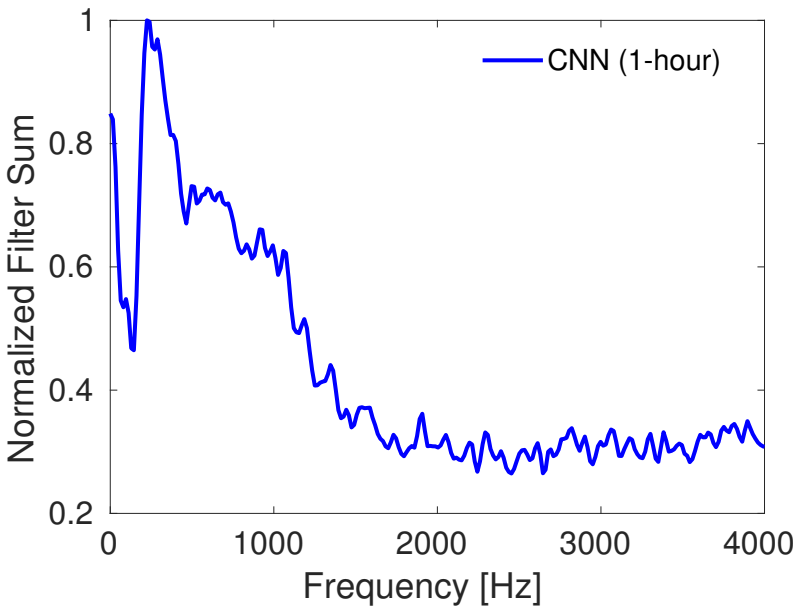
from 'Speaker Recognition from Raw Waveform with SincNet'

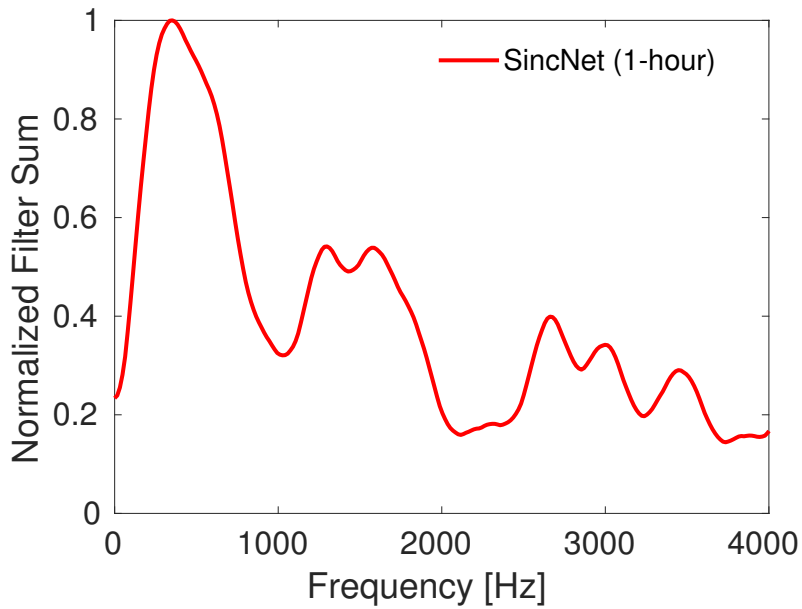
by M. Ravanelli, Y. Bengio

Appendix: SincNet

1811.09725



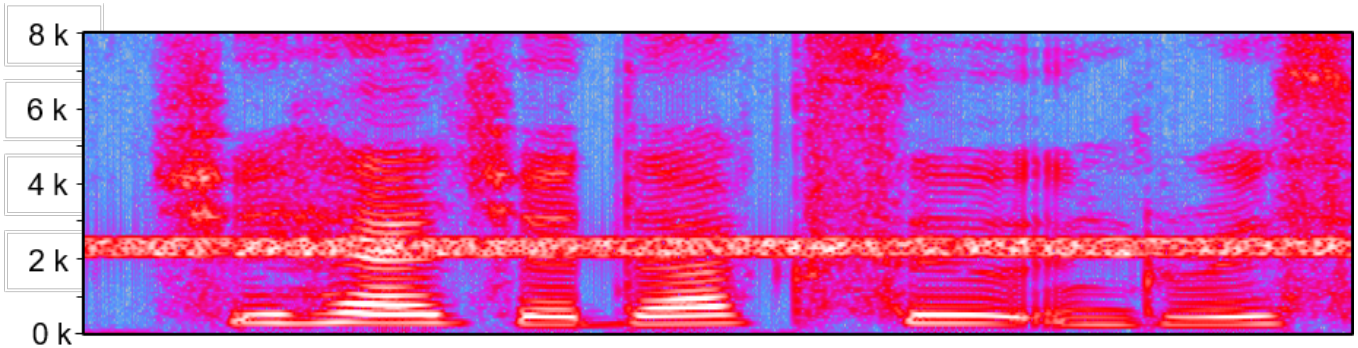


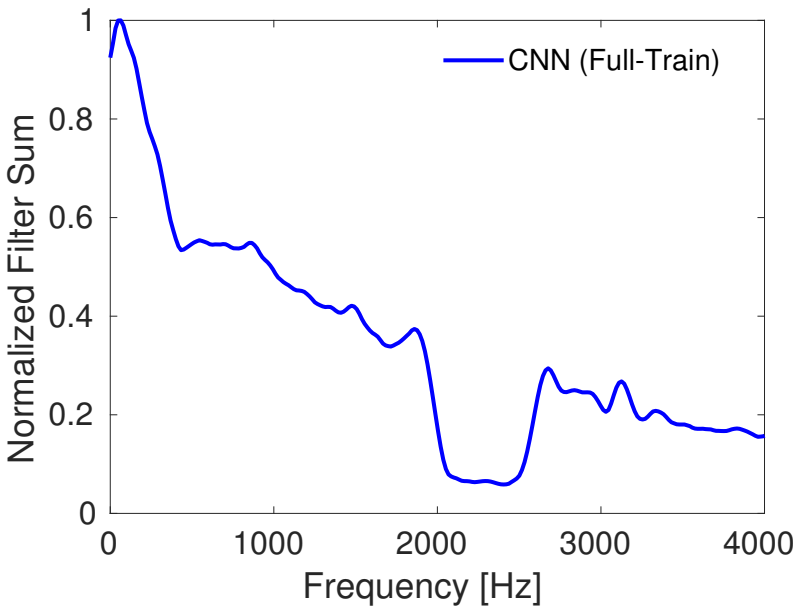


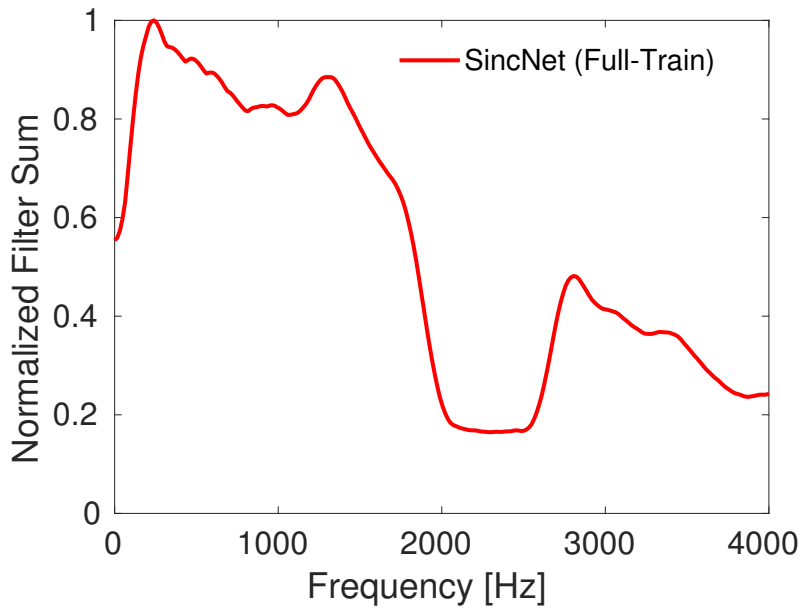
'Interpretable Convolutional Filters with SincNet', M. Ravanelli, Y. Bengio

Appendix: SincNet

1811.09725







'Interpretable Convolutional Filters with SincNet', M. Ravanelli, Y. Bengio