Neural Architectures for Music Representation Learning

Sanghyuk Chun, Clova AI Research

NAVER CLOVA

Contents

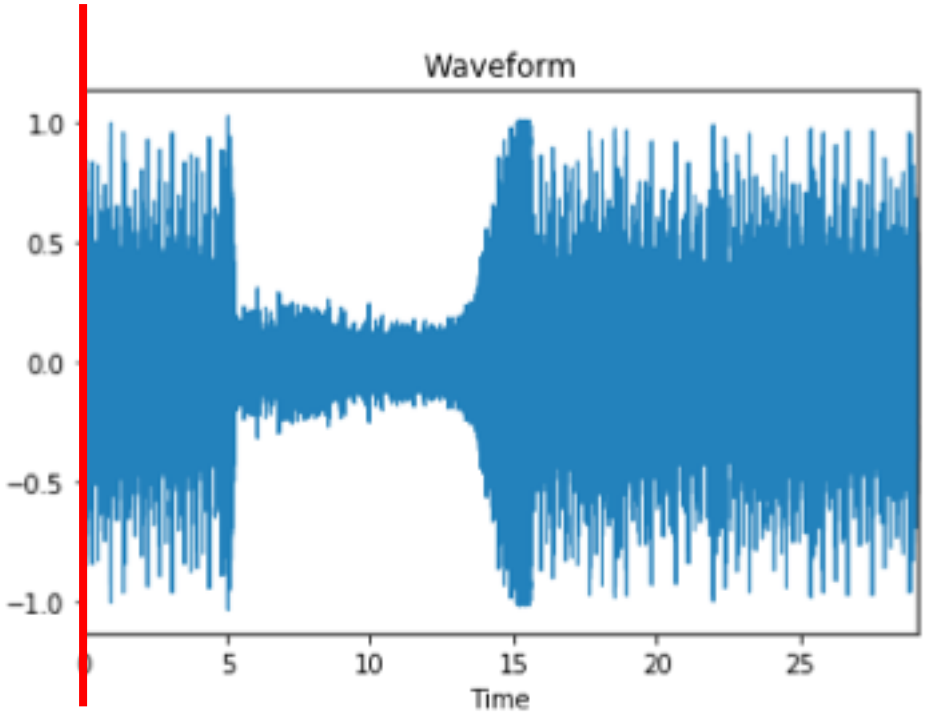
Understanding audio signals

Raw audio

Spectrogram

Mel filter bank

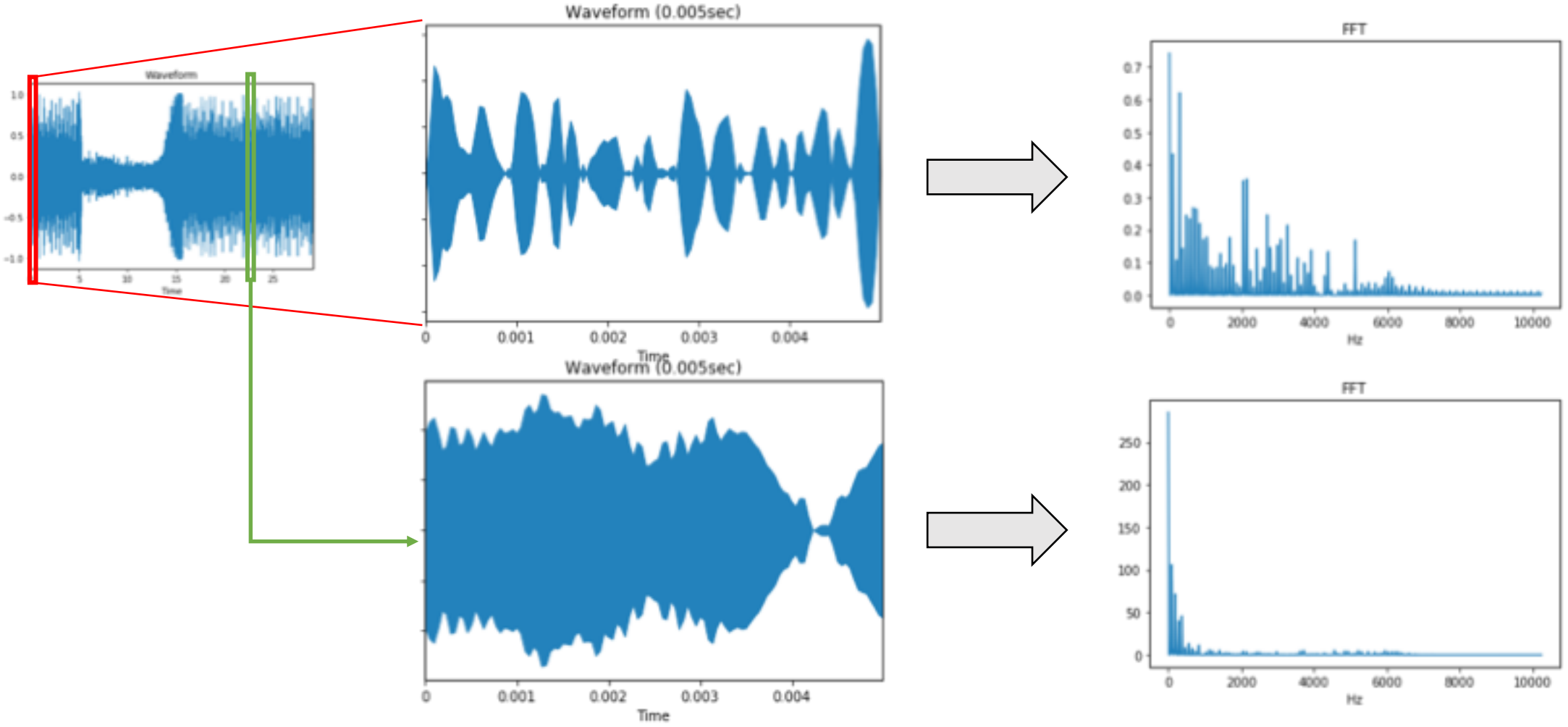
Understanding audio signals in time domain.



[0.001, -0.002, -0.005, -0.004, -0.003, -0.003, -0.003, -0.002, -0.001, ...]

“Waveform” shows “magnitude” of the input signal across time.
How can we capture “frequency” information?

Understanding audio signals in frequency domain.

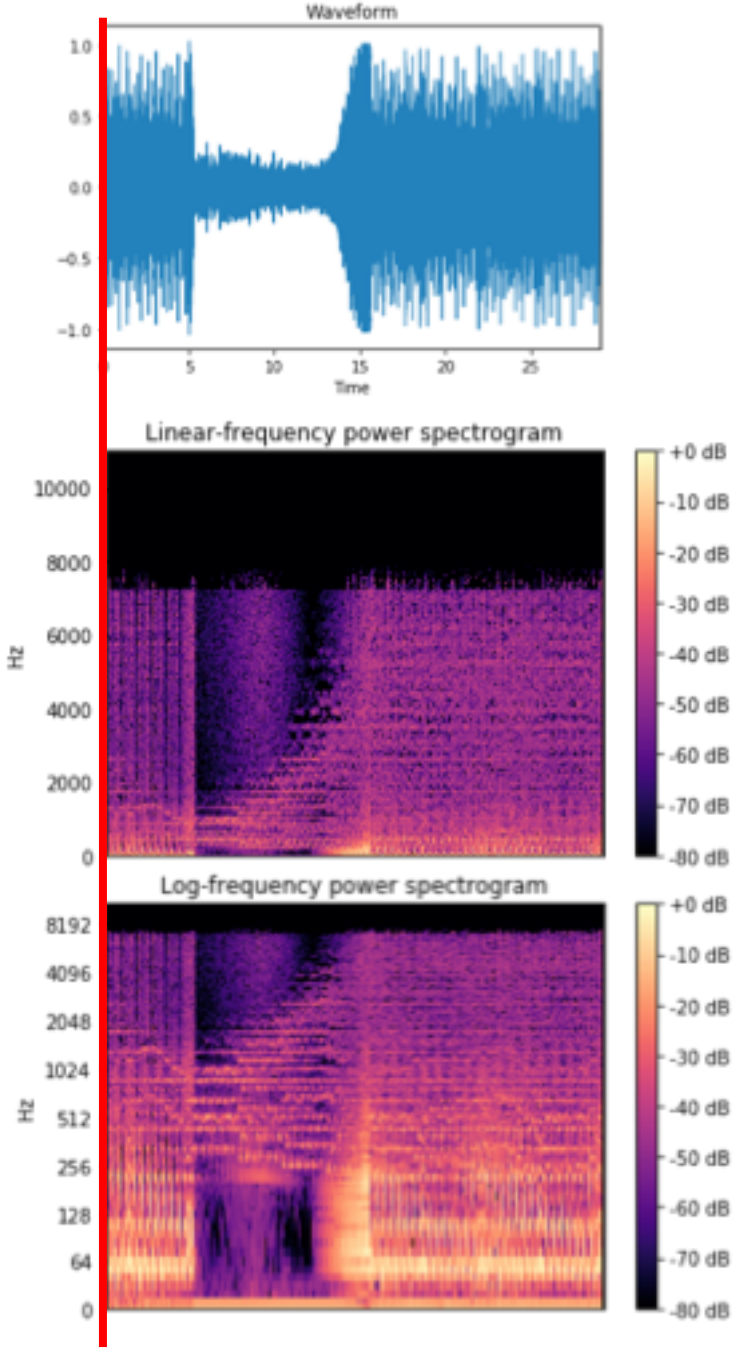


Time-amplitude => Frequency-amplitude

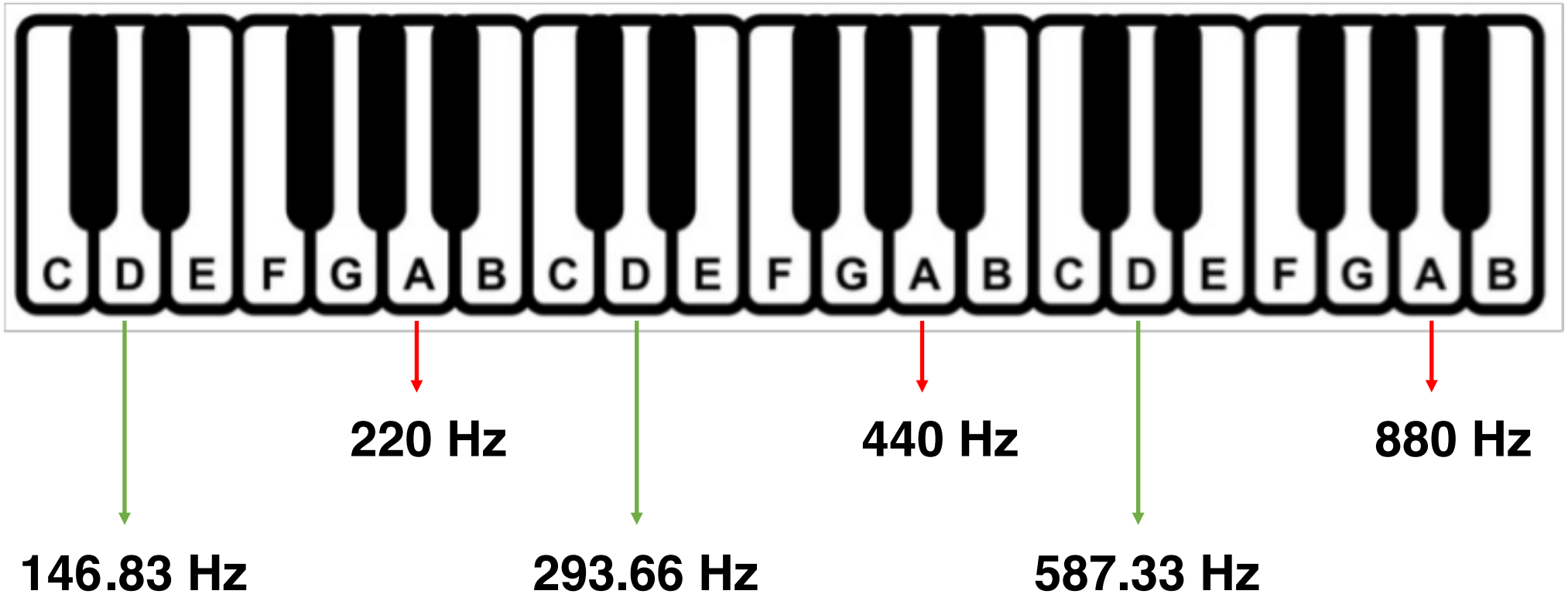
Understanding audio signals in time-frequency domain.

Types for audio inputs

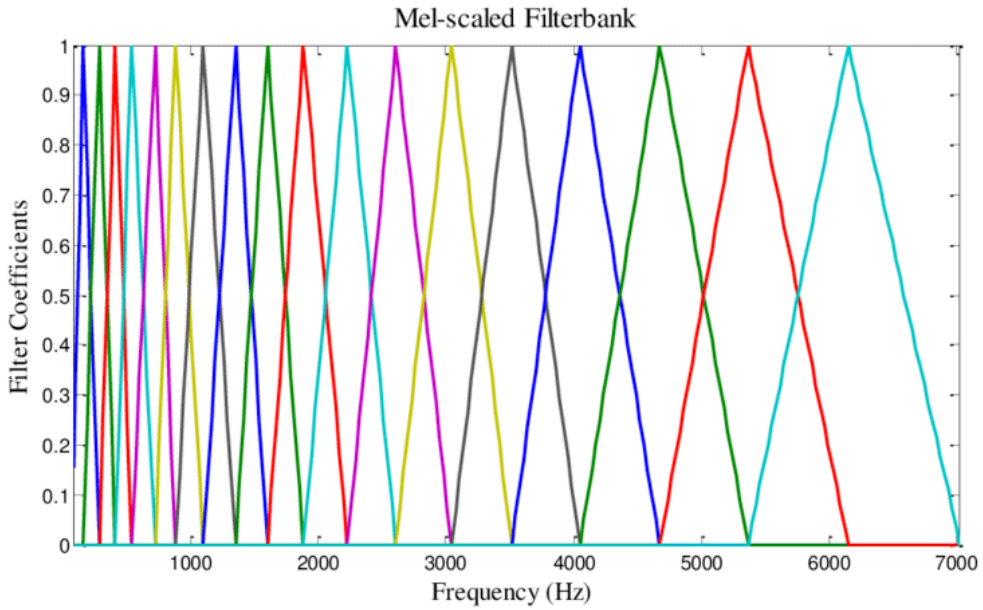
- Raw audio waveform
- Linear spectrogram
- Log-scale spectrogram
- **Mel spectrogram**
- Constant Q transform (CQT)

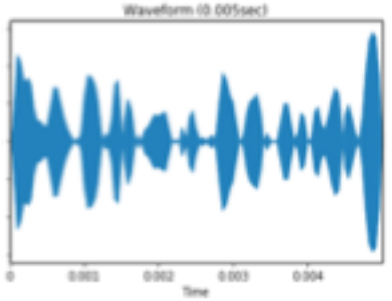


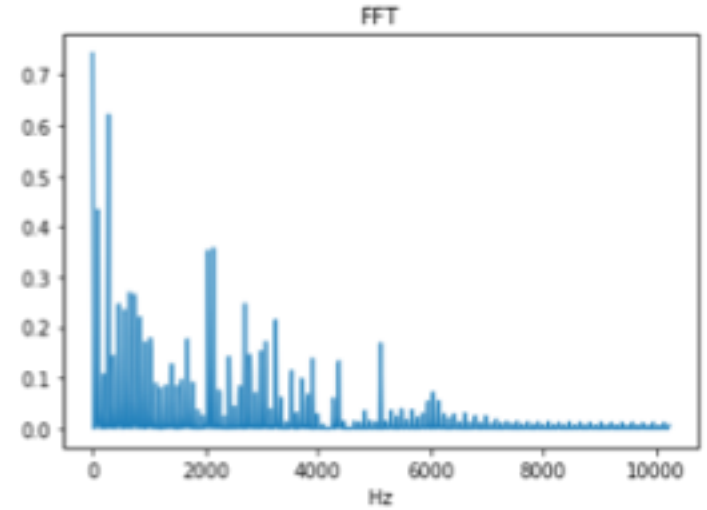
Human perception for audio: Log-scale.

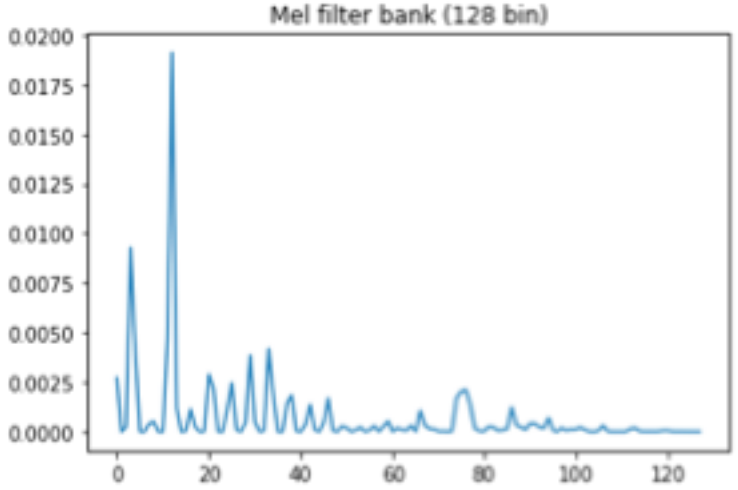


Mel filter banks: Log-scale filter bank.

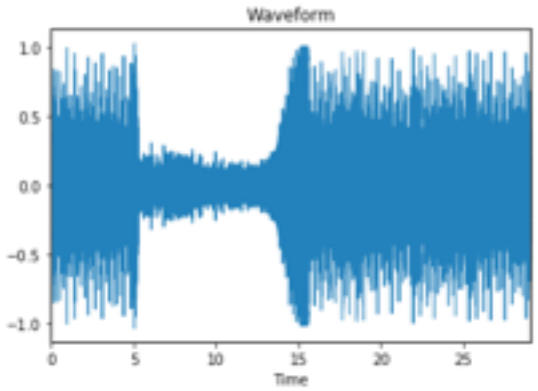
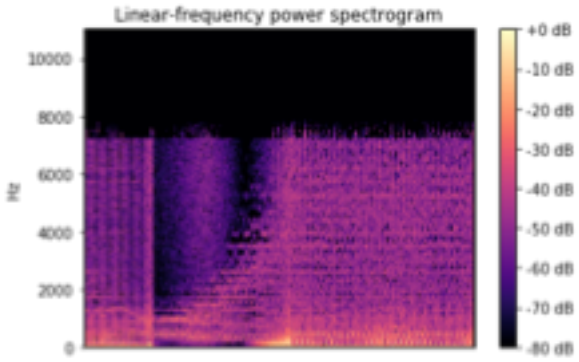
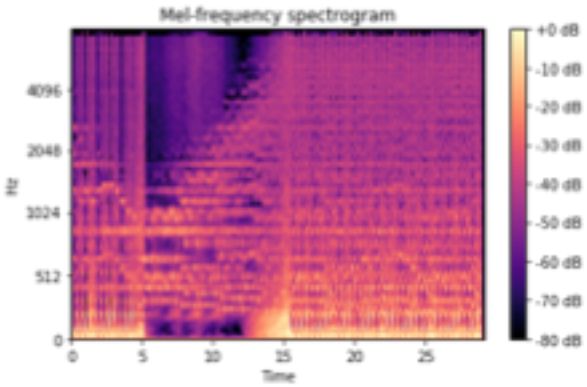






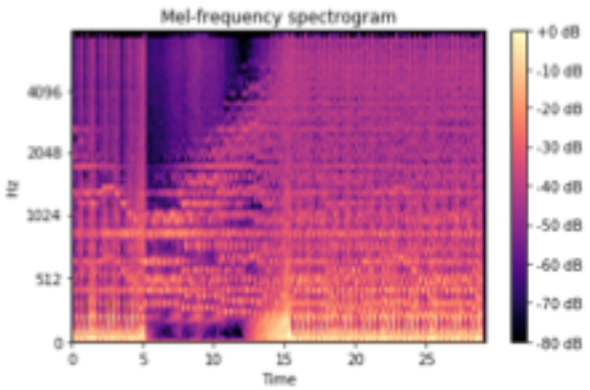


Mel-spectrogram.

	Raw audio	Linear spectrogram	Mel-spectrogram
			
		Related to many hyperparams (hop size, window size, …)	
Input shape	1D: Sampling rate * audio length = 11025 * 30 = 330K	2D: (# fft / 2 + 1) X # frames = (2048 / 2 + 1) X 1255 = [1025, 1255]	2D: (# mel bins) X # frames = [128, 1255]
Information density	Very sparse in time axis (need very large receptive field) 1 sec = sampling rate	Sparse in freq axis (need large receptive field) Each time bin has 1025 dim	Less sparse in freq axis Each time bin has 128 dim
loss	No loss (if SR > Nyquist rate)	“Resolution”	“Resolution” + Mel filter

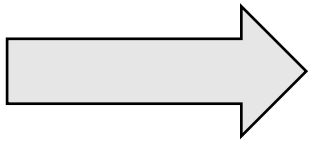
Bonus:
MFCC (Mel-Frequency Cepstral Coefficient).

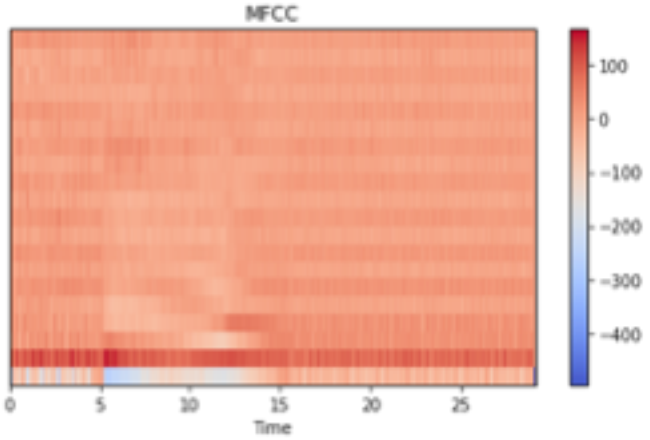
Mel-spectrogram



2D: (# mel bins) X # frames
= [128, 1255]

DCT (Discrete Cosine Transform)





2D: (# mfcc) X # frames
= [20, 1255]

Frequently used in the speech domain.
Very lossy representation to the high-level music representation learning (c.f., SIFT)

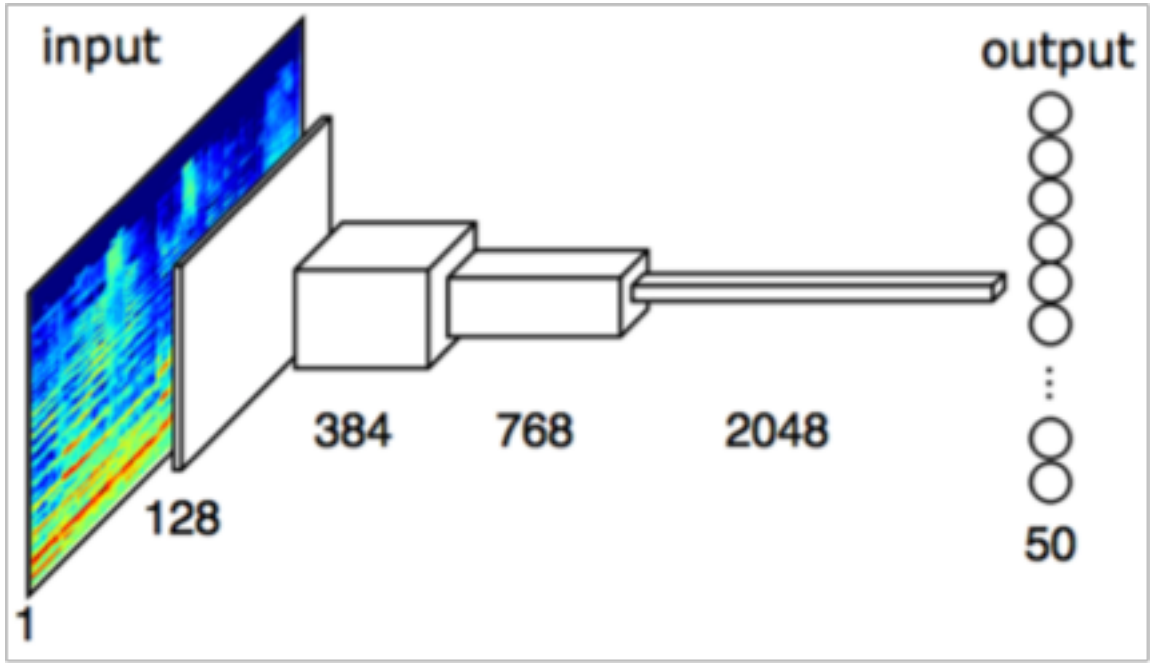
Front-end and back-end framework

Fully convolutional neural network baseline

Rethinking convolutional neural network

Front-end and back-end framework

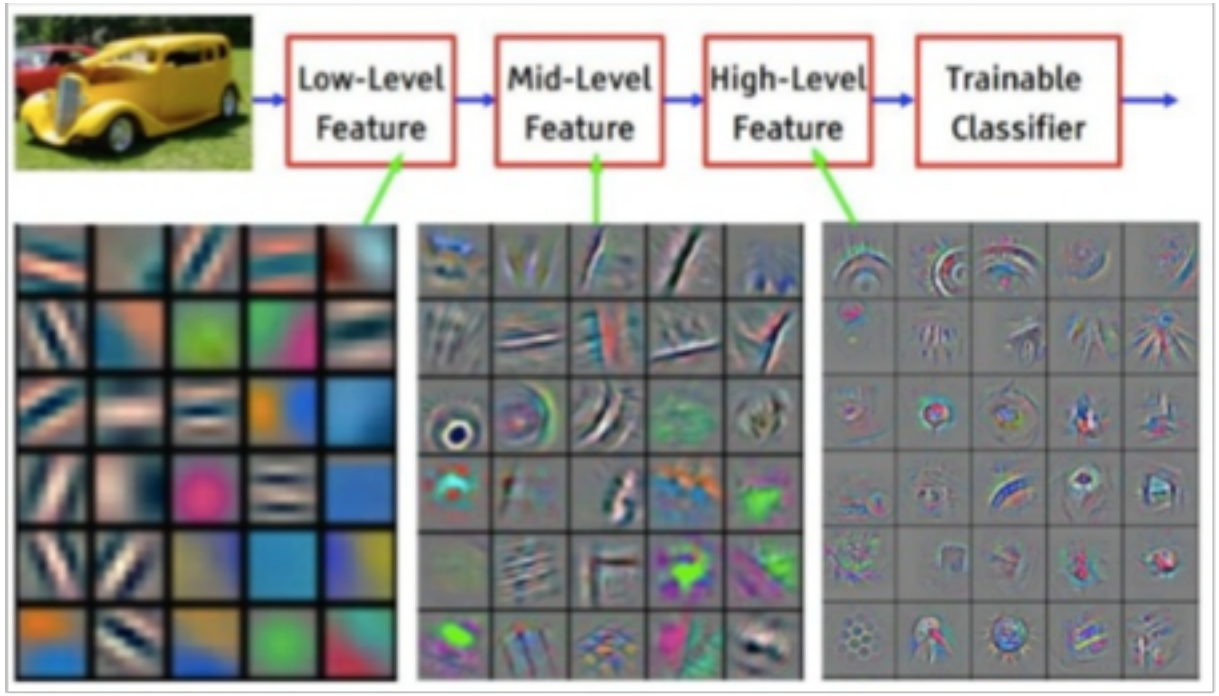
Fully convolutional neural network baseline for automatic music tagging.



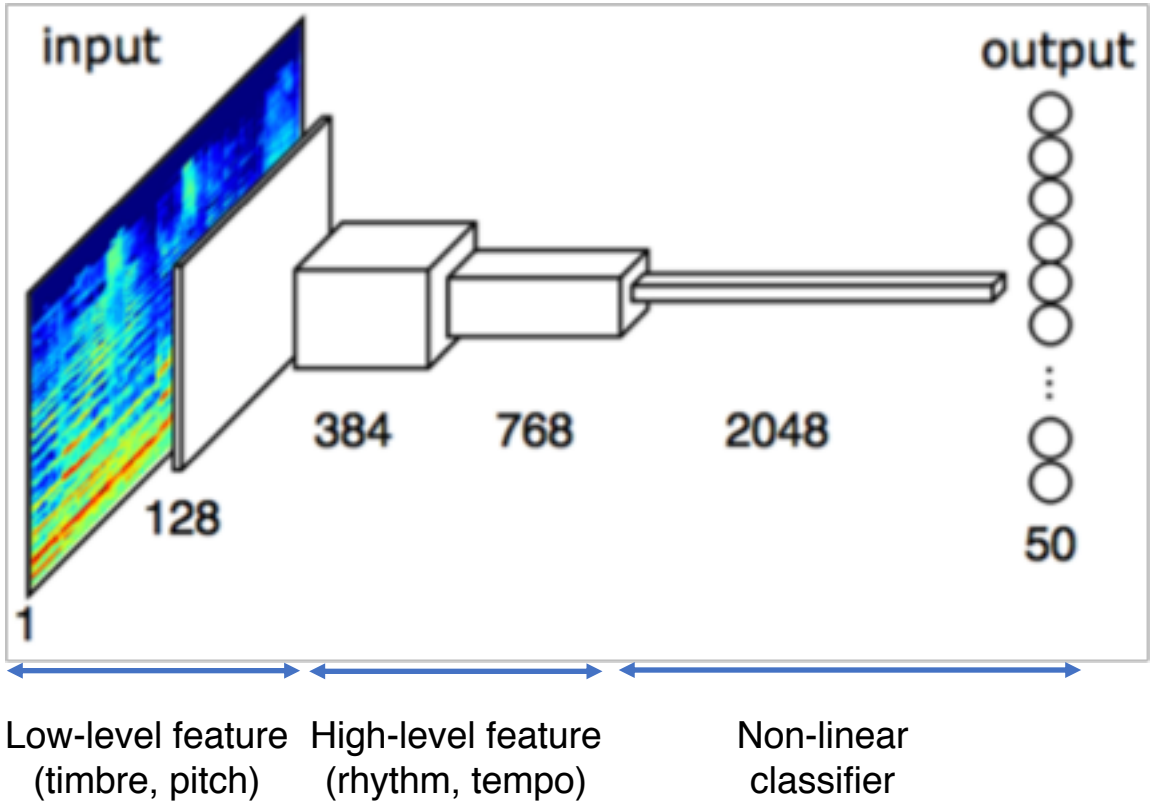
FCN-4
Mel-spectrogram *(input: 96×1366×1)*
Conv 3×3×128
MP (2, 4) *(output: 48×341×128)*
Conv 3×3×384
MP (4, 5) *(output: 24×85×384)*
Conv 3×3×768
MP (3, 8) *(output: 12×21×768)*
Conv 3×3×2048
MP (4, 8) *(output: 1×1×2048)*
Output 50×1 (sigmoid)

[ISMIR 2016] "Automatic tagging using deep convolutional neural networks.", Keunwoo Choi, George Fazekas, Mark Sandler

Rethinking CNN as feature extractor and non-linear classifier.



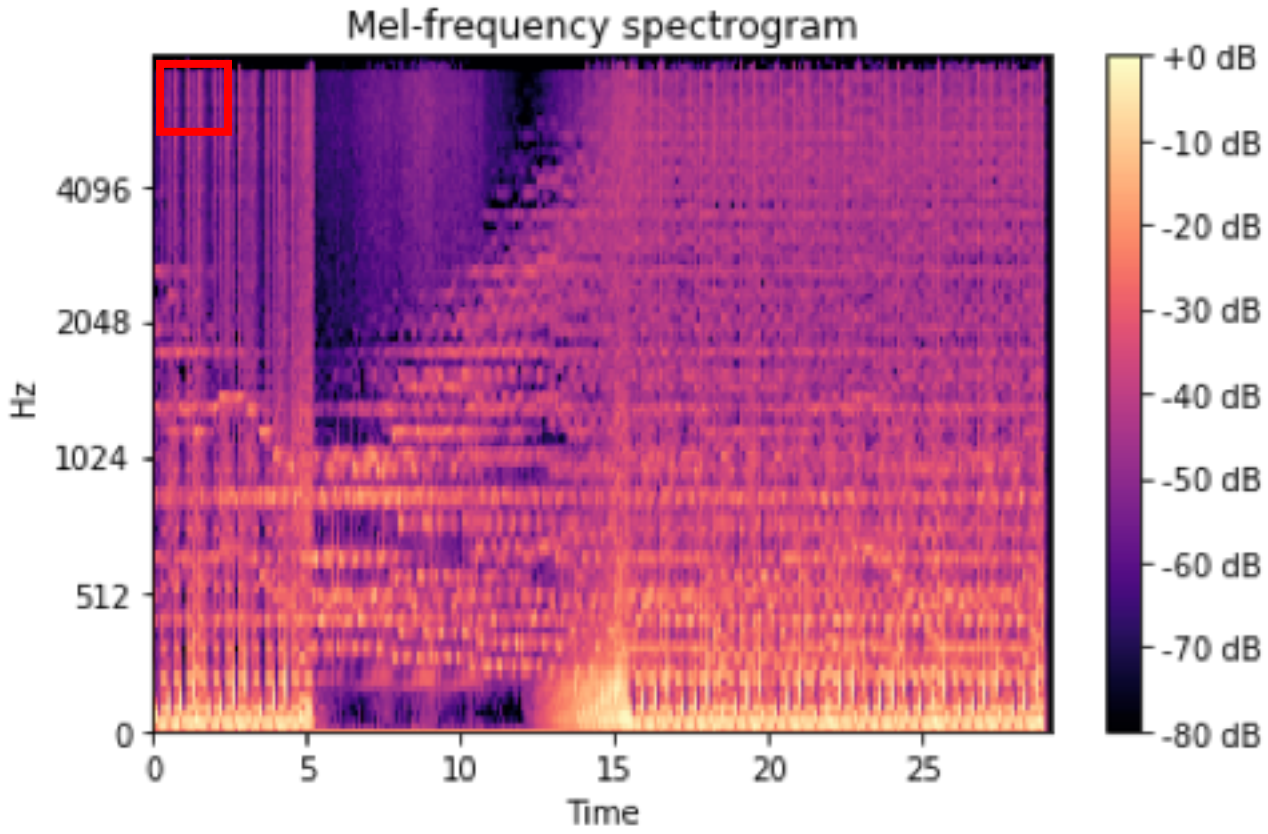
Rethinking CNN as feature extractor and non-linear classifier.



FCN-4
Mel-spectrogram *(input: 96×1366×1)*
Conv 3×3×128
MP (2, 4) *(output: 48×341×128)*
Conv 3×3×384
MP (4, 5) *(output: 24×85×384)*
Conv 3×3×768
MP (3, 8) *(output: 12×21×768)*
Conv 3×3×2048
MP (4, 8) *(output: 1×1×2048)*
Output 50×1 (sigmoid)

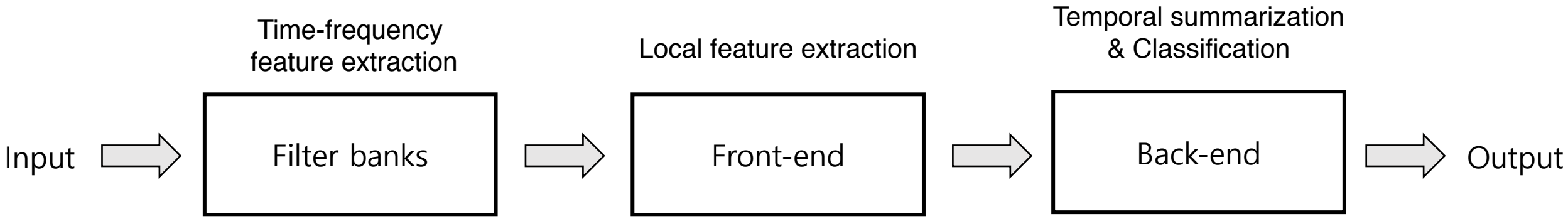
[ISMIR 2016] "Automatic tagging using deep convolutional neural networks.", Keunwoo Choi, George Fazekas, Mark Sandler

Rethinking CNN as feature extractor and non-linear classifier.

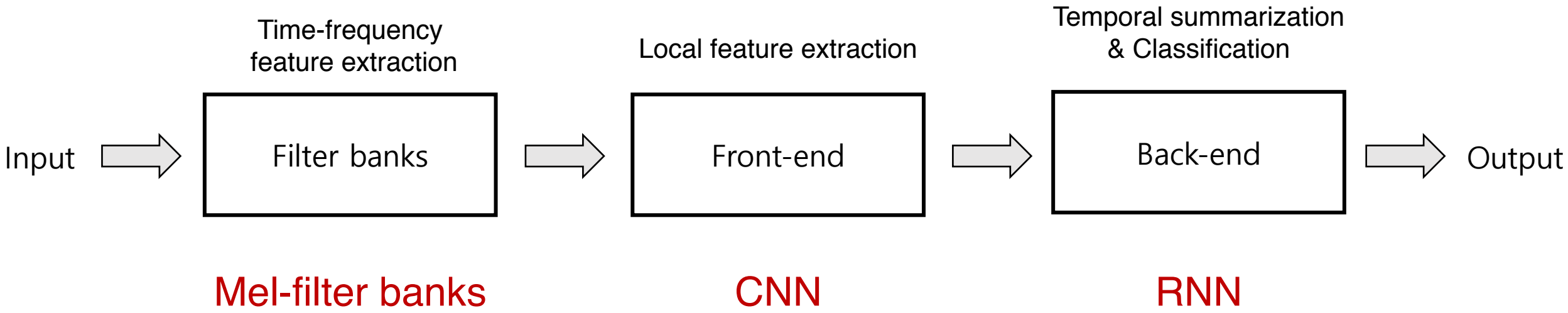


FCN-4
Mel-spectrogram *(input: 96×1366×1)*
Conv 3×3×128
MP (2, 4) *(output: 48×341×128)*
Conv 3×3×384
MP (4, 5) *(output: 24×85×384)*
Conv 3×3×768
MP (3, 8) *(output: 12×21×768)*
Conv 3×3×2048
MP (4, 8) *(output: 1×1×2048)*
Output 50×1 (sigmoid)

Front-end and back-end framework

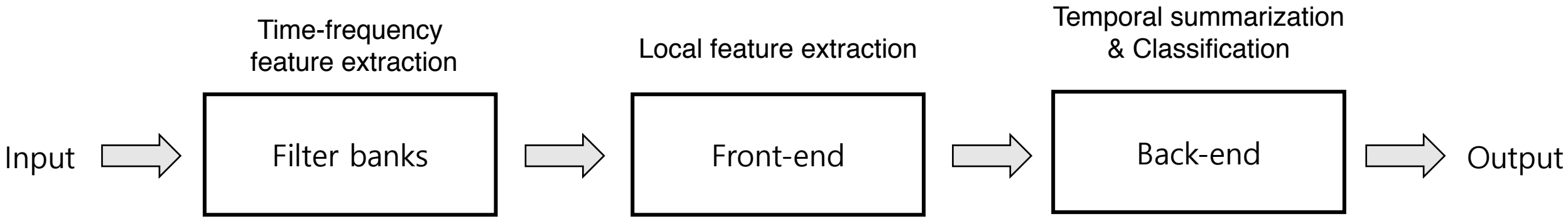


Front-end and back-end framework



[ICASSP 2017] "Convolutional recurrent neural networks for music classification.", Choi, et al.

Front-end and back-end framework

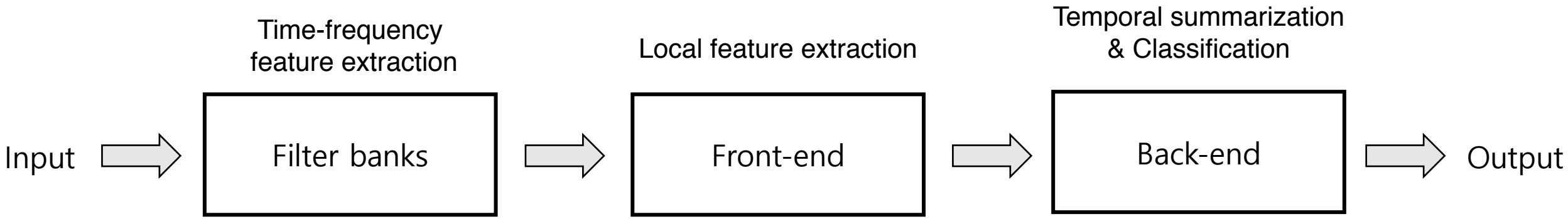


Mel-filter banks CNN RNN

Mel-filter banks CNN CNN

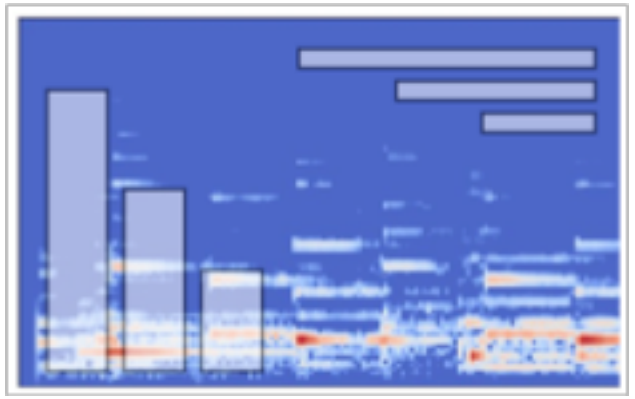
[ISMIR 2018] "End-to-end learning for music audio tagging at scale.", Pons, et al.

Front-end and back-end framework



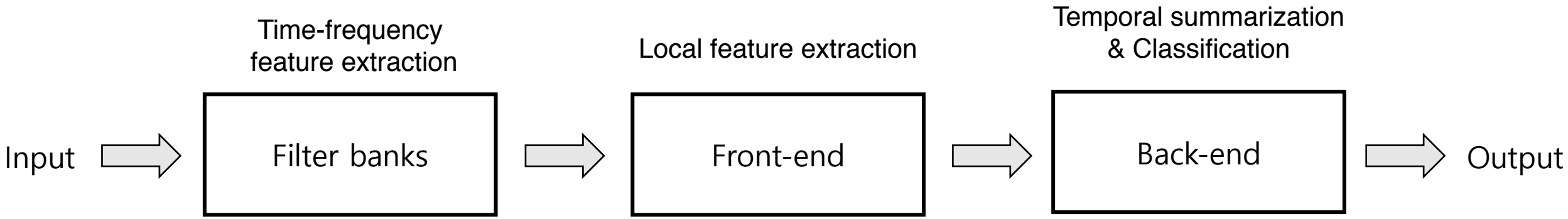
Mel-filter banks | CNN | RNN

Mel-filter banks | CNN | CNN



[ISMIR 2018] "End-to-end learning for music audio tagging at scale.", Pons, et al.

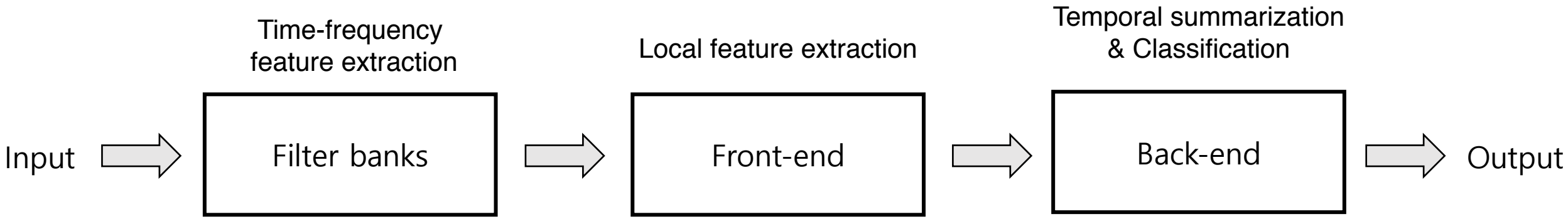
Front-end and back-end framework



Mel-filter banks	CNN	RNN
Mel-filter banks	CNN	CNN
Fully-learnable	CNN	CNN

[SMC 2017] "Sample-level Deep Convolutional Neural Networks for Music Auto-tagging Using Raw Waveforms.", Lee, et al.

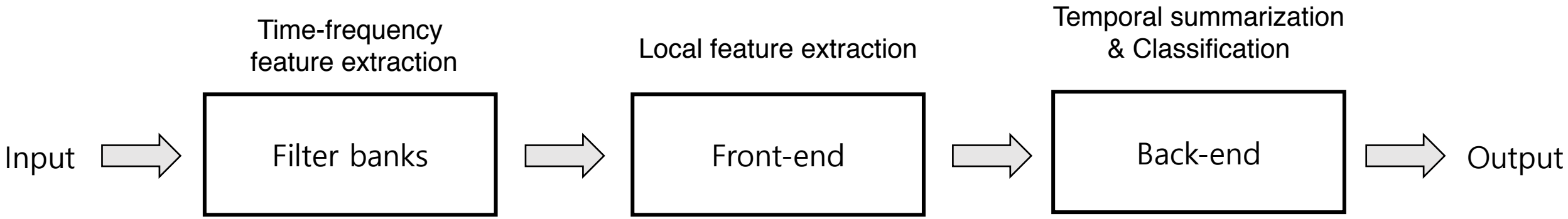
Front-end and back-end framework



Mel-filter banks	CNN	RNN
Mel-filter banks	CNN	CNN
Fully-learnable	CNN	CNN
Partially-learnable	CNN	CNN

[ICASSP 2020] "Data-driven Harmonic Filters for Audio Representation Learning.", Won, et al.

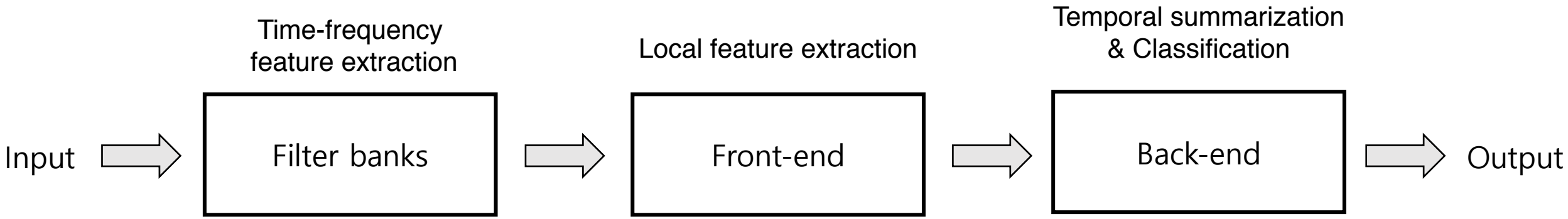
Front-end and back-end framework



Mel-filter banks	CNN	RNN
Mel-filter banks	CNN	CNN
Fully-learnable	CNN	CNN
Partially-learnable	CNN	CNN
Mel-filter banks	CNN	Self-attention

[ArXiv 2019] "Toward Interpretable Music Tagging with Self-attention.", Won, et al.

Front-end and back-end framework

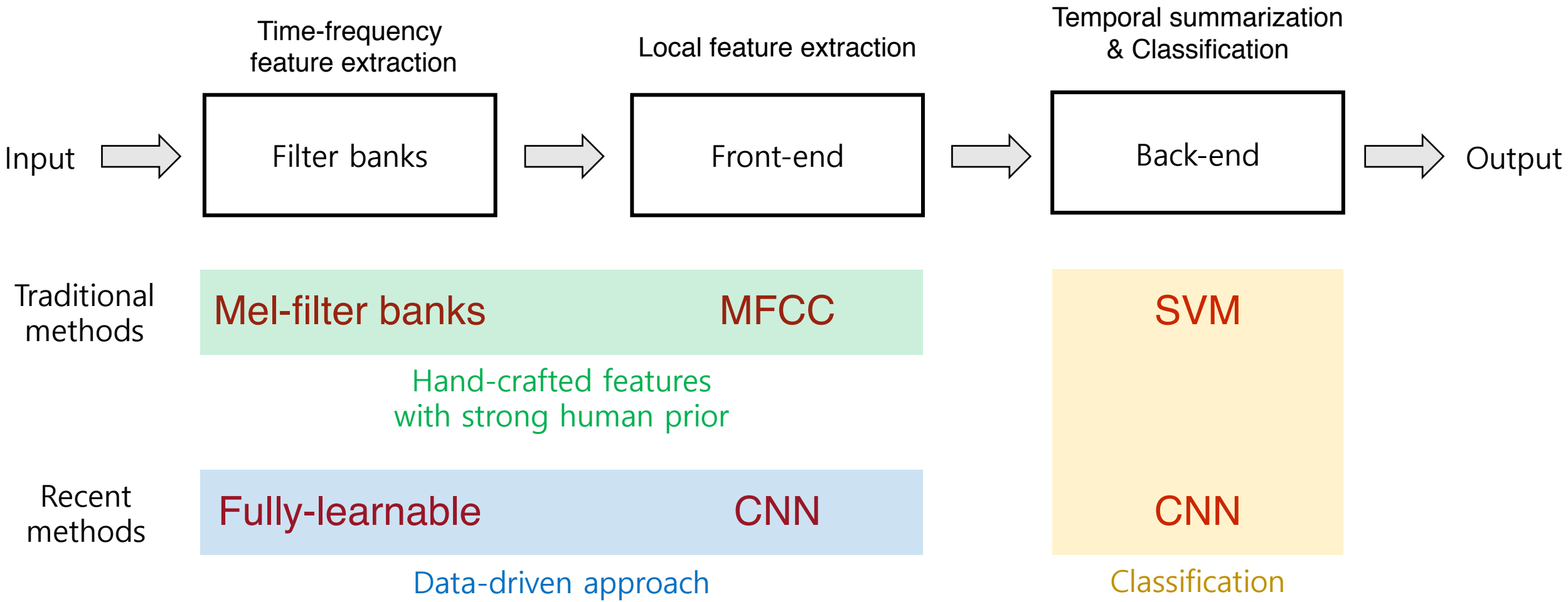


Mel-filter banks	CNN	RNN
Mel-filter banks	CNN	CNN
Fully-learnable	CNN	CNN
Partially-learnable	CNN	CNN
Mel-filter banks	CNN	Self-attention

Powerful front-end with Harmonic filter banks

[ISMIR 2019 Late Break Demo] Automatic Music Tagging with Harmonic CNN.
[ICASSP 2020] Data-driven Harmonic Filters for Audio Representation Learning.
[SMC 2020] Evaluation of CNN-based Automatic Music Tagging Models.

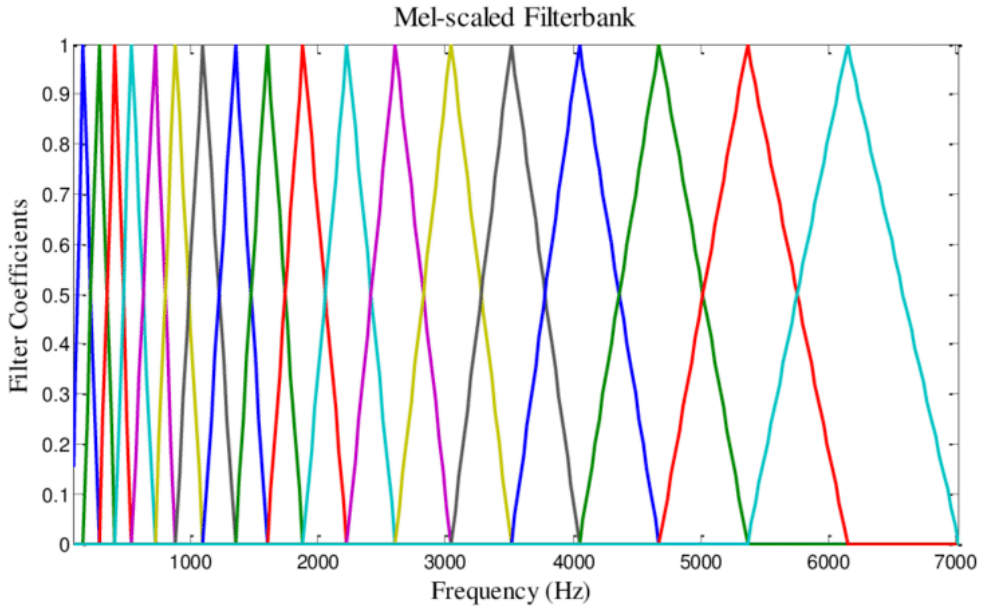
Motivation: Data-driven, but human-guided



DATA-DRIVEN HARMONIC FILTERS

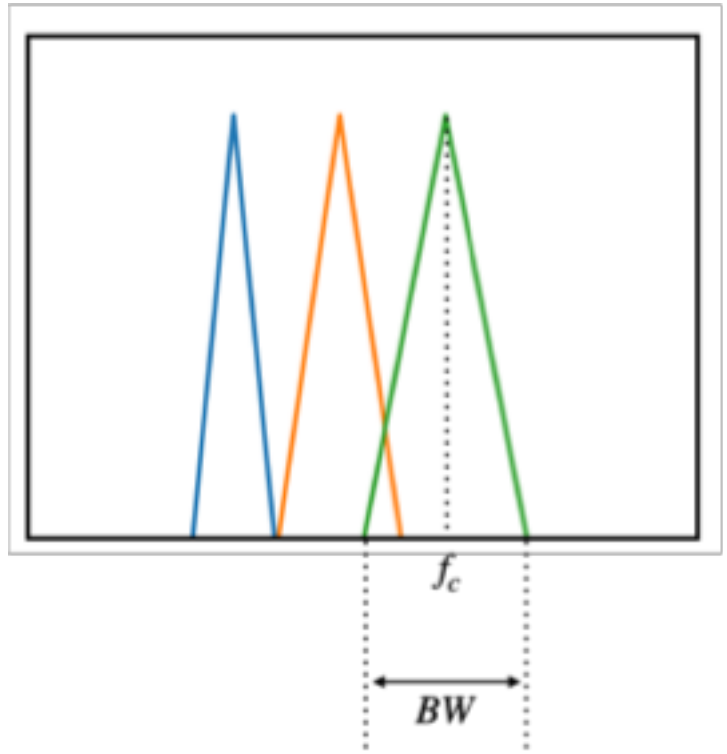
DATA-DRIVEN HARMONIC FILTERS

Data-driven filter banks



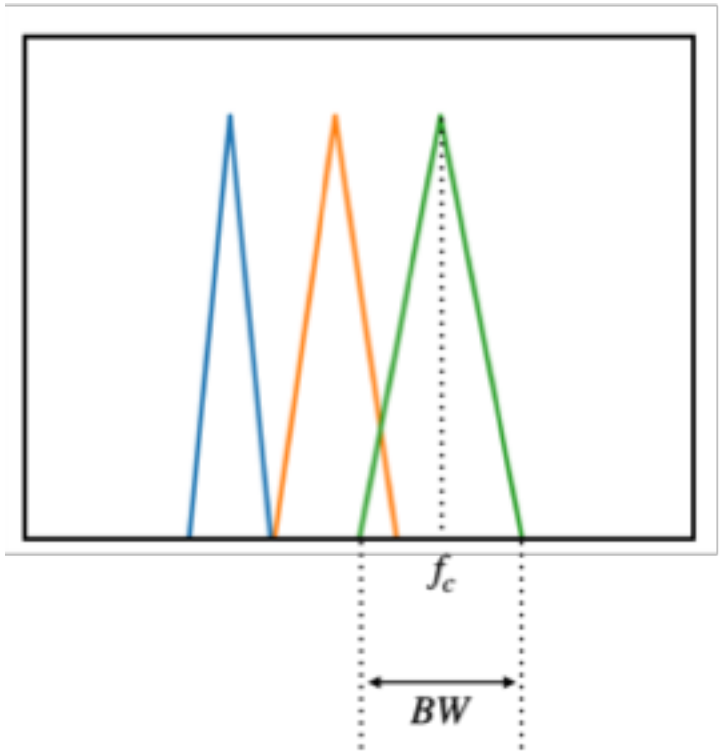
$$H_m(k) = \begin{cases} 0 & k < f(m-1) \\ \dfrac{k - f(m-1)}{f(m) - f(m-1)} & f(m-1) \le k \le f(m) \\ \dfrac{f(m+1) - k}{f(m+1) - f(m)} & f(m) \le k \le f(m+1) \\ 0 & k > f(m+1) \end{cases}$$

f(m): pre-defined frequency values depending on Sampling rate, FFT size, mel bin size, …



Proposed data-driven filter is parameterized by
- f_c: center frequency
- BW: bandwidth

Data-driven filter banks



$$\Lambda(f; f_c, BW) = \left[1 - \frac{2|f - fc|}{BW}\right]_+$$

Derived from
equivalent rectangular bandwidth (ERB)

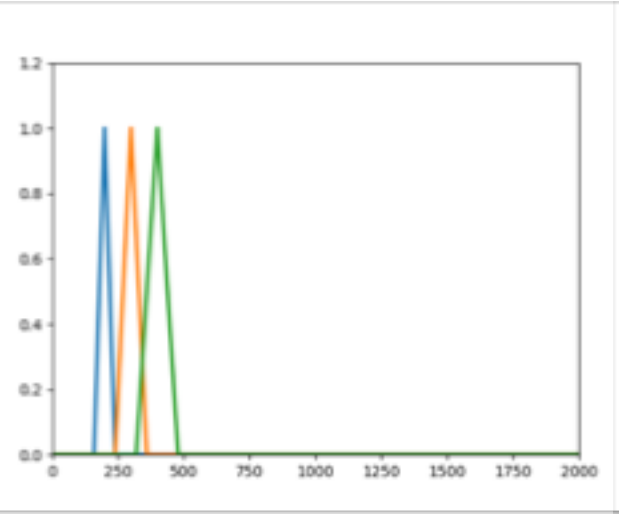
$$BW \simeq 0.1079 f_c + 24.7$$

$$BW = (\alpha f_c + \beta)/Q$$

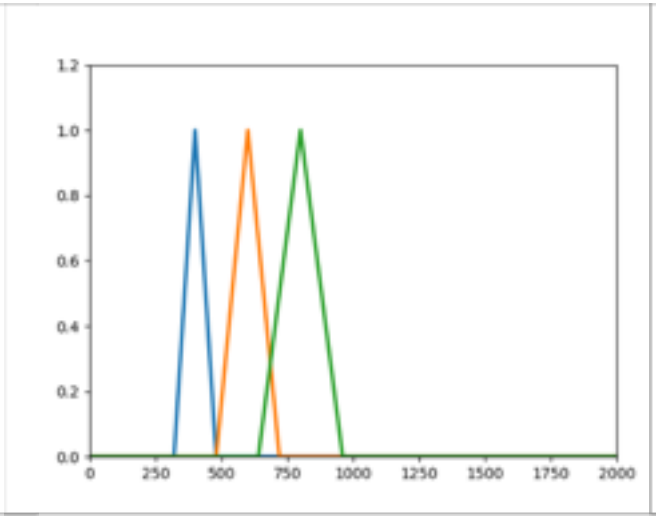
Trainable Q!

DATA-DRIVEN HARMONIC FILTERS

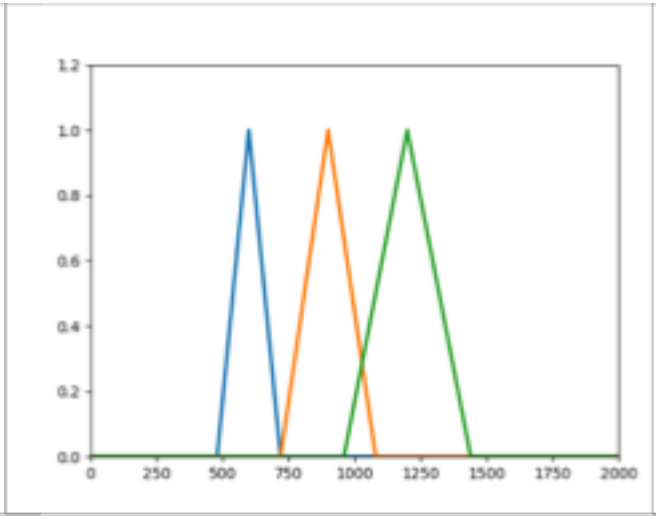
Harmonic filters



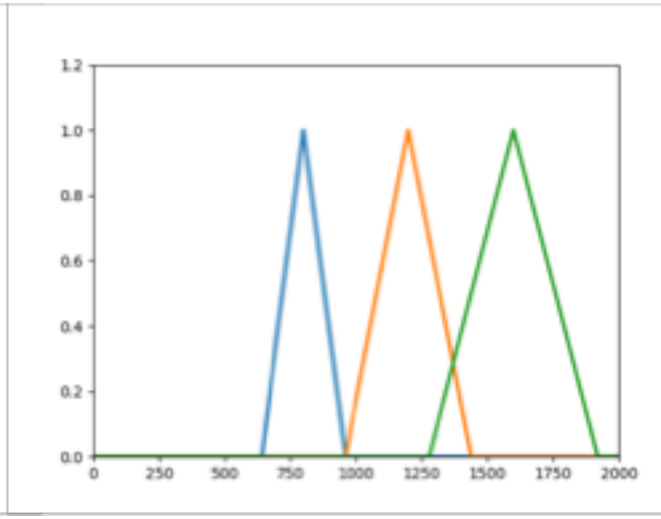
n=1



n=2



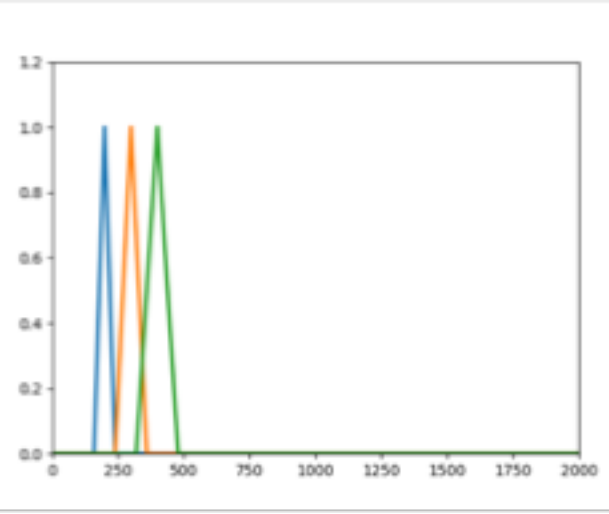
n=3



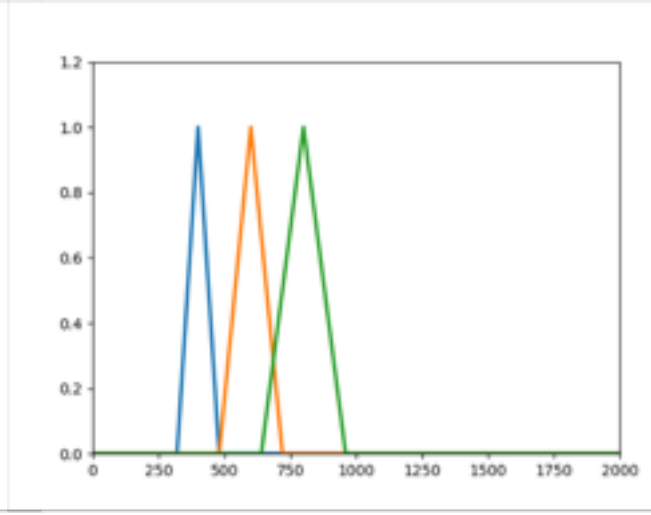
n=4

$$\Lambda_n(f; f_c, \alpha, \beta, Q) = \left[1 - \frac{2|f - n \cdot f_c|}{(n \cdot \alpha f_c + \beta)/Q}\right]_+$$

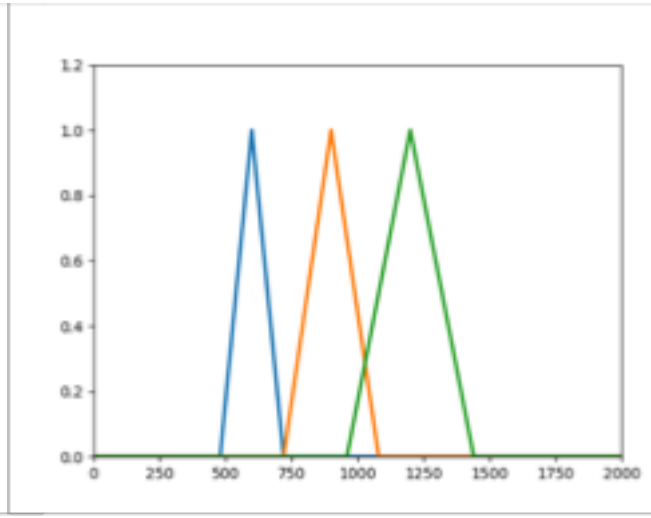
Output of harmonic filters



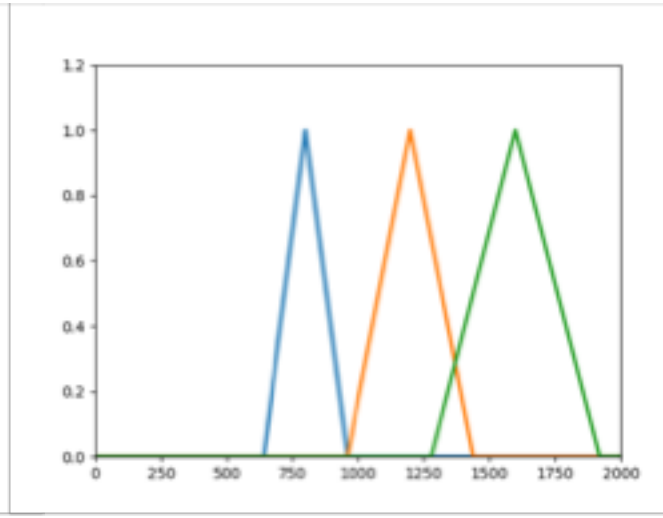
n=1



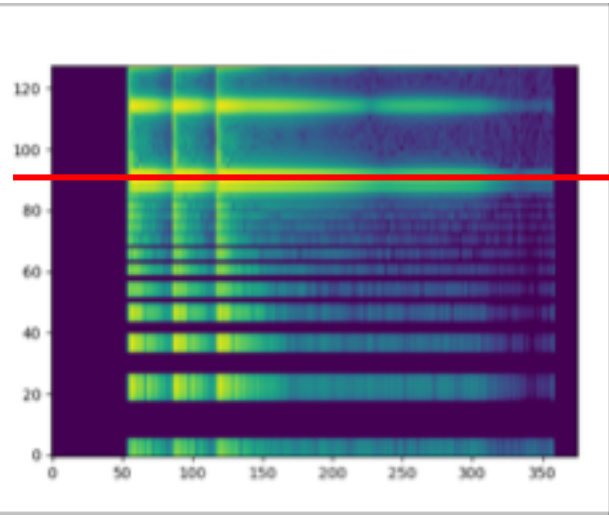
n=2



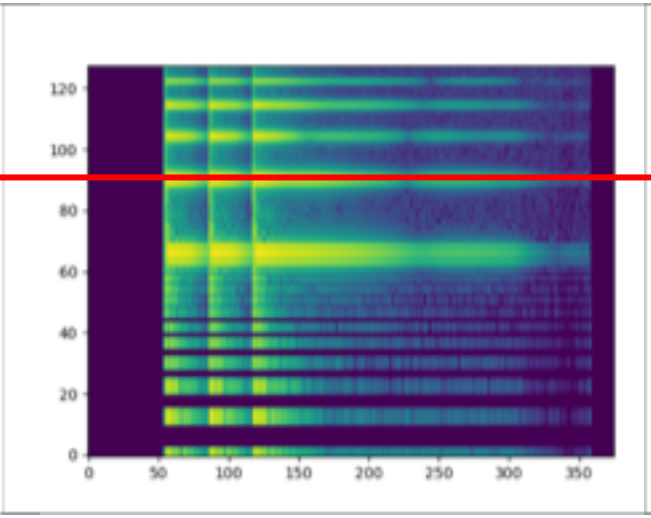
n=3



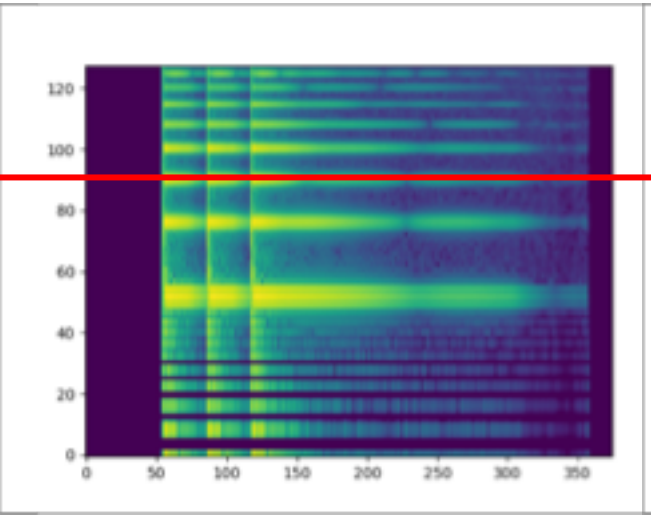
n=4



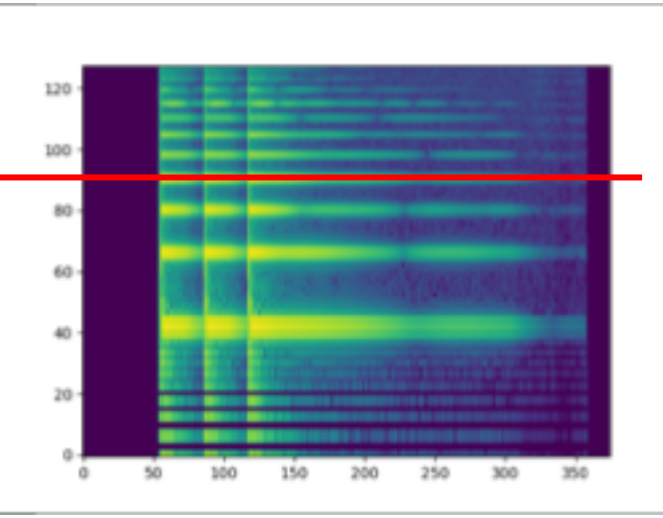
Fundamental frequency



2nd harmonic of Fund freq

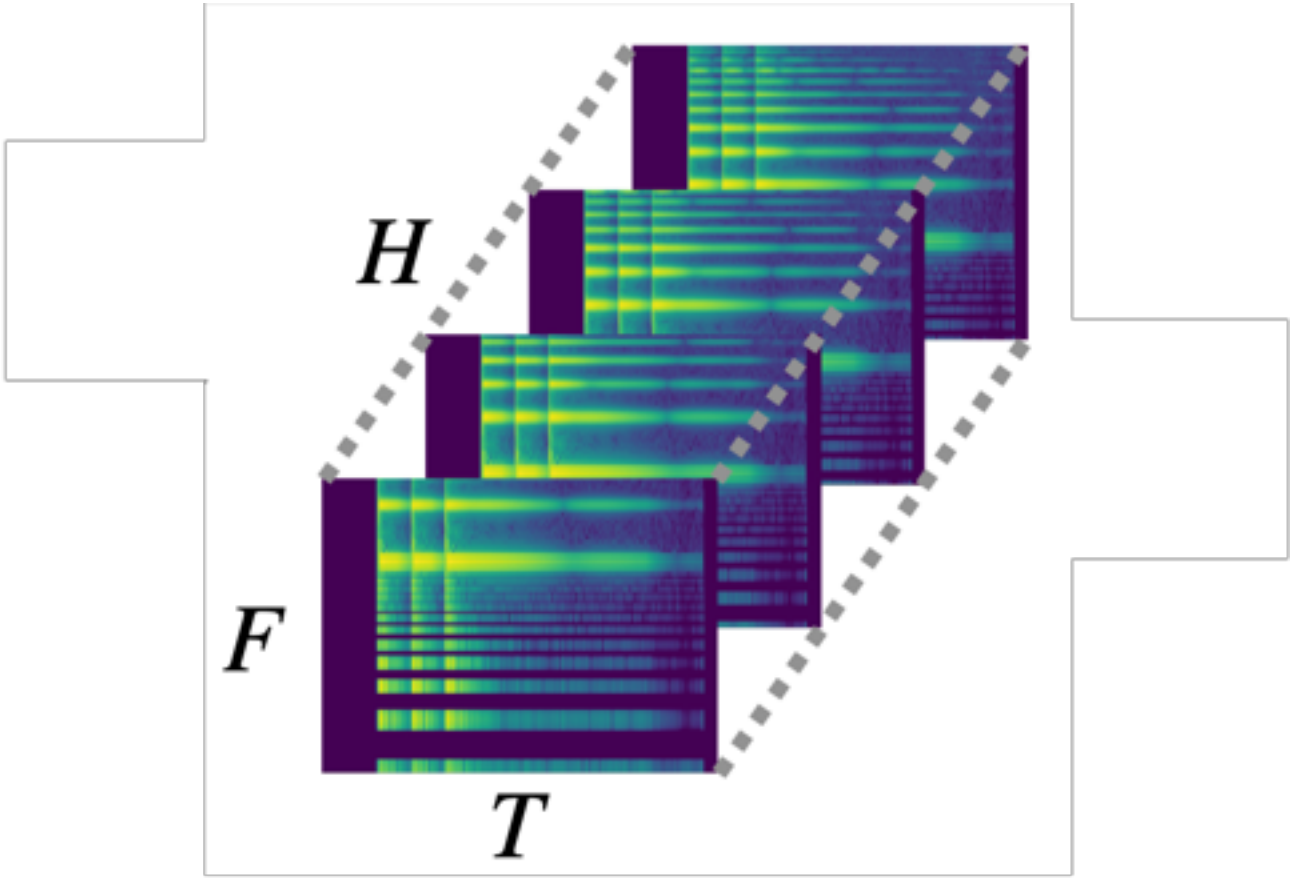


3rd harmonic of Fund freq

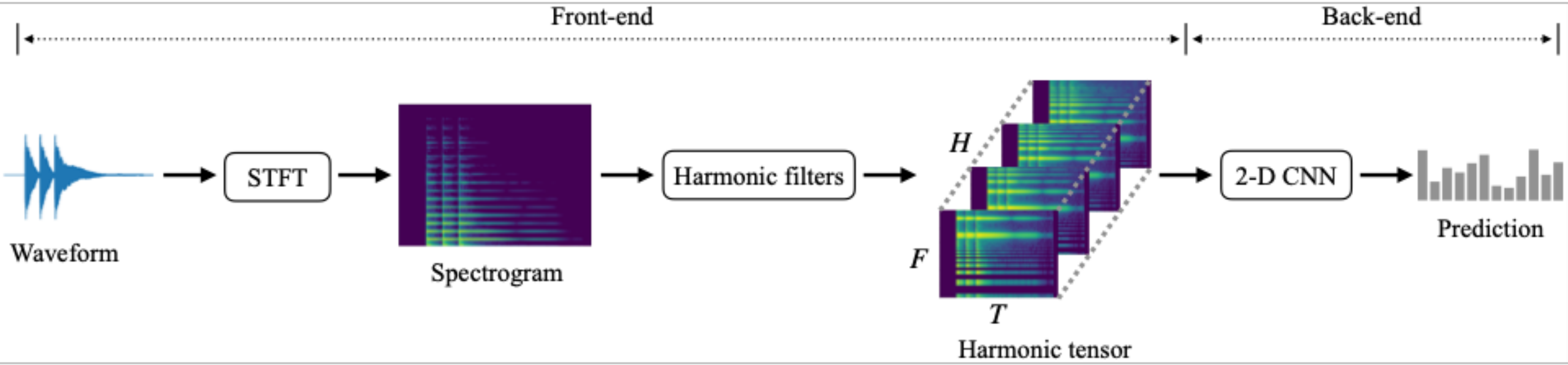


4th harmonic of Fund freq

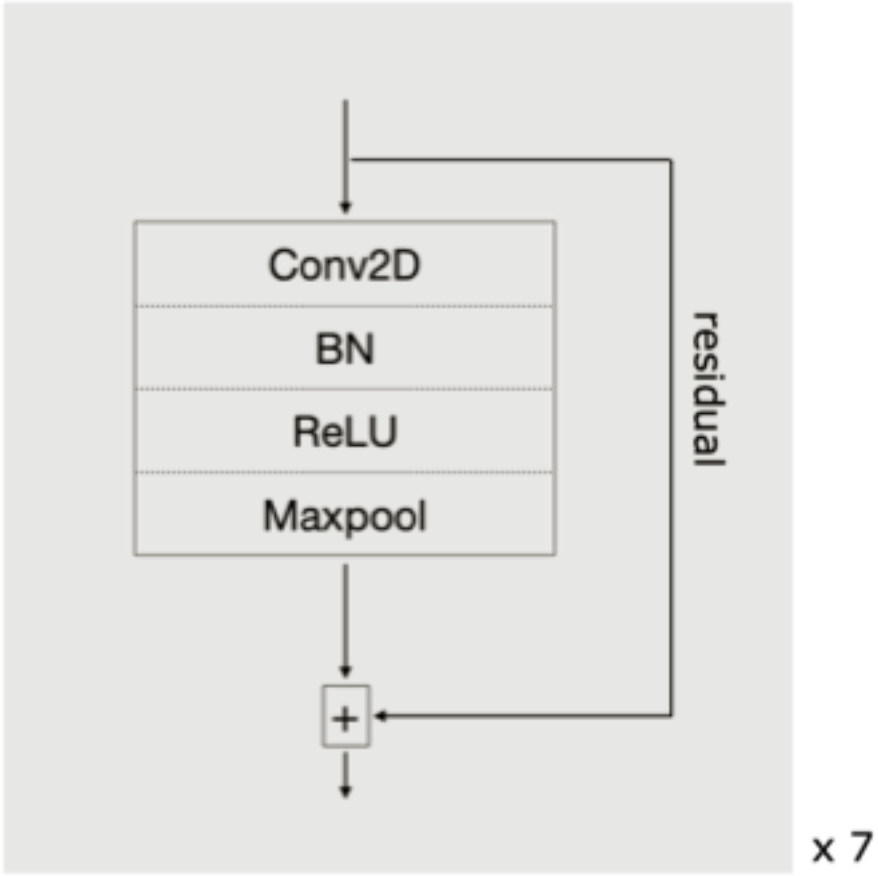
Harmonic tensors



Harmonic CNN



Back-end

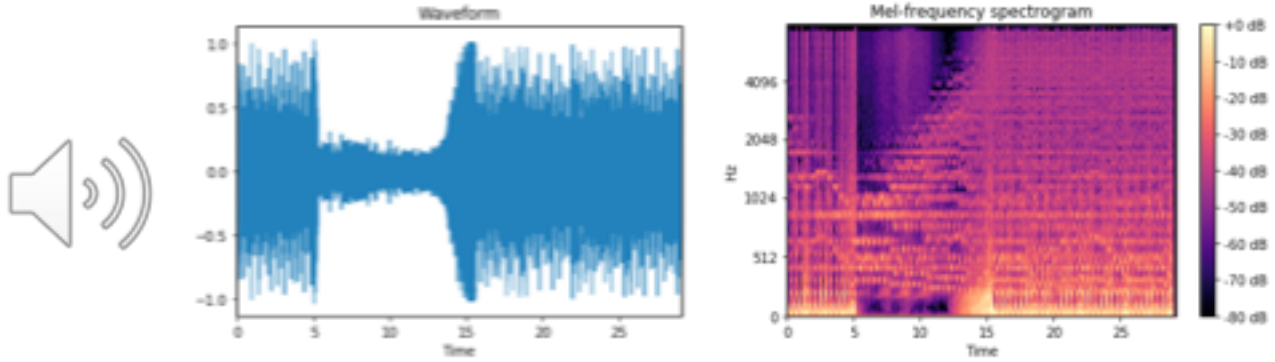


Experiments

	Music tagging	**Keyword spotting**	**Sound event detection**
# data	21k audio clips	106k audio clips	53k audio clips
# classes	50	35	17
Task	Multi-labeled	Single-labeled	Multi-labeled

Experiments

	Music tagging	**Keyword spotting**	**Sound event detection**
# data	21k audio clips	106k audio clips	53k audio clips
# classes	50	35	17
Task	Multi-labeled	Single-labeled	Multi-labeled

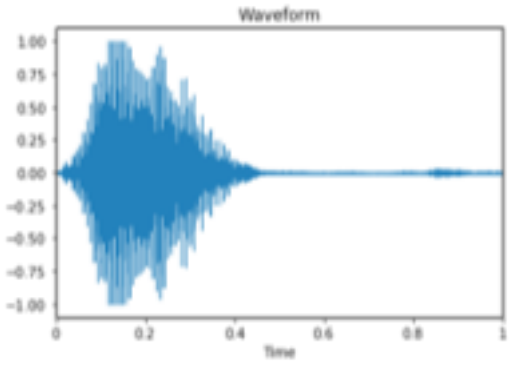


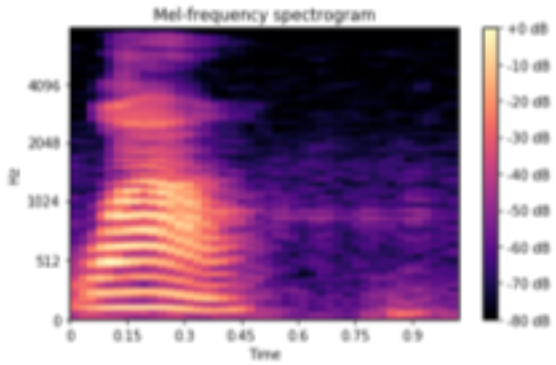
“techno”, “beat”, “no voice”, “fast”, “dance”, …

Many tags are highly related to harmonic structure, e.g., timbre, genre, instruments, mood, …

Experiments

	Music tagging	**Keyword spotting**	Sound event detection
# data	21k audio clips	106k audio clips	53k audio clips
# classes	50	35	17
Task	Multi-labeled	Single-labeled	Multi-labeled





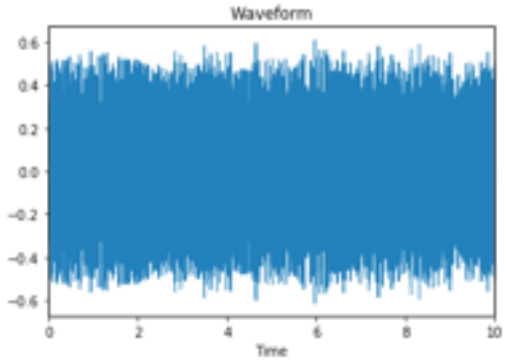
“wow”

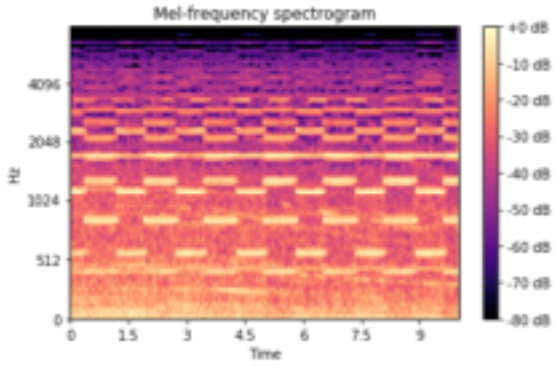
(other labels: “yes”, “no”, “one”, “four”, …)

Harmonic characteristic is well-known important feature for speech recognition (MFCC)

Experiments

	Music tagging	Keyword spotting	**Sound event detection**
# data	21k audio clips	106k audio clips	53k audio clips
# classes	50	35	17
Task	Multi-labeled	Single-labeled	Multi-labeled





"Ambulance (siren)", "Civil defense siren"

(other labels: "train horn", "Car", "Fire truck"…)

Non-music and non-verbal audio signals are expected to have "inharmonic" features

Experiments

Filters	Front-end	back-end
Mel-spectrogram	CNN	CNN
Mel-spectrogram	CNN	Attention RNN
Linear / Mel spec, MFCC	Gated-CRNN	
Fully-learnable	CNN	CNN
Partially-learnable	CNN	CNN

Methods	Music Tagging		Keyword Spotting	Sound Event Tagging	
	MTAT		Speech Commands	DCASE 2017	
	ROC-AUC	PR-AUC	Accuracy	F1 (0.1)	F1 (opt)
Musicnn [5]	0.9089*	0.4503*	-	-	-
Attention RNN [18]	-	-	0.9390	-	-
Surrey-cvssp [19]	-	-	-	-	0.5560
Sample-level [2]	0.9054	0.4422	0.9253	0.4213	-
+ SE [20]	0.9083	0.4500	0.9395	0.4582	-
+ Res +SE [20]	0.9075	0.4473	0.9482	0.4607	-
Proposed	**0.9141**	**0.4646**	**0.9639**	**0.5468**	**0.5824**

Effect of harmonic

H	1	2	3	4	5	6	7*
ROC-AUC	0.9132	0.9115	0.9118	0.9118	0.9129	0.9141	**0.9146**
PR-AUC	0.4599	0.4541	0.4550	0.4555	0.4562	**0.4646**	0.4617

Harmonic CNN is efficient and effective architecture for music representation

Methods	MTAT		MSD		MTG-Jamendo	
	ROC-AUC	PR-AUC	ROC-AUC	PR-AUC	ROC-AUC	PR-AUC
FCN [1]	0.9005	0.4295	0.8744	0.2970	0.8255	0.2801
FCN (with 128 Mel bins)	0.8994	0.4236	-	-	-	-
Musicnn [2]	0.9106	0.4493	0.8803	0.2983	0.8226	0.2713
Musicnn (with 128 Mel bins)	0.9092	0.4546	-	-	-	-
Sample-level [3]	0.9058	0.4422	0.8789	0.2959	0.8208	0.2742
Sample-level + SE [4]	0.9103	0.4520	0.8838	0.3109	0.8233	0.2784
CRNN [6]	0.8722	0.3625	0.8499	0.2469	0.7978	0.2358
CRNN (with 128 Mel bins)	0.8703	0.3601	-	-	-	-
Self-attention [7]	0.9077	0.4445	0.8810	0.3103	0.8261	0.2883
Harmonic CNN [9]	0.9127	0.4611	**0.8898**	**0.3298**	0.8322	0.2956

All models can be reproduced by the following repository
https://github.com/minzwon/sota-music-tagging-models

Harmonic CNN is more generalizable to realistic noises than other methods

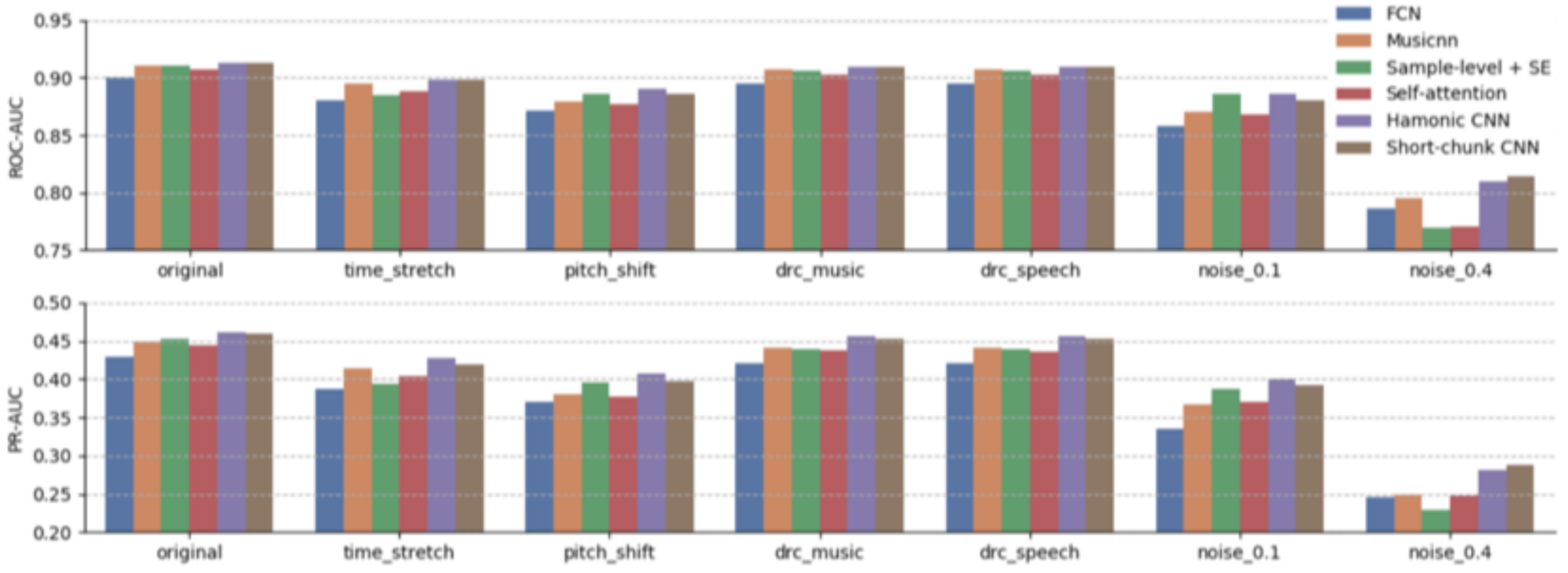


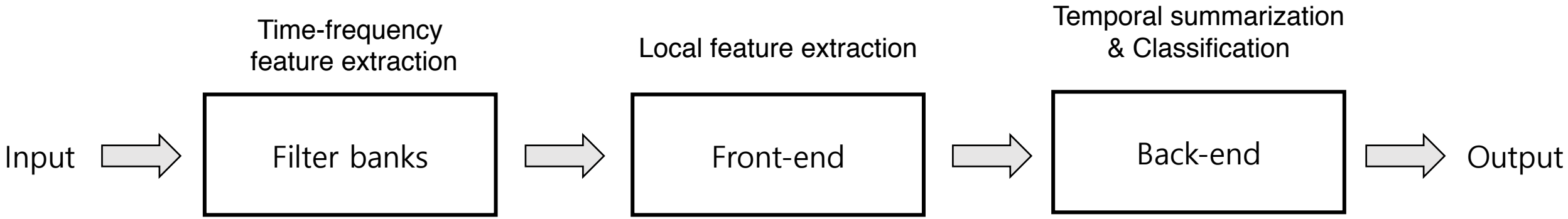
Figure 2: Evaluations metrics with perturbed audio inputs. Dynamic range compression is shortened as "drc" in the plot.

Interpretable back-end with self-attention mechanism

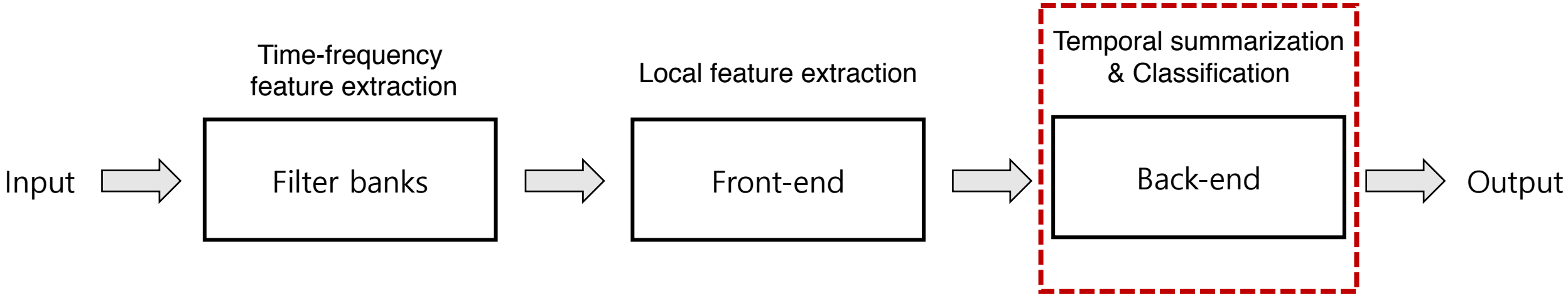
[ICML 2019 Workshop] Visualizing and Understanding Self-attention based Music Tagging.

[ArXiv 2019] Toward Interpretable Music Tagging with Self-attention.

Recall: Front-end and back-end framework

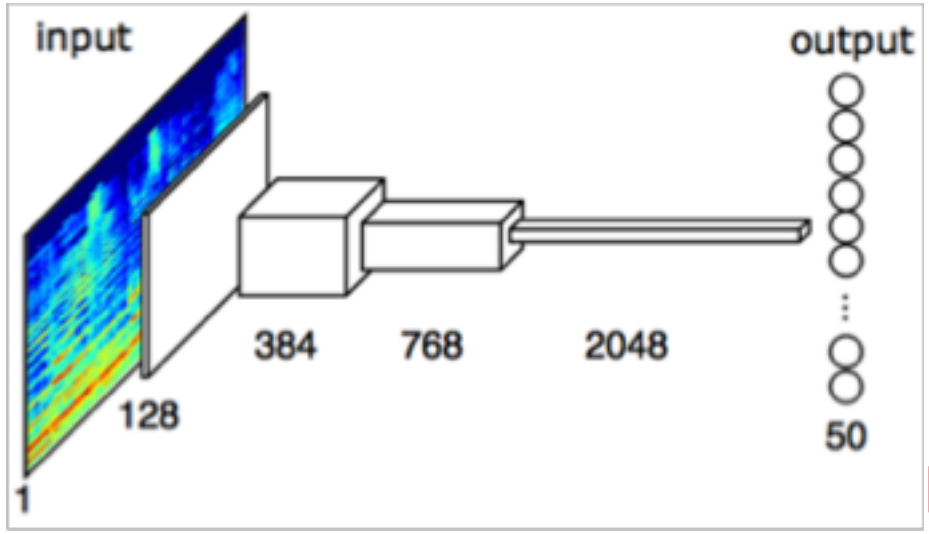


Recall: Front-end and back-end framework



What's happening in the back-end?

CNN-based back-end cannot capture “temporal property”.

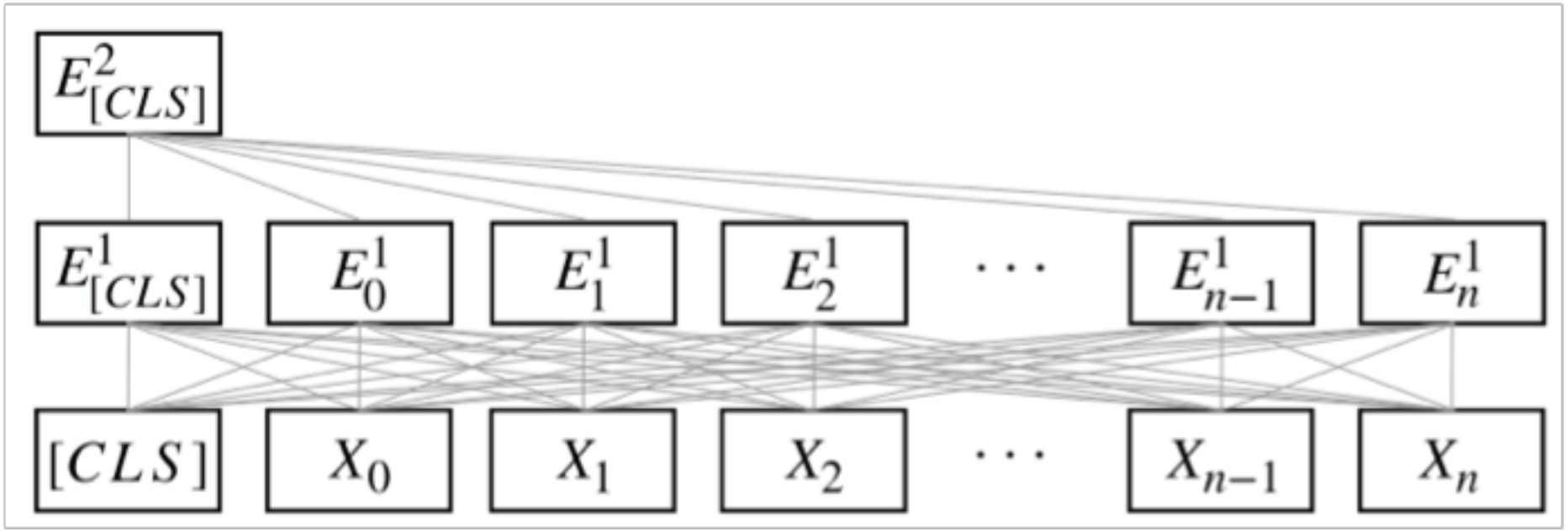


Loose locality

FCN-4
Mel-spectrogram *(input: 96×1366×1)*
Conv 3×3×128
MP (2, 4) *(output: 48×341×128)*
Conv 3×3×384
MP (4, 5) *(output: 24×85×384)*
Conv 3×3×768
MP (3, 8) *(output: 12×21×768)*
Conv 3×3×2048
MP (4, 8) *(output: 1×1×2048)*
Output 50×1 (sigmoid)

[ISMIR 2016] "Automatic tagging using deep convolutional neural networks.”, Keunwoo Choi, George Fazekas, Mark Sandler

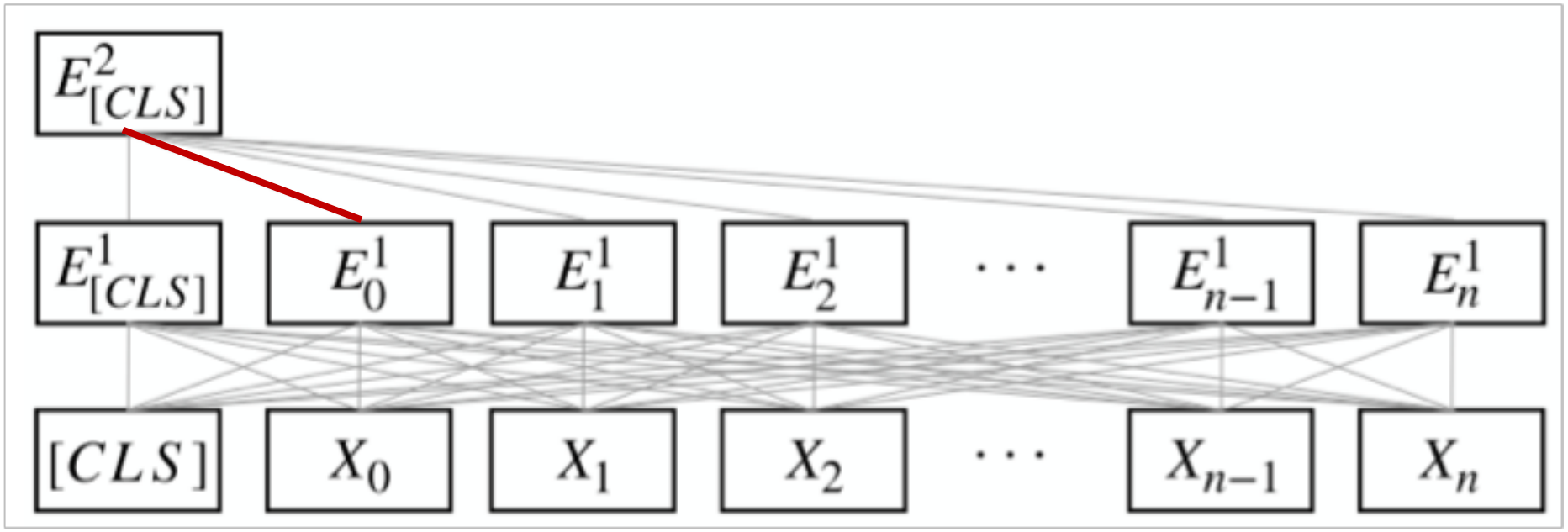
Self-attention back-end
to capture long-term relationship



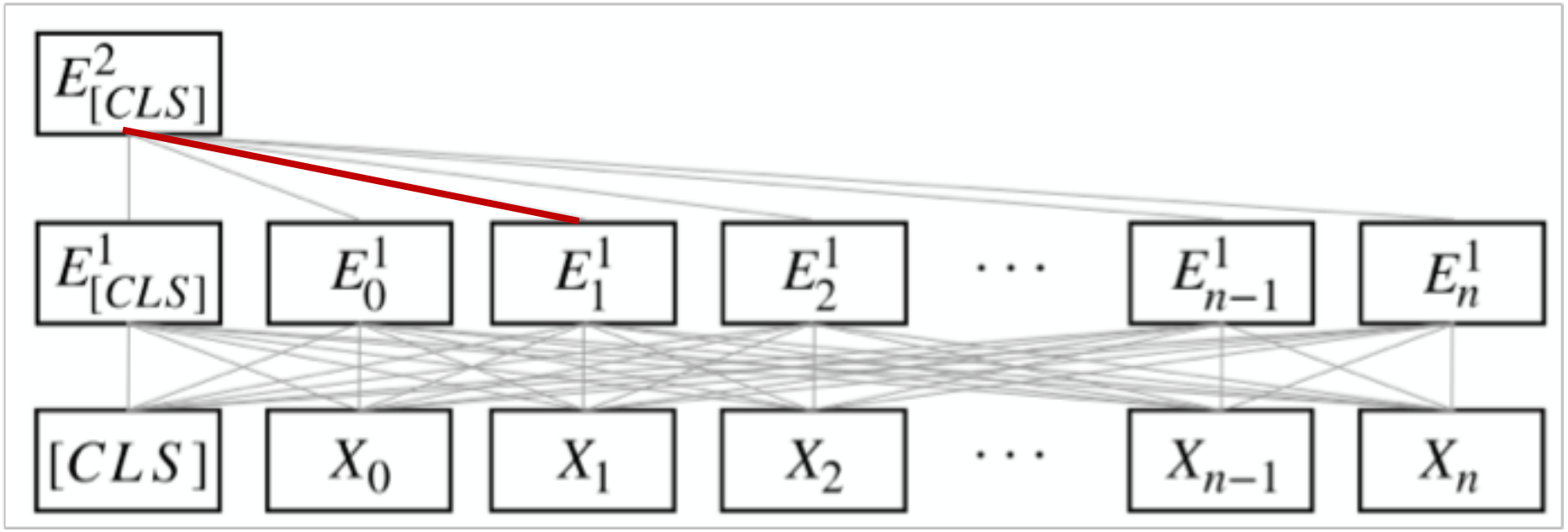
Experiments (music tagging)

		MTAT		MSD	
Front-end	Back-end	AUROC	AUPR	AUROC	AUPR
Raw [20]	CNN_L [20]	90.62	44.20	88.42*	-
Raw [20]	*Att* (Ours)	90.66	44.21	88.07	29.90
Spec [28]	CNN_P [28]	90.89	45.03	88.75*	31.24*
Spec [28]	*Att* (Ours)	90.80	44.39	88.14	30.47

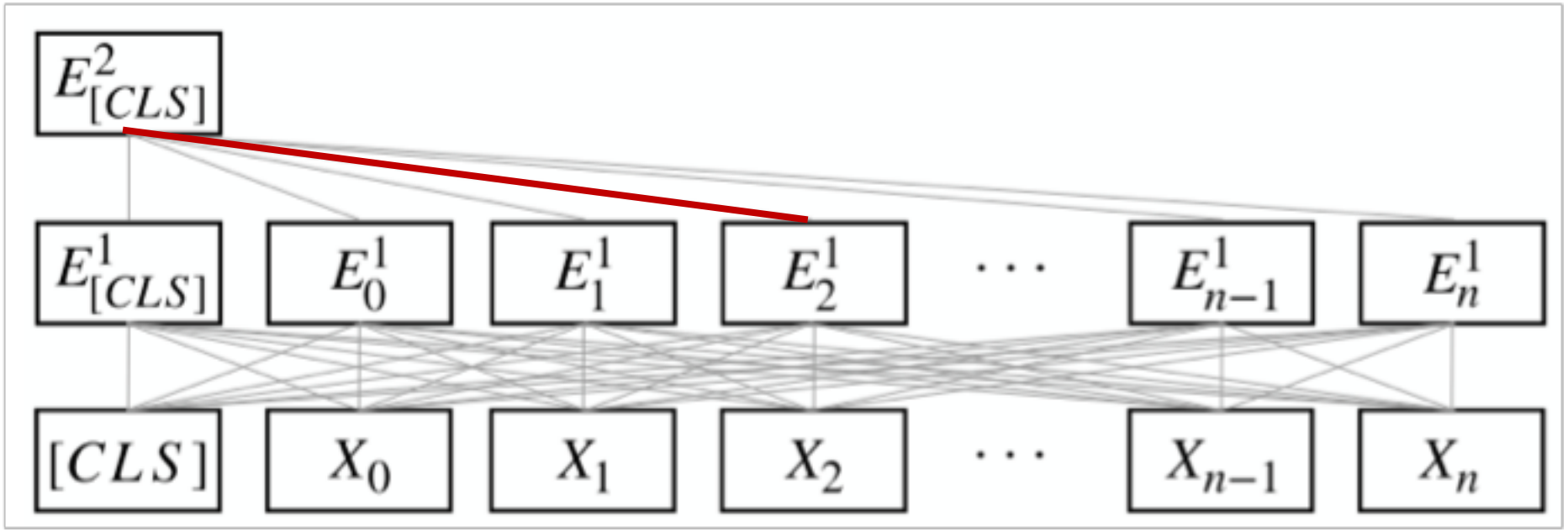
Self-attention back-end for better interpretability



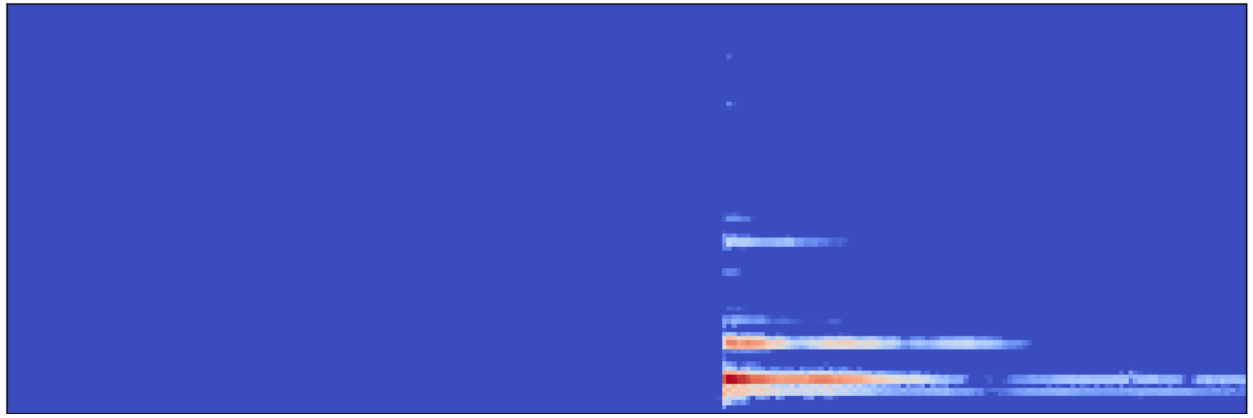
Self-attention back-end for better interpretability

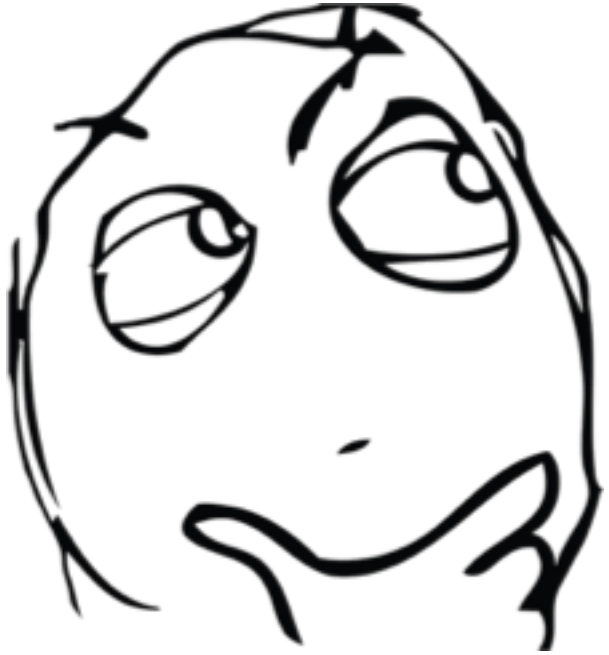


Self-attention back-end for better interpretability



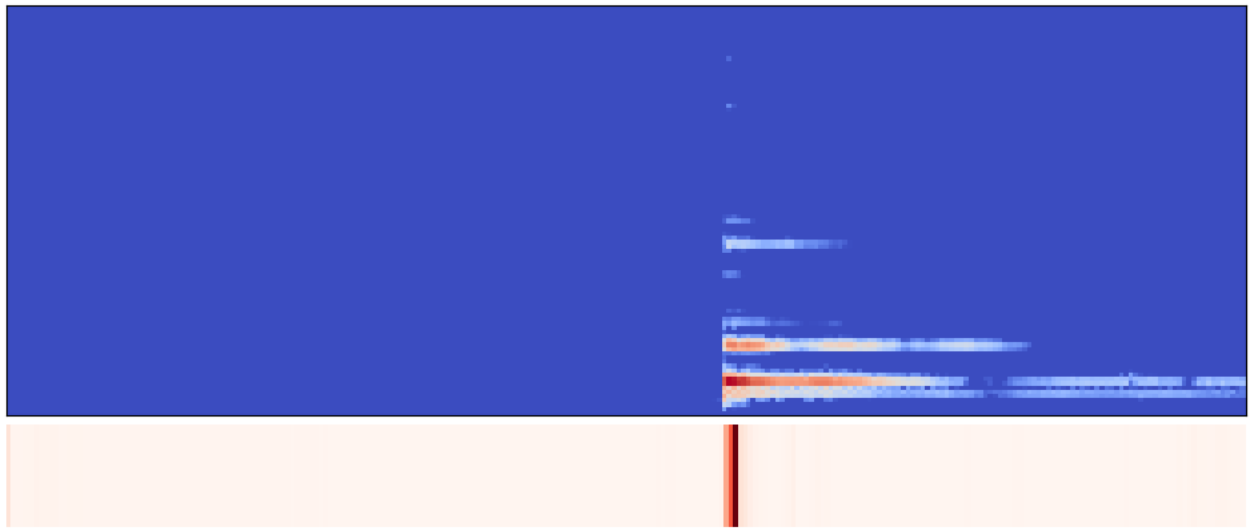
Attention score visualization.



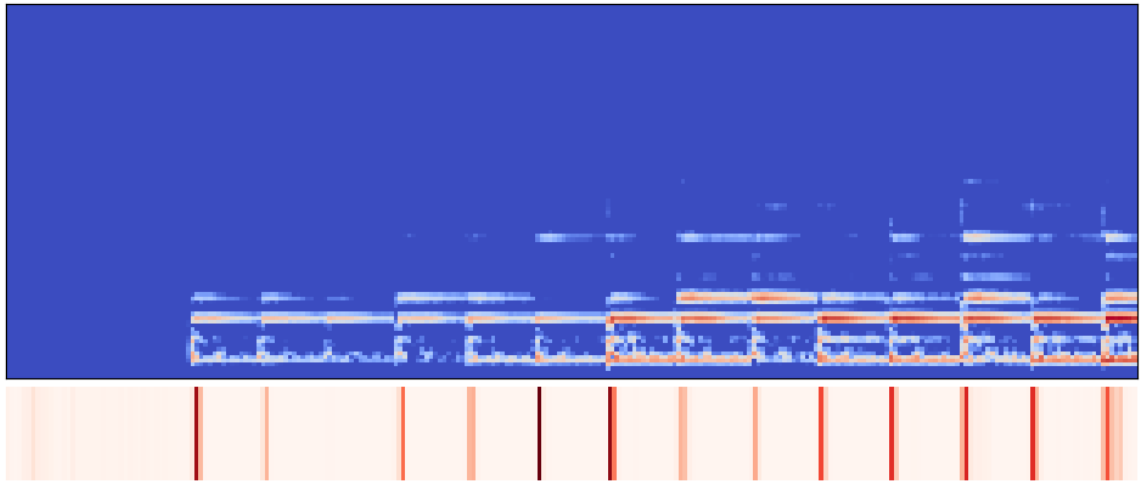


My network says it has a “quiet” tag.
But why?

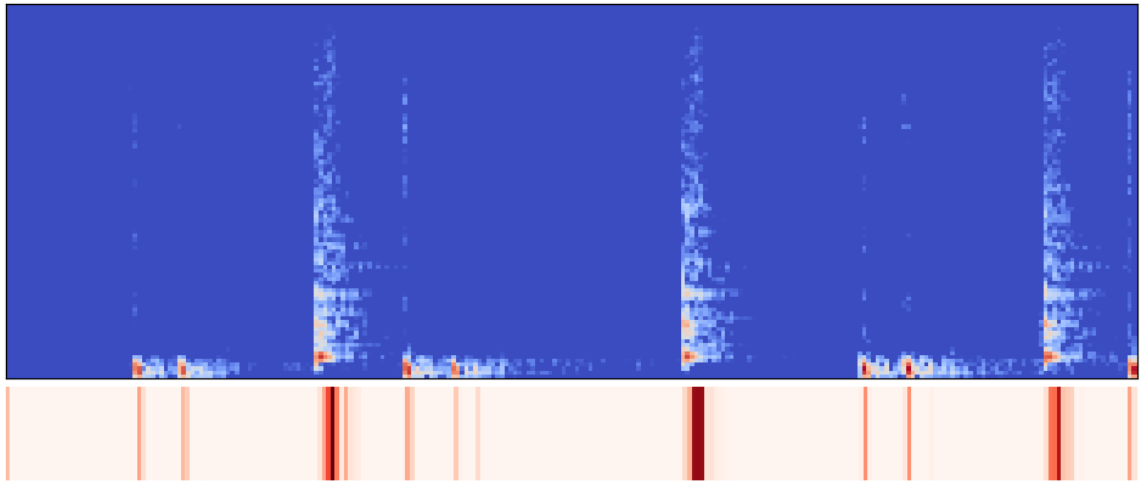
Attention score visualization.



Back-end as “sound” detector: Positive tags case

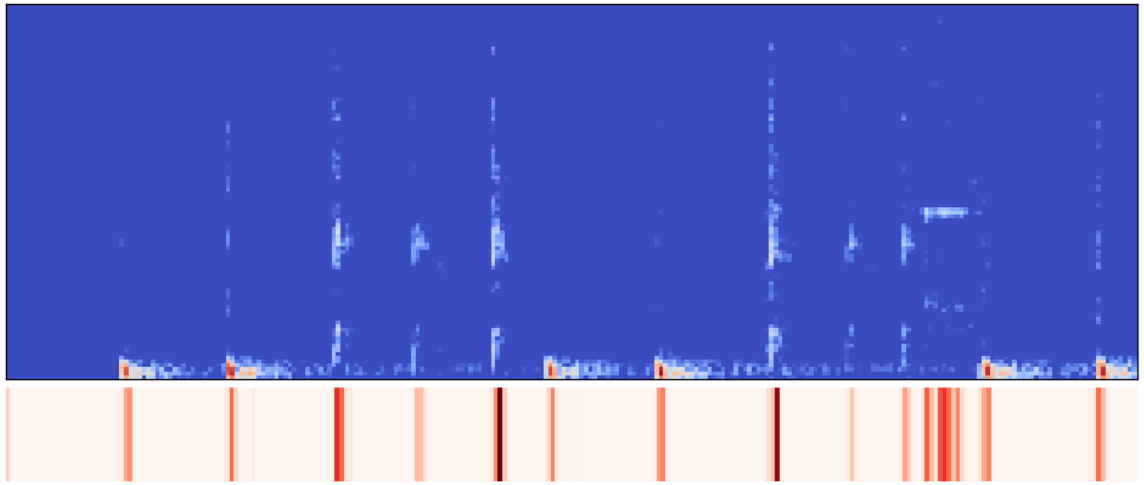


“Piano”

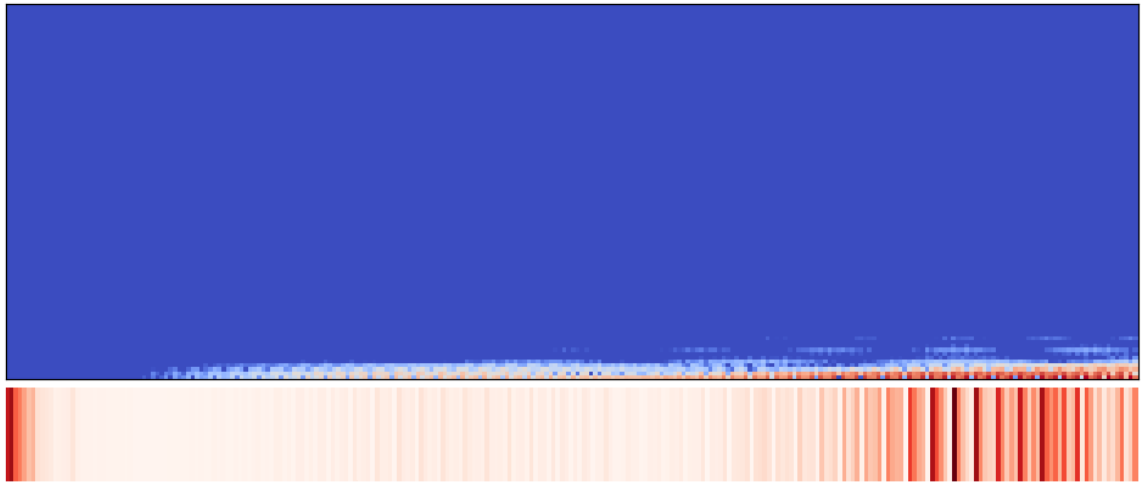


“Drum”

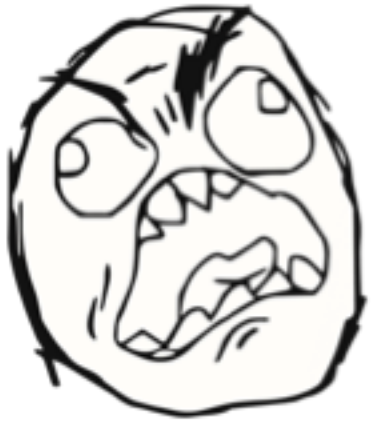
Back-end as “sound” detector: Negative tags case



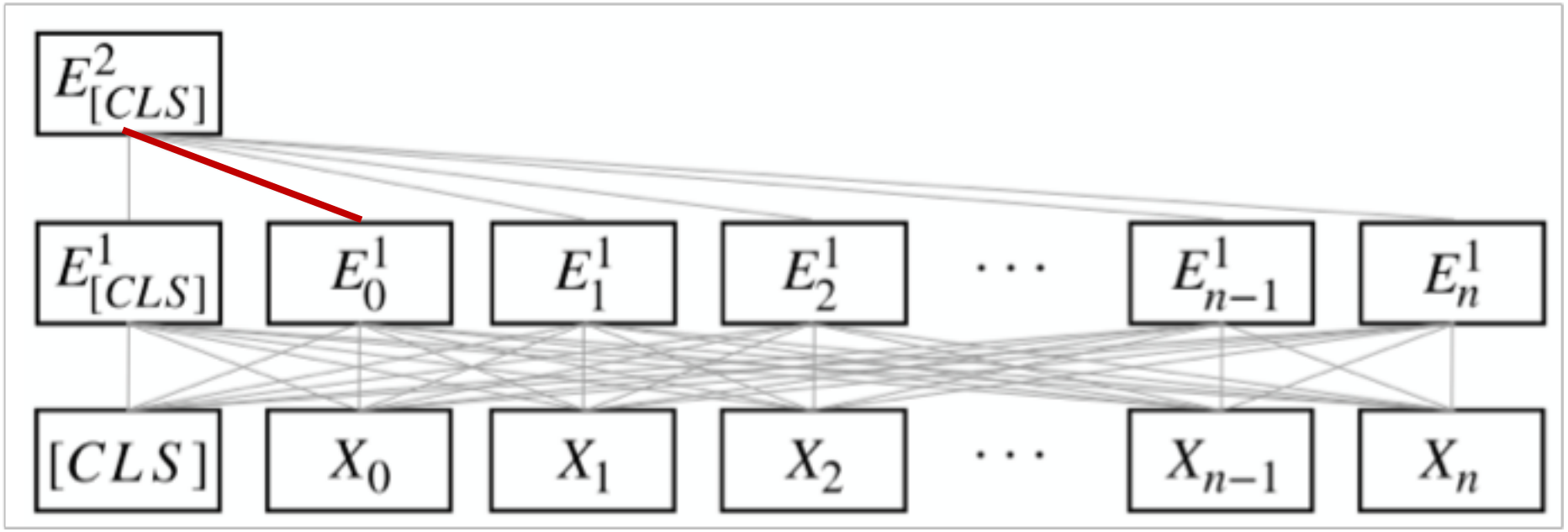
“No vocal”



“Quiet”

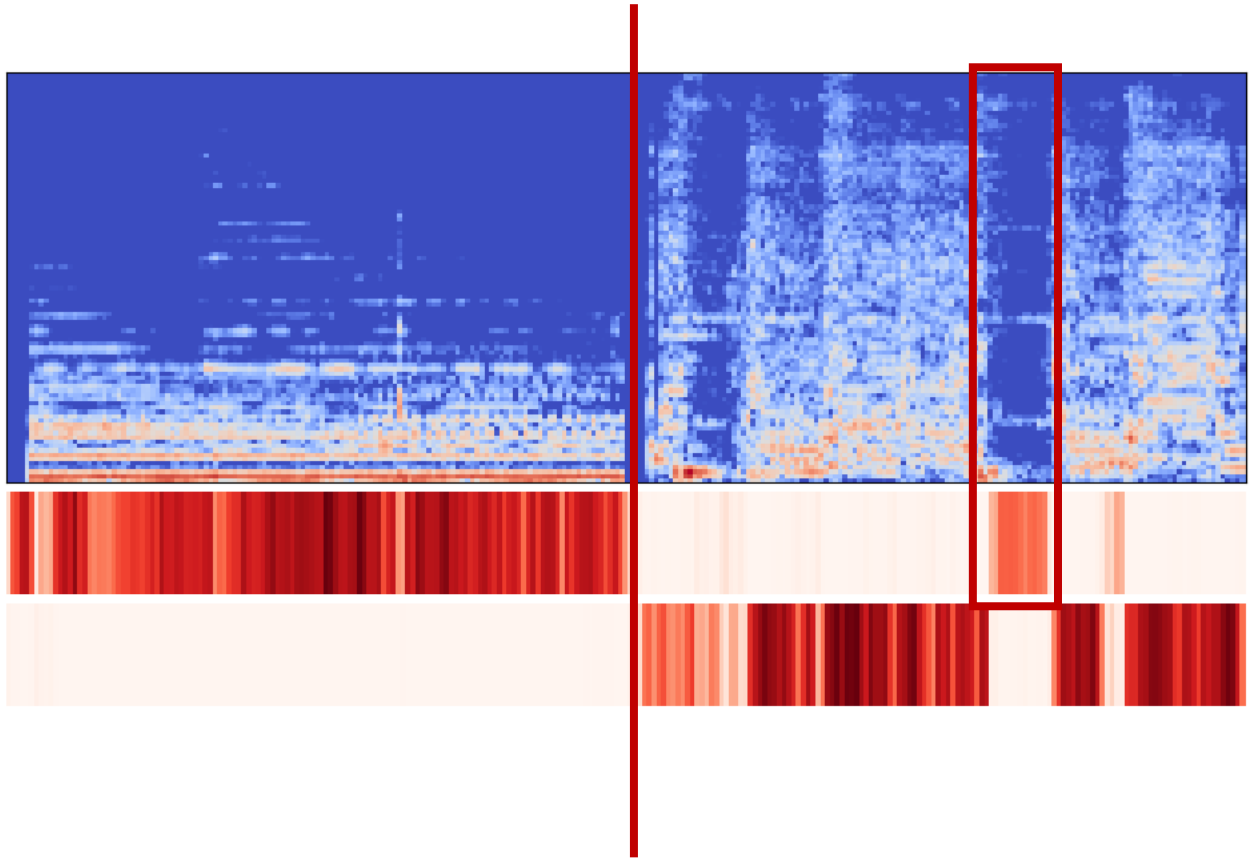


Tag-wise heatmap contribution

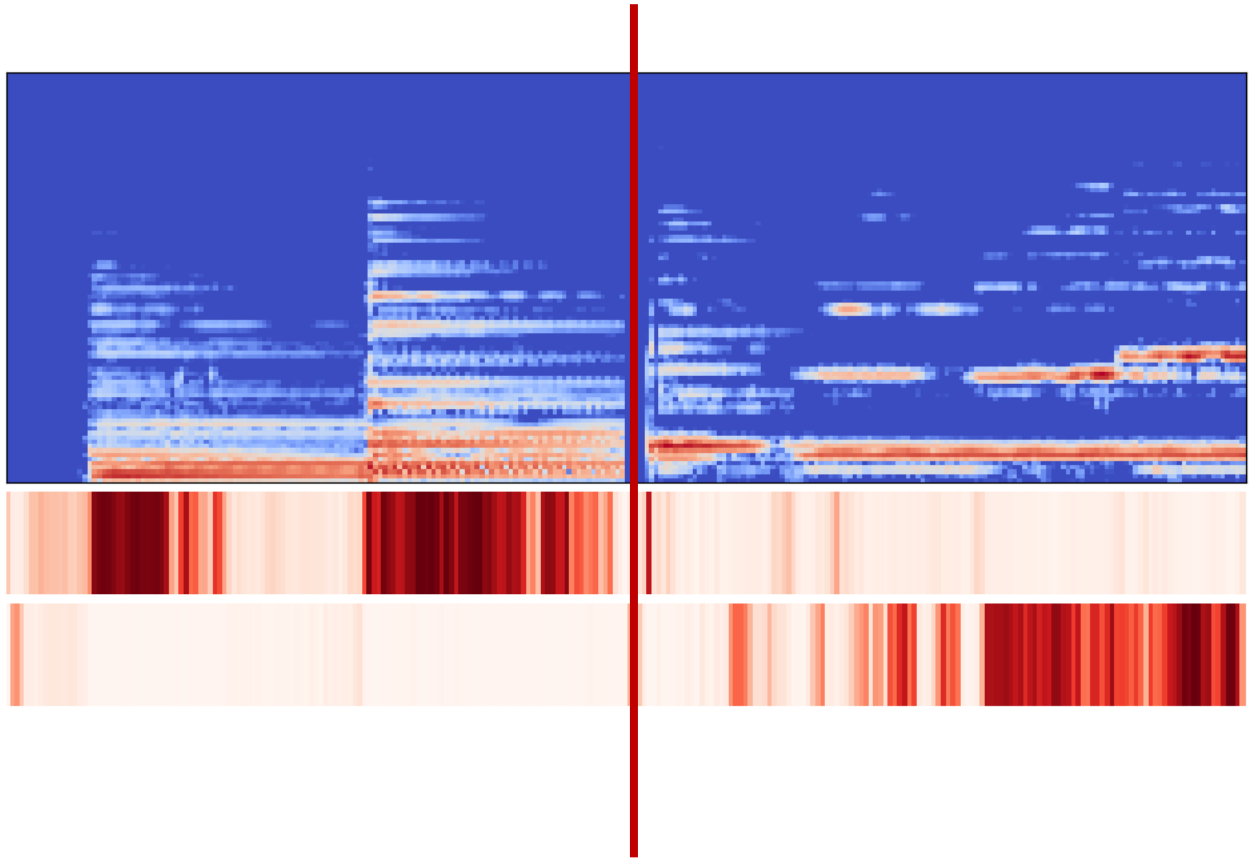


Set the selected attention weight as 1, while others are set to 0

Tag-wise heatmap contribution



Tag-wise heatmap contribution



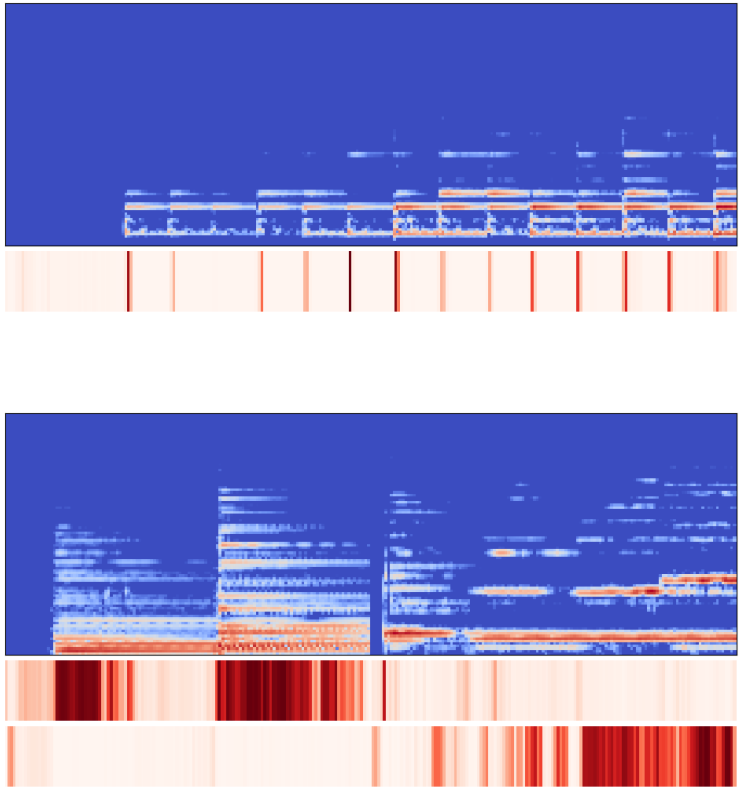
Confidence of “Piano”

Confidence of “Flute”

Conclusion

Powerful front-end by Harmonic CNN
Interpretable back-end by self-attention

Methods	Music Tagging		Keyword Spotting	Sound Event Tagging	
	MTAT		Speech Commands	DCASE 2017	
	ROC-AUC	PR-AUC	Accuracy	F1 (0.1)	F1 (opt)
Musicnn [5]	0.9089*	0.4503*	-	-	-
Attention RNN [18]	-	-	0.9390	-	-
Surrey-cvssp [19]	-	-	-	-	0.5560
Sample-level [2]	0.9054	0.4422	0.9253	0.4213	-
+ SE [20]	0.9083	0.4500	0.9395	0.4582	-
+ Res +SE [20]	0.9075	0.4473	0.9482	0.4607	-
Proposed	**0.9141**	**0.4646**	**0.9639**	**0.5468**	**0.5824**



Reference

- [ISMIR 2019 Late Break Demo] Automatic Music Tagging with Harmonic CNN,

  Minz Won, **Sanghyuk Chun**, Oriol Nieto, Xavier Serra

- [ICASSP 2020] Data-driven Harmonic Filters for Audio Representation Learning.

  Minz Won, **Sanghyuk Chun**, Oriol Nieto, Xavier Serra

- [SMC 2020] Evaluation of CNN-based Automatic Music Tagging Models.

  Minz Won, Andres Ferraro Dmitry Bogdanov, Xavier Serra

- [ICML 2019 Workshop] Visualizing and Understanding Self-attention based Music Tagging.

  Minz Won, **Sanghyuk Chun**, Xavier Serra

- [ArXiv 2019] Toward Interpretable Music Tagging with Self-attention.

  Minz Won, **Sanghyuk Chun**, Xavier Serra