Codec independent lossy audio compression detection

Romain Hennequin · Jimena Royo-Letelier · Manuel Moussallam

Deezer, 10-12 rue d'Athènes, 75009 Paris, FRANCE — research@deezer.com

Summary

- **Method for detecting marks of lossy compression encoding, (MP3, AAC . . .), from PCM audio without frame-alignment**
- **Based on a convolutional neural network applied to audio spectrograms**
- **Trained with various lossy audio codecs and bitrates**
- **High performances on a large database**
- **Robustness to codec type and resampling**

Perceptual codecs

Standard approach shared by many codecs: filterbank output is quantized using a psychoacoustic model.

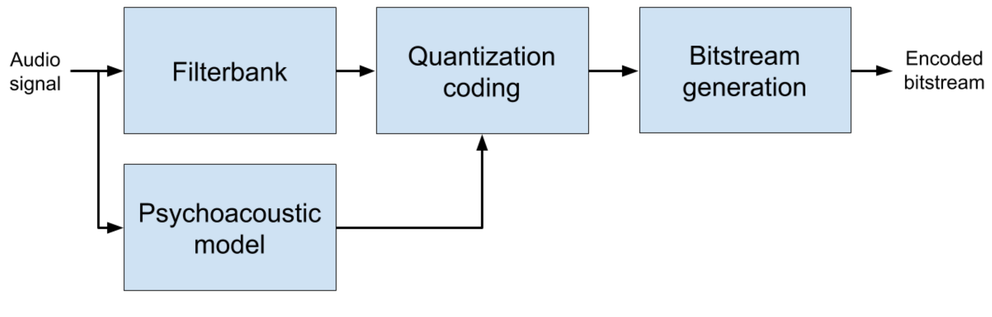


Figure 1: Typical lossy perceptual codec pipeline. The quantization step is the only **lossy** operation.

Generates common *visible* artifacts on audio spectrograms:

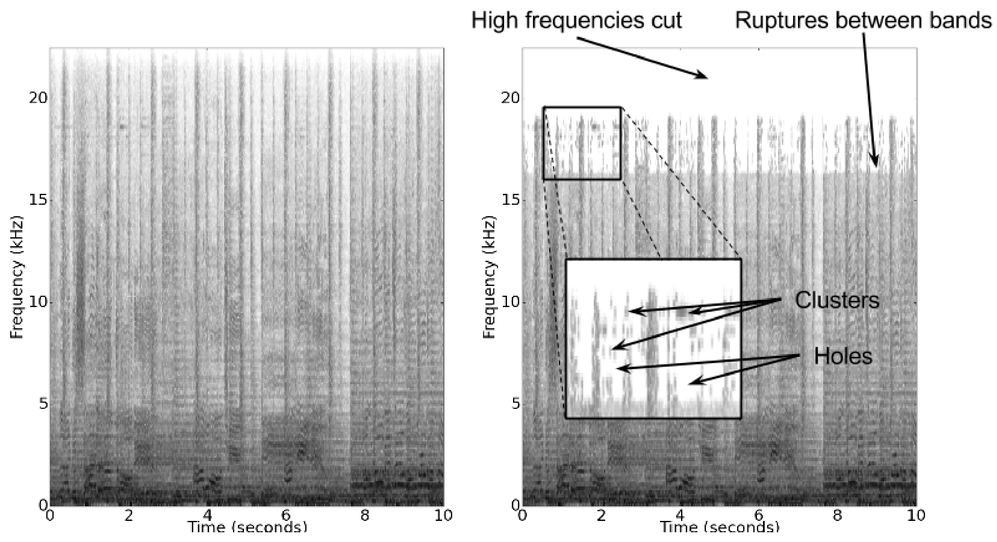


Figure 2: Log-Magnitude STFT of 10 seconds of audio: Left: Original, unaltered file. Right: After lossy compression. Quantization artifacts are easily observed in high frequencies.

Database

- Need for a large database of Unaltered (unprocessed) and Altered (processed by a lossy perceptual coder) audio files.
- Altered files are easily obtained by encoding Unaltered ones.
- Unaltered files are trickier to obtain since there is no guarantee of unalteration.

Active Learning like approach:

- Select a large (about 30000) amount of flac files among the millions that have been delivered to Deezer.
- **Assumption:** most of them are unaltered files.
- Generation of altered files using various codecs and various bitrates.
- Train a classifier (same as presented after) to discriminate unaltered from altered files.
- **Manually Check** unaltered files classified as altered. Remove those that seems to be altered.
- Iterate multiple times until confidence in the unaltered files being truly unaltered is high enough. Leaving approx. 28.6K files in the dataset.

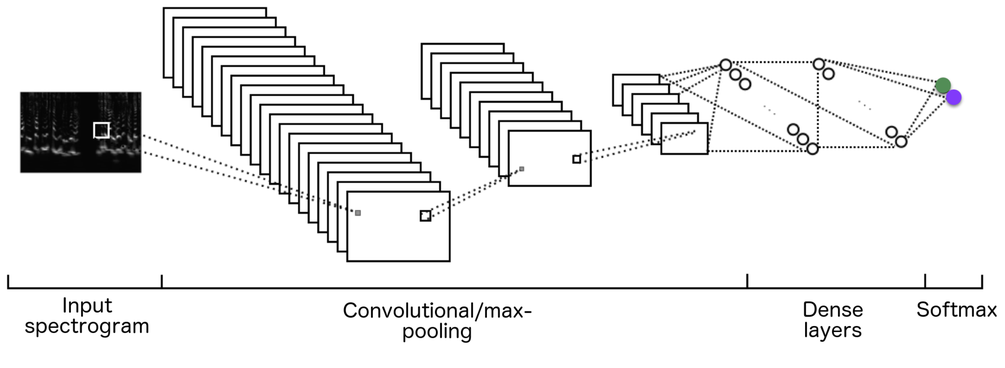
Remark: this method may remove files that exhibit content similar to lossy codec artefact but are actually unaltered.

Altered Files:

- Codecs: Advanced Audio Coding (AAC), MPEG1audio layer 3 (MP3), Vorbis, WindowsMedia Audio 7 (WMAV1), Windows Media Audio 8 (WMAV2), MPEG1 audio layer II (MP2) and Dolby AC3 (AC3).
- Bitrates from 32kbps to 320kbps.

Classification

- CNN (see [2]) consisting of **4 convolutional and max-pooling layers** followed by **2 fully-connected layers** and a logistic regression at top. Built with Theano Python library [1]. Also already used for codec analysis in [3].
- As opposed to most papers in the litterature, features are **not frame-synchronized**: we use raw Log-Magnitude Short Time-Fourier Transform:
  - mostly encompasses small time offset in the phase component, which is discarded.
  - allows fast computation while still revealing compression artifacts.



Results

Confusion matrix

	Cld Altered	Cld Unaltered
Altered	98.1%	1.9%
Unaltered	0.9%	99.1%

Codecs robustness experiment:

Removed Vorbis from training set

Codec	Bitrate	Detection
flac		99.3%
ac3	192k	99.3%
mp3	128k	99.1%
mp3	32k	99.1%
wmav1	32k	99.1%
mp3	192k	99.0%
wmav1	192k	99.0%
vorbis	6	98.3%
mp3	320k	96.2%
aac	256k	95.3%
aac	320k	0.0%

Detection rate for each codec/bitrate

Codec	Bitrate	Detection	Codec	Bitrate	Detection
ac3	192k	99.3%	mp3	192k	99.0%
mp2	192k	99.2%	wmav1	192k	99.0%
vorbis	6	99.1%	mp3	320k	98.1%
wmav1	32k	99.1%	aac	256k	94.3%
mp3	32k	99.1%	aac	320k	2.3%
flac		99.1%			

Table 1: Codec-specific performance. Codec/bitrate combination not shown have 100% detection rate. AAC at 320k is the only problematic case.

Sampling rate robustness experiment

Codec	Bitrate	Detection	Codec	Bitrate	Detection
ac3	192k	99.3%	aac	192k	98.4%
wmav2	64k	99.2%	flac		98.4%
mp2	320k	99.2%	wmav1	256k	98.2%
wmav2	320k	99.1%	mp3	320k	98.1%
wmav1	320k	99.0%	wmav1	192k	96.9%
mp3	256k	98.7%	aac	256k	95.8%
mp3	192k	98.5%	aac	320k	32.4%

Table 2: Codec-specific performance after training database is enriched with resampled files.

Conclusion

- **CNN-based method to detect audio that has been compressed using a perceptual codec from PCM material.**
- **State-of-the-art method 98.6% (however on multiple codecs).**
- **Robust to unknown perceptual codecs and sampling rate changes.**

Future works:

- **Study with non-generic codecs (speech codecs) and more modern generic codecs (MP3PRO).**
- **Robustness to additive noise artifact masking.**

References

[1] James Bergstra, Olivier Breuleux, Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, Guillaume Desjardins, Joseph Turian, David Warde-Farley, and Yoshua Bengio. Theano: A cpu and gpu math compiler in python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 3 – 10, 2010.

[2] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[3] D. Seichter and L. Cuccovillo and P. Aichroth AAC encoding detection and bitrate estimation using a convolutional neural network In *proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* , 2016.