CODEC INDEPENDENT LOSSY AUDIO COMPRESSION DETECTION

Romain Hennequin *Jimena Royo-Letelier* *Manuel Moussallam*

Deezer, 12 rue d'Athènes, 75009 Paris, France

research@deezer.com

ABSTRACT

In this paper, we propose a method for detecting marks of lossy compression encoding, such as MP3 or AAC, from PCM audio. The method is based on a convolutional neural network (CNN) applied to audio spectrograms and trained with the output of various lossy audio codecs and bitrates. Our method shows good performances on a large database and robustness to codec type and resampling.

Index Terms— Audio encoding detection, audio quality detection, Convolutional Neural Network

1. INTRODUCTION

Generic perceptual audio codecs, such as MP3 [1], AAC [2], Vorbis [3], WMA, AC3 and others, are a popular tool with which to reduce audio files size and make the sharing of these files easy. These types of codec have been widely used since their creation in the 90s. This popularity has led to misuse of the codecs, such as reencoding audio from a low bitrate to a higher bitrate or back to Pulse Code Modulation (PCM). Reasons for this could be a misconception that it will result in better quality (bitrate being generally strongly associated with quality), or an intention to cheat on the actual quality of the audio content.

This kind of misuse can also be found in content that is provided to musical streaming services by record labels. At Deezer, record labels are asked to provide music files in FLAC (Free Lossless Audio Codec) format. FLAC is a lossless audio codec that slightly reduces audio file size and that is strictly equivalent to PCM in terms of audio content. For various reasons, it is not uncommon that a small number of audio files delivered to Deezer clearly shows artifacts characteristic of lossy codecs: this indicates that the audio content has been compressed with a lossy codec, maybe for more compact archiving, and then decoded back to PCM before being delivered to Deezer.

Several musical streaming services (including Deezer) have specific offers to stream audio in CD quality (PCM 44100Hz, 16bit) and some specialized online music stores charge more for lossless audio than for lossy ones. Ensuring that audio files provided by the labels have not been previously compressed with a lossy coder is, therefore, of particular interest.

This paper proposes a method for detecting whether PCM audio content had been previously encoded in a lossy format. PCM audio that had not been previously encoded with lossy codec will be referred to as *unaltered* audio, and audio that was previously encoded (possibly multiple times) by a lossy codec as *altered* audio.

Generic perceptual audio codecs use psychoacoustic principles to reduce file size by allocating less information in time/frequency areas where the audio content is supposed to be less audible. These codecs produce artifacts that result in characteristic patterns in audio spectrograms such as high frequency cuts, ruptures between frequency bands and the presence of holes or isolated clusters. These spectrogram artifacts are especially visible when spectrogram parameters (window duration, hop size, window type) match those used by the codec, but can also be seen at different resolutions.

The contribution of this work is to propose a codec-independent system that relies only upon detecting characteristic patterns of spectrogram from perceptual compressed audio. This is possible because we use a generic classification system designed to detect patterns in 2D inputs. Moreover, we do not make use of framing detection, a common step in encoding detection systems, that considerably increases their computational cost. The method we propose is therefore fast enough to make scaling to a catalog comprising tens of millions of tracks possible.

The rest of the paper is organized as follows: in Section 2 we review previous works on lossy compressed audio detection, in Section 3 general characteristics of perceptual audio codecs are presented, in Section 4, we describe how the database that we used for training and testing the proposed algorithm was built, in Section 5, we present the system that we used, in Section 6, we report the results that we obtained with the proposed method, and finally, conclusions and perspectives are drawn in Section 7.

2. PREVIOUS WORKS

Several papers have already addressed the problem of double encoding detection or bitrate detection from PCM, but most of them are dedicated to a single type of codec. In [4], a method to estimate the bitrate of decoded MP3Pro audio is proposed. It uses explicitly the structure of the MP3Pro encoder, especially Spectral Band Replication. In [5], double MP3 compression detection is addressed using statistics on the quantized Modified Discrete Cosine Transform (MDCT) coefficients. In [6], MP3 bitrate is estimated with high accuracy using a Support Vector Machine classifier on the 16kHz to 20KHz band of spectrograms. However, this method does not detect unaltered audio, and although it might generalize to other codecs, results were reported for MP3 only. In [7], AAC encoding parameters, such as filterbank and quantization parameters, are estimated from the audio in order to regenerate the coded bitstream. Once again, the algorithm is designed for AAC only. In [8], the same kind of algorithm is proposed for MP3. In [9], bitrate detection and distinction from unaltered PCM is done with quite high accuracy on AAC encoded audio using a convolutional neural network (CNN). Although the CNN is a very generic and robust classifier, the input features were tailored to highlight MDCT coefficient quantization, and the method cannot adapt easily to other codecs. This kind of engineered features approach would result in an exponential increase in possibilities to take into account, and would therefore not be worth considering to deal with multiple codecs with different characteristics.

Detection of traces of lossy compression was also used for tampering detection in [10, 11] for a limited set of codec types.

All these papers use framing detection as a first step. Framing

detection consists of retrieving the original position of the encoding frames in the audio. The knowledge of the framing is a huge prior for detecting bitrates/traces of lossy compression since it allows us to retrieve the quantized coefficients exactly. However, detecting the frame grid is computationally very costly because the filter bank representation must be computed for every sample offset and every parameters of the filter bank. Moreover, framing is codec-dependent and considering all possible framing for a large set of codecs results in a high computation complexity which makes it unsuitable for large scale detection. Finally, it is not robust to a lot of simple transformations such as resampling, small edits or multiple encoding.

In [12], the number of inactive or weakly active coefficients is used to detect double encoded MP3 without addressing the frame synchronization problem. However the paper deals exclusively with low bitrates for MP3 only. In [13], a generic method for bitrate detection is presented for multiple codecs, however, discrimination of unaltered audio from altered audio is done only for low bitrates (64kbps and below). Results for altered/unaltered discrimination are outperformed by those produced by our approach.

As opposed to these specialized approaches, our system is designed to be agnostic to the type of codec used, and to rely only upon common codec characteristics. As any encoder, possibly several, may have altered the final version of the audio, it is important to propose an algorithm that is codec-independent.

To the authors' knowledge, this is the first paper that addresses detecting lossy compression from multiple codecs. Moreover, this is the only paper to include a training database of several thousands of audio files.

3. PERCEPTUAL CODECS CHARACTERISTICS

3.1. Codecs description

Lossy audio codecs usually rely upon a quantization of time-frequency coefficients parametrized by a psychoacoustic model. The time-frequency coefficients are generally computed using the MDCT and are sometimes combined with a polyphase filter bank (for instance MP3 [1]). Lossy compression algorithms usually group MDCT coefficients into non-uniformly spaced frequency bands to operate quantization. The effect of strong quantization is particularly visible in spectrograms: 0-bit quantization results in some frequency bands having all coefficients set to 0. This is very common for low bitrates, for which not encoding the high frequencies at all is quite a good psychoacoustic choice. 1-bit quantization results in *holes* or *isolated non zero coefficients*.

Generic perceptual audio coders are composed of the following parts, as represented in Figure 1:

- A filter bank: the signal is decomposed in frequency bands, using a MDCT and/or a Polyphase Quadrature Filter (PQF).
- Psychoacoustic model: masking thresholds are estimated.
- Quantization: the output coefficients of the filter bank are quantized according to the psychoacoustic model in order to make the quantization noise as inaudible as possible.

The quantized signal is then encoded and compressed with lossless algorithms but this part does not alter the signal further. The filter bank being invertible, the only phase at which the signal is altered is during quantization.

The parameters of the filter bank depend on the codec: MP2, uses a 32 band Polyphase Quadrature Filter. MP3 uses an hybrid filter bank composed of the same PQF as in MP2 followed by a MDCT with 6 or 18 frequency bands depending on the stationary or transient character of the audio. All other codecs used in this paper use a MDCT as the filter bank with varying numbers of frequency bands: AAC uses a MDCT switching between 128 and 1024 frequency bands, Vorbis uses any power of 2 between 32 and 4096, AC3 uses 128 or 256 and WMA uses any power of 2 between 128 and 2048.

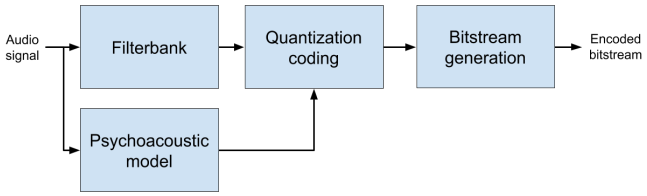

Fig. 1. Generic perceptual encoder scheme.

3.2. Common artifacts of encoded signals

In this section, we review the kind of artifacts produced by lossy audio codecs. Rather than describing these artifacts from a perceptual perspective, we focus on characteristic patterns (see Figure 2) that are frequently encountered in time/frequency representations of encoded audio signals, such as:

1. High frequencies cut (due to quantification with 0 bits in high frequency bands, these bands being considered less informative by the underlying psychoacoustic model of the codec).
2. Ruptures between frequency bands (due to different global scale factors between bands).
3. Holes/isolated clusters in the spectrogram (due to very low bit quantization in some time/frequency areas).

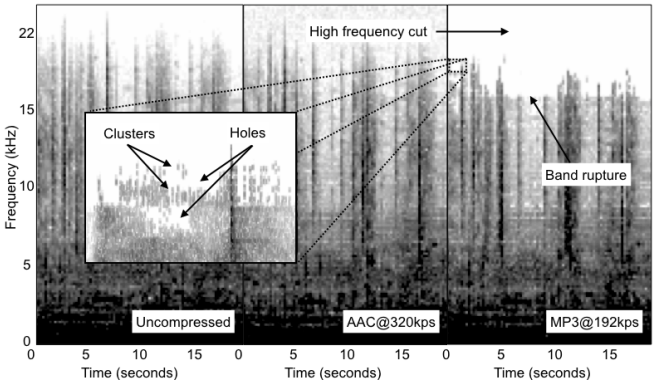

Fig. 2. Codecs artifacts in spectrogram. From left to right: unaltered audio, compression with AAC@320kps and compression with MP3@192kps.

4. DATABASE DESCRIPTION

Supervised classifiers usually require considerable amounts of annotated samples from which to learn their internal parameters. In our case, we need a large database containing uncompressed audio files and altered ones. However, such a database is not easy to gather (even for music streaming companies that have access to millions of files) since, as mentioned earlier, a fair amount of provided PCM audio files cannot be considered as unaltered audio.

To tackle this problem, we create a database by an iterative process of semi-automatic cleaning of uncompressed audio files. In the first step, we gather FLAC files, generate compressed versions of them and train a classification system. Then we remove suspect FLAC files misclassified by the system and iterate by retraining the system with the new, cleaner database. The cleaning thus consists in iterative steps of:

- Training the classifier to discriminate unaltered audio from lossy compressed audio on the current state of the database.
- Manually checking unaltered audio files that were classified as altered by looking for clear evidences of lossy compression marks at the spectrogram of these audio files.
- Removing the files that showed clear evidences of alteration in the previous steps from the unaltered database.

This procedure reduces the number of manual annotations required to a fairly small set, considering the size of the large database needed for training. This first step of cleaning might include a bias since removed files from this first step are probably the most challenging to classify. While aware of this limitation, the satisfactory error rates obtained in different tasks by our system show an effective procedure.

Finally, the generated database is split into training, validation and test databases, and samples from validation and test databases are manually verified (following the same procedure as above) to ensure that they contain only clean, uncompressed audio files.

Our original collection of unaltered files was composed of 30k FLAC files randomly chosen using MD5 hashes. This collection is representative of Deezer catalog: it contains mainly music from varied genres but it may also contains some speech (e.g.: audio books). We remark that a small bias on duplicated songs with almost identical signals but different MD5 could be introduced in this way.

The altered audio files were encoded with AAC [2], MPEG1 audio layer 3 (MP3) –using the LAME encoder–, Vorbis, Windows Media Audio 7 (WMAV1), Windows Media Audio 8 (WMAV2), MPEG1 audio layer II (MP2) and Dolby AC-3 (AC3) codecs. The last two were used because they are the standard codecs for DVD, and thus might be common for decoded audio that come from video. Various bitrates were used (32kbps, 64kbps, 96kbps, 128kbps, 192kbps, 256kbps and 320kbps) for all codecs except for Vorbis that does not support a constant bitrate parameter but only a *quality* parameter. All quality values from 1 to 7 were used for the later codec. Only one encoder was used for each codec. Notice that although there can be differences between encoders, we considered that such variability should be far less important than differences between different codecs.

As expected, this first iteration of the procedure found a quite large number (about 10%) of files that were previously encoded with a lossy codec. We stopped the procedure when the rate of suspect unaltered files fell under 1% (about ten steps). The final database contained 26844 files of uncompressed audio.

All three datasets (training, validation, test) do not contain excerpts coming from the same file. Classes (Altered/Unaltered) were balanced in each dataset.

5. SYSTEM DESCRIPTION

5.1. Convolutional Neural Network

CNNs have been successfully used in several areas, becoming the state-of-the-art method, in particular image classification [14]. CNNs were also already used for codec analysis in [9] with a similar architecture as the one we propose, but using frame-aligned MDCT coefficients as features that were optimized for AAC. As CNNs are very efficient at retrieving patterns in image, it is quite natural to think that they are able to detect the typical artifact patterns described in Section 3.2.

We use a classical network architecture consisting of 4 convolutional layers followed by 3 fully connected layers. Every convolutional layer is followed by a 2×2 max-pooling layer, has rectified linear-unit activation, 16 output feature maps and uses 3×3 filters.

The two first fully connected layers have rectified linear-unit activation and 256 hidden cells. Dropout is used at training in these 2 layers in order to reduce over-fitting. The last fully connected layer has a softmax activation and 2 outputs that model the probability of being in the altered class or in the unaltered class. The class with the maximum probability was chosen as the output of the classifier. Spectrograms as described in Section 5.2 are used as inputs.

The network was trained according to a cross entropy loss function, with mini-batch gradient descent with adagrad optimization using 10-samples batches. Stabilization of validation error was reached after a few epochs (usually about 10 epochs).

5.2. Choice of features

As generic perceptual codecs are usually based on a bank filter generally using a MDCT it is quite clear that a time-frequency representation with linear frequency scale can be a good choice. Although it may be tempting to just use MDCT (or several stacked ones) recall that:

1. As commonly reported in the literature [6, 8, 9, 10], if MDCT is not aligned on the original framing grid of encoding, the quantized coefficients are far less visible.
2. Sample rate of the audio signal may have been changed after decoding, which makes frame synchronization useless.
3. MDCT is usually switching dynamically between several frame sizes, based on the audio content (stationary or transient).
4. Different codecs use different number of frequency bands.

Unlike the MDCT, spectrograms computed from Short Time Fourier Transform (STFT) mostly encompass small time offset in the phase component, which can be discarded. Using the STFT magnitude spectrogram instead of a non-synchronized MDCT allows fast computation while still revealing most compression artifacts. We thus chose it as input features of our classifier.

We found that a window size of 512 samples with 50% overlap and a Hamming window gave the best performance. Results reported in Section 6 were obtained with these values. Spectrogram were computed on 10s excerpts of audio. Spectrograms were preprocessed with a $\log$ (dynamic compression) and then standardized (transformed to 0 mean and unit variance).

6. RESULTS

6.1. Main experiment

We first report results for the main experiment of altered/unaltered classification. We observed a detection rate of 98.6%. The confusion matrix in Table 1 shows that the error is dominated by altered audio examples being classified as unaltered.

In comparison to the 99.1% rate of unaltered audio detected as unaltered by our system, [13] reports a 89.6% rate, and [9] reports a 96.9% rate for unaltered audio detected as unaltered in their bitrate

classification task, the former being limited to low bitrates (64kbps and under) and the latter being restricted to AAC only.

	Classified Altered	Classified Unaltered
Is Altered	98.1%	1.9%
Is Unaltered	0.9%	99.1%

Table 1. Altered/Unaltered audio confusion matrix.

Detection rates are reported for each codec and each bitrate in Table 2. For sake of clarity, detection rates of 100% are not displayed. Errors are very low for bitrate under $192kbps$: all error rates are under 1%. Error rates for bitrates above $192kpbs$ are also under 1% with the exception of AAC at 320kbps and at 256kbps and MP3 at 320kbps. For such high bitrates, artifacts are very weakly perceptible in spectrograms, as can be seen in Figure 2. This is why the CNN fails at detecting typical artifact patterns. This is especially true for AAC at 320kbps where the performance of the classifier collapses to an error rate of approximately 98%. A possible explanation for such a low performance is a failure to manually detect AAC at 320kbps alteration in spectrograms while creating the database (see Section 4).

Codec	Bitrate	Detection rate	Codec	Bitrate	Detection rate
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 2. Detection rate for each codec/bitrate.

It is worth noting that errors are more or less ranked according to their seriousness, low bitrates having very low error rates and higher error rates happening for high bitrates. This is consistent with perceptual measures of quality on compressed audio.

6.2. Robustness analysis

6.2.1. Detection of unknown codecs

As our approach aims to be codec independent, it is important to test that the system is able to adapt to other generic codecs and detect alteration of audio from an unknown codec. We therefore performed a similar experiment as that presented in Section 6.1 but removing all audio encoded with Vorbis in the training and in the validation dataset. Thus the system was not able to learn from Vorbis encoded material. Results are reported for each codec in Table 3. They are very similar to those reported in Section 6.1, in particular performance on Vorbis are almost unaffected, which shows that the system is able to detect altered audio from other codecs than the one against which it was trained.

6.2.2. Detection with changing sampling rate

In a similar way, our approach aims to be robust to sampling rate changes in the history of audio material: if a piece of audio was encoded with a generic codec with sampling frequency 48kHz and after decoding resampled to 44.1kHz, frame synchronized techniques will fail at detecting alteration. We tested our system for robustness to sampling frequency change. The first attempt was to change only the test database by adding signals resampled from 44.1kHz to 48kHz, then compressed at 48kHz and then resampled back to 44.1Hz. This unfortunately led to quite bad classification results, the system being unable to generalize to previously unseen sample rates. However, as there are only a few commonly used sample rates (the two most common for musical content being 48kHz and 44.1kHz), we conducted a new experiment by adding signals with sampling rate changes to the training database and in the validation database. The global detection rate was 98.4% and results detailed by codec are reported in Table 4: once again, they are very similar to the ones reported in Section 6.1, with a slight decrease of detection rate of unaltered audio files. This confirms that the approach can be robust to sample rate changes if the system is trained with examples that had such a change.

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

Table 3. Detection rate for each codec/bitrate for the codec robustness experiment.

Codec	Bitrate	Detection rate	Codec	Bitrate	Detection rate
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 4. Detection rate for each codec/bitrate for the sampling rate robustness experiment. Only rates under 99.5% are reported.

7. CONCLUSION

In this paper, we presented a CNN-based method to detect audio that has been compressed using a perceptual codec from PCM material. To the authors knowledge, this is the only study that includes many different codecs and that uses a database of 26844 unaltered audio files. The method reaches a detection rate of 98.6% which is comparable to state-of-the-art methods designed for a single codec, while being robust to codecs and sampling rate changes.

Future works may focus on detecting artifacts that are the most likely to be audible, thus estimating an actual perceptual quality of the audio content. It may also be interesting to include non-generic codecs in a future study, such as speech codecs. As it is probably easy to cheat the algorithm by adding small level of noise that would mask the artifacts, it would be interesting to study robustness to such processes as well.

8. ACKNOWLEDGEMENTS

The authors would like to thank Matt Mould who assisted in the proof-reading of the paper.

9. REFERENCES

[1] Karlheinz Brandenburg and Gerhard Stoll, "Iso/mpeg-1 audio: A generic standard for coding of high-quality digital audio," *Journal of the Audio Engineering Society*, vol. 10, no. 42, pp. 780–792, 1994.

[2] Marina Bosi, Karlheinz Brandenburg, Schuyler Quackenbush, Louis Fielder, Kenzo Akagiri, Hendrik Fuchs, and Martin Dietz, "Iso/iec mpeg-2 advanced audio coding," *Journal of the Audio Engineering Society*, vol. 10, no. 45, pp. 789–814, 1997.

[3] Xiph.Org Foundation, "Vorbis I specification," `https://xiph.org/vorbis/doc/Vorbis_I_spec.pdf`, February 2015.

[4] Paul Bießmann, Daniel Gärtner, Christian Dittmar, Patrick Aichroth, Michael Schnabel, Gerald Schuller, and Ralf Geiger, "Estimating mp3pro encoder parameters from decoded audio," in *Proceedings of the 2nd Workshop Audiosignal- und Sprachverarbeitung (WASP)*, Koblenz, Germany, September 2013.

[5] Tiziano Bianchi, Alessia De Rosa, Marco Fontani, Giovanni Rocciolo, and Alessandro Piva, "Detection and localization of double compression in mp3 audio tracks," *EURASIP Journal on Information Security*, 2014.

[6] Brian D'Alessandro and Yun Q. Shi, "Mp3 bit rate quality detection through frequency spectrum analysis," in *Proceedings of the 11th ACM Workshop on Multimedia and Security*, New York, NY, USA, 2009, MM&Sec '09, pp. 57–62, ACM.

[7] Jürgen Herre and Michael Schug, "Analysis of decompressed audio-the inverse decoder," in *109th AES Convention*, Los Angeles, California, USA, September 2000.

[8] Sascha Moehrs, Jürgen Herre, and Ralf Geiger, "Analysing decompressed audio with the inverse decoder - towards an operative algorithm," in *112th AES Convention*, May 2002.

[9] D. Seichter, L. Cuccovillo, and P. Aichroth, "Aac encoding detection and bitrate estimation using a convolutional neural network," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 2069–2073.

[10] Rui Yang, Zhenhua Qu, and Jiwu Huang, "Detecting digital audio forgeries by checking frame offsets," in *Proceedings of the 10th ACM Workshop on Multimedia and Security*, New York, NY, USA, 2008, pp. 21–26, ACM.

[11] Daniel Gärtner, Christian Dittmar, Patrick Aichroth, Luca Cuccovillo, Sebastian Mann, and Gerald Schuller, "Efficient cross-codec framing grid analysis for audio tampering detection," in *136th AES Convention*, April 2014.

[12] Rui Yang, Yun-Qing Shi, and Jiwu Huang, "Defeating fake-quality mp3," in *Proceedings of the 11th ACM Workshop on Multimedia and Security*, New York, NY, USA, 2009, pp. 117–124, ACM.

[13] S. Hicsonmez, E. Uzun, and H. T. Sencar, "Methods for identifying traces of compression in audio," in *Communications, Signal Processing, and their Applications (ICCSPA), 2013 1st International Conference on*, Feb 2013, pp. 1–6.

[14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton, "Classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems*, December 2012.