Impact of audio codec and quality on genre classificaton and BPM recognition in Essentia

Sjoerd Hulleman
Supervisors: Cynthia C. S. Liem , Jaehun Kim
EEMCS, Delft University of Technology, The Netherlands
{s.hulleman}@student.tudelft.nl, {c.c.s.liem, j.h.kim}@tudelft.nl

Abstract

Music Information Retrieval (MIR) is a field of research that focusses on extracting information from music related data. This includes the genre of music and the beats per minute (BPM) of a song. Pipelines that extract this information from music are called feature extractors. Essentia is a library for such feature extraction. Often, the audio codec and quality is not considered in research setups within the field of MIR, while this could have an influence on the results. Therefore the main research question is "How do different audio codecs and audio quality impact genre classification and beats per minute (BPM) recognition in Essentia?". To answer this, the genre has been narrowed down to rock and the chosen audio codecs are FLAC, MP3 LAME and OGG Voribs. In collaboration with Muziekweb, a Dutch music library that collects all music that has been released in The Netherlands, it was possible to gather music files in lossless format. To degrade the audio quality, classify songs and recognize BPM, python pipelines for codec conversion, rock genre classification and BPM recognition were created an ran on this data. It has been concluded that changes in audio codec and quality have an influence on genre classification and BPM recognition in Essentia. It has not been concluded which codec and quality is best to use in the field of MIR. Further research is needed to answer this.

1 Introduction

Music Information Retrieval (MIR) focuses on extracting audio features from music, such as tempo, genre and tone. Genre classification and BPM recognition are part of the practice of Music Information Retrieval. Feature extractors transform music into numbers such as rhythm descriptors and tonal descriptors. These numbers are used in pipelines to automatically associate humanly meaningful information to these numbers. In many cases these feature extractors make use of machine learning techniques (Liem and Kim, 2021), which will also be the case for this research. Essentia is such a feature extractor. It "is an open-source C++ library for audio analysis and audio-based music information retrieval" (Essentia, 2019).

Over the past years it has been questioned if machine learning procedures which extra audio features actually extract musically meaningful information (Liem and Kim, 2021). From this, the background of this research originates. Earlier research has been performed on this subject. Urbano et al. (2014) researched the robustness of some of the popular music signal features with varying audio quality. Their results showed that these are robust within reasonable limits. In this context, robustness means that the extracted features have little to no change compared to the results with the original audio quality.

Liem and Mostert (2020) have conducted research, from which it is concluded that audio quality and codecs could have an influence on results of feature extraction. However, this is not concluded with certainty. Furthermore, lossy formats "distort the original signal and therefore may affect the computation of descriptors" (Urbano et al., 2014).

In the field of MIR, datasets are not always supplied with their original music corpus. This means researchers will have to obtain the music files themselves (Urbano et al., 2014). A reason for datasets not including original music files can be copyright. Spreading original music files without the artist's permission is not legal. In the end this means that MIR is often based on music that may use different audio encodings (Urbano et al., 2014).

According to Liem and Mostert (2020), anomalous behaviour in feature extraction can be caused by audio codecs and compression rates. These are also "rarely explicitly considered and reported in evaluation setups" (Liem and Mostert, 2020). This means we are not sure if audio codecs and compression rates have an impact on audio feature extraction.

Impact of audio quality and codecs therefore is an important subject to research, since this can possibly have a major impact on performance of, for example, genre classification. Essentially, if an audio file is encoded with a lower bitrate, meaning less bits are used per second, less data is present in the file. This is a loss of information, thus a music extractor will have less information to use. If audio quality and codec has a large impact on this process, it should always be considered when performing audio feature extraction.

This research focuses on the following research question: "How do different audio codecs and audio quality impact genre classification and beats per minute (BPM) recognition in Essentia?". In this research, the FLAC (Free Lossless Audio Codec) encoding will be used as the original lossless format. This will then be degraded to different MP3 and OGG encodings with different bitrates, which are lossy encodings. The research will be split into the following sub-questions:

1. What is the performance of genre classification with FLAC encoding?
2. What is the performance of BPM recognition with FLAC encoding?
3. How is genre classification influenced by MP3 quality/bitrates?
4. How is genre classification influenced by OGG quality/bitrates?
5. How is BPM recognition influenced by MP3 bitrates?
6. How is BPM recognition influenced by OGG quality/bitrates?

The paper is structured as following, the data, algorithms and pipelines are discussed in section 2. The results of the research are shown with accompanying figures in section 3. In this same section these results will be discussed and conclusions will be drawn. Future work will be presented in section 4. In section 5, the context of responsible research will be discussed.

2 Methodology

In this section, all components needed for the research and their details will be discussed. In subsection 2.5 it is explained how these components are combined and in the final subsection the retrieval of results is explained. The code used for this research and the results of the research can be found on a GitLab[1] repository.

2.1 Essentia

To conduct this research, Essentia was used. With this library, a Python environment was set up to process the audio files and extract audio features. The audio features which were used for this research are genre and BPM.

2.2 Dataset and justification

To be able to conduct research on audio, a dataset is needed. In this case, this audio was rock music. To justify the choice of rock music, results of previous work of Sturm (2014) were used. This work researched the effects of "irrelevant transformations" on audio files to genre classification. Irrelevant transformations are transformations on the music audio files which do not change the music itself. This can for example be the addition of white noise or a change in bitrate or encoding.

Figure 1 shows the results of Sturm (2014). From these results it can be concluded that the genre classifiers "rock_pop", "metal_punk" and "jazz_blues" are most sensitive to these transformations. Therefore it will be interesting to research if audio codecs and quality will have a major impact on the classification of these genres. This research will focus on the genre rock , following from the "rock_pop" classifier used by Sturm (2014).

To evaluate the robustness, often a ground truth is needed. A ground truth means in this case, a dataset which has the correct labels which describe the genre and BPM of the songs included. In collaboration with Muziekweb, it has been possible to create a large dataset of music. Muziekweb is a Dutch music library which exists since 1961, they collect all music that has been released in The Netherlands (Muziekweb, 2021b). Muziekweb is run by a team of music professionals, among which musicologist (Muziekweb, 2021a). In collaboration with the TU Delft, they can provide songs based on our needs. All songs were labeled with their genre. This way it was possible to create a dataset which had a ground truth.

Gathering data within the field of music information retrieval can be challenging due to copyright issues. It is not possible to simply share artists' full music without making agreements. Therefore the research group has signed a data delivery and non-disclosure agreement with Muziekweb, which means we cannot share the music with anyone.

To create a dataset of rock songs, the rock page of Muziekweb has been explored. From this, 66 albums within the rock genre have been selected. This was done by hand-picking albums on the rock page of Muziekweb[2]. Hand-picking these albums should not have influenced the results of this research, since these albums have already been classified within the rock genre by the experts of Muziekweb. The total songs in all albums amount to 1003 songs. These songs are produced by 53 different artists.

[1] https://gitlab.ewi.tudelft.nl/cse3k-21q2-music-faithfulness/project-sjoerd-hulleman

[2] https://www.muziekweb.nl/en/Link/T00000000354/Rock

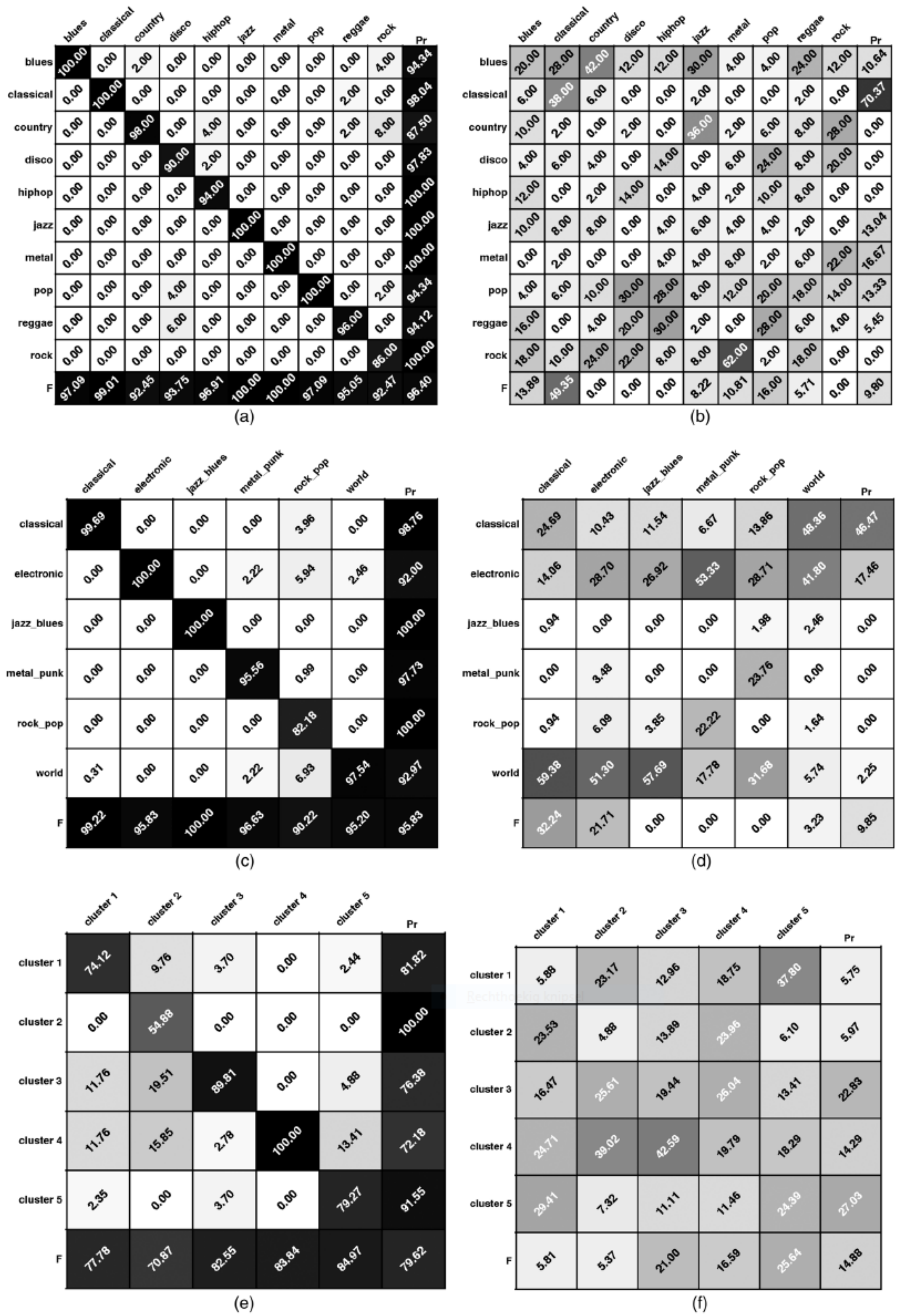

Figure 1: Genre classification before (left) and after (right) transformations. Reprinted from "A Simple Method to Determine if a Music Information Retrieval System is a 'Horse'", by Sturm B. L., 2014, *IEEE Transactions on Multimedia, 16*(6), p. 1636-1644.

2.3 Codecs and quality

Changing audio quality and codec was done with the AudioSegment module of the Pydub library (Jiaaro, 2021). This library internally uses the FFmpeg library which can convert to LAME MP3 VBR (Variable Bit Rate) and OGG Vorbis format (FFmpeg, 2014). A Python pipeline was created to load all FLAC files provided by Muziekweb and then convert them into the following codecs:

- MP3 96 kbps
- MP3 128 kbps
- MP3 256 kbps
- MP3 320 kbps
- OGG Vorbis 64 kbps
- OGG Vorbis 96 kbps
- OGG Vorbis 128 kbps
- OGG Vorbis 320 kbps

FLAC is a lossless audio codec, this means "audio is compressed in FLAC without any loss in quality" (Xiph, 2019). The audio files supplied by Muziekweb were in CD quality, meaning it is at 44100 Hz sample rate at 16 bit. This is one of the highest digital audio qualities and therefore was suitable to function as the ground truth data quality for this research, which also functions as the data to answer sub-questions 1 and 2.

MP3 is also called MPEG-1/2 Layer-3, which is an open standard (Brandenburg, 2001). This means anyone can create an encoder and decoder which complies with the MP3 standard. The difference in MP3 encoders/decoders lies within the method of decoding and encoding. This method can be patented by companies. Companies cannot be owner of the MP3 standard (Brandenburg, 2001).

The bitrate range of the MP3 standard is 96 kbps to 320 kbps (Adobe, 2022). Kbps means kilobits per second, so how many bits are used to encode the audio signal every second. This is a lossy format, this means "the higher the compression ratio becomes, the lower the resulting final audio quality" (Hans and Schafer, 2001). A higher compression ratio means a lower bitrate.

MP3 encoders can encode with CBR, VBR and ABR. CBR stands for Constant Bitrate, this means the encoder will encode on a preset bit rate (e.g. 320 kbps) over the entire file. VBR is Variable Bitrate, this means the encoder will only write as many bits as needed to create a MP3 file. An advantage of VBR over CBR is that it will use less space for the same bitrate. However the file size of a CBR encoded file is predictable, that of a VBR encoded file is not due to the variable amounts of bits used every second. ABR, Average Bitrate, is a comprimise between VBR and CBR. It is supplied with a target bitrate and varies its encoded bits around that target bitrate (Hydrogenaudio, 2020).

LAME is an encoder/decoder within the MP3 standard. It is open source licensed. Currently "LAME is considered the best MP3 encoder at mid-high bitrates and at VBR" (LAME, 2017). The development of LAME mainly focuses on increasing the speed and quality of encoding.

OGG Vorbis is a "fully open, non-proprietary, patent-and-royalty-free, general-purpose compressed audio format for mid to high quality (8kHz-48.0kHz, 16+ bit, polyphonic) audio and music at fixed and variable bitrates" (Xiph, 2016). This means OGG Vorbis also is a lossy codec. OGG Vorbis has a better compression rate, this means OGG Vorbis will sound better at the same bitrate as e.g. MP3. Or at the same audio quality, OGG Vorbis will have a smaller file size (Xiph, 2003).

The quality of OGG Vorbis is "not best measured in kilobits per second, but on a scale from -1 to 10 called 'quality'" (Xiph, 2003). But to easily compare to MP3, we can express this quality setting in an average bitrate. A quality setting of 0 is similar to 64 kbps average and a quality setting of 10 is similar to 400kbps (Xiph, 2003). A maximum of 320 kbps is used since this already ensures a very high quality and will outperform MP3 320 kbps in terms of quality.

2.4 Genre classification and BPM recognition

For genre classification the pre-trained TensorFlow models of Essentia were used. TensorFlow is an "end-to-end open source platform for machine learning" (TensorFlow, 2021). Specifically for genre classification the TensorflowPredictVGGish algorithm of Essentia was used[3]. This algorithm, in combination with the genre_rosamerica[4] dataset available in Essentia, has been shown to have the highest accuracy with 0.94 (Essentia Labs, 2020). A CNN model in combination with the genre_electronic dataset has a higher accuracy of 0.95, but this dataset does not contain the rock genre.

To recognize the BPM or tempo of songs, the RhythmExtractor2013 algorithm[5] of Essentia has been used. This is the most up to date and effective algorithm of Essentia to give a tempo estimation for the input song.

2.5 Implemented pipeline

All before mentioned algorithms resulted in pipelines created in Python. An overview of these pipelines are shown in figure 2. To convert the FLAC files supplied by Muziekweb into MP3 and OGG Vorbis, an audio converter was written. This audio converter loaded all songs in FLAC format using the AudioSegment module from the Pydub library. To make the audio converter as efficient as possible, it loaded every FLAC file only once and then converted it into all before specified MP3 and OGG Vorbis bitrates using AudioSegment. All

[3]https://essentia.upf.edu/reference/streaming_TensorflowPredictVGGish.html

[4]https://mtg.github.io/essentia-labs/news/tensorflow/2020/01/16/tensorflow-models-released/

[5]https://essentia.upf.edu/reference/std_RhythmExtractor2013.html

formats of the audio files were saved to provide the option of re-using them if needed.

These audio files were then fed into a pipeline which classified the song into genres and estimated its BPM. For this, the MonoLoader included in Essentia was used[6]. The MonoLoader mixed the stereo channels of each audio file into a single channel. This was needed since the RhythmExtractor2013 and TensorflowPredictVGGish algorithms only accepted a single channel.

For the TensorflowPredictVGGish algorithm, the song was loaded with a sample rate of 16000 Hz. This was needed since this algorithm only works at this sample rate according to the instructions of Essentia Labs (2020). The song fed into the RhythmExtractor2013 algorithm was sampled at its original sample rate of 44100 Hz, since this algorithm needed this sample rate in order to work correctly. All results were saved into a CSV file.

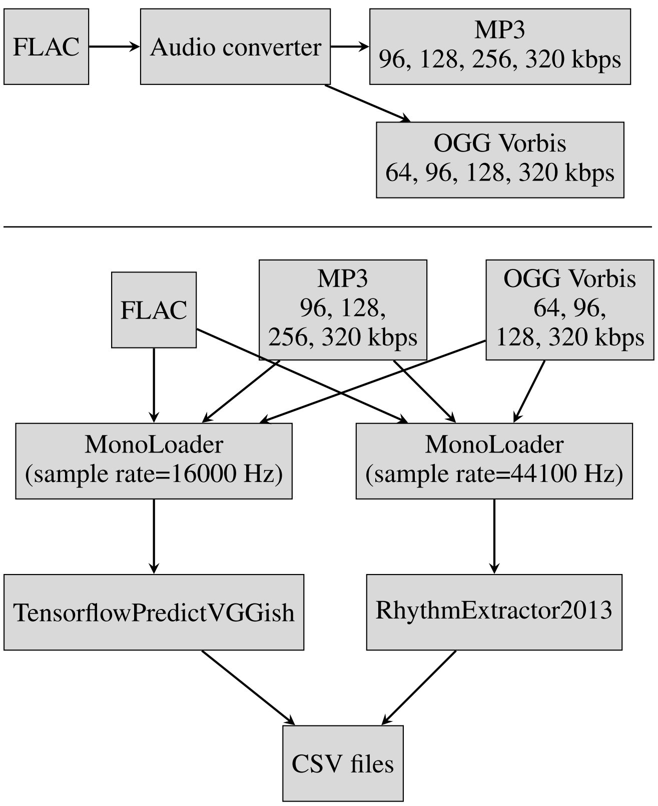

Figure 2: Overview of the implemented pipelines.

2.6 Calculating results

The results found by these pipelines are expressed in the probability per genre and an estimation of the amount of beats per minute for each song. Results for the genres rock, pop, jazz, rnb, dance, hip hop, classic and also speech were given by the implemented pipeline, but only the rock genre was used for this research.

[6]https://essentia.upf.edu/reference/std_AudioLoader.html

For the rock genre probability we have taken the mean of all results for each codec. In the same manner, we have taken the mean of the BPM estimations per codec.

Furthermore, another good measure to show deviations in results is the mean squared error (MSE). The MSE is a better measure to show deviations instead of the mean deviation. With a mean deviation results might cancel out due to deviations being positive as well as negative numbers. The MSE does not have this side effect. The formula for the MSE can be found in equation 1.

$$\frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{1}$$

In this equation, when calculating the MSE for rock probability, Y_i is the probability for each song for each codec and $\hat{Y}_i$ is the probability for the same song in FLAC. When calculating the MSE for BPM estimation Y_i is the BPM estimation for each song for each codec and $\hat{Y}_i$ again is the BPM estimation for the same song in FLAC. Sub-questions 1 and 2 form the baseline for the calculations with MP3 and OGG results. We calculated the MSE for the results of OGG and MP3 file encodings to answer sub-questions 3 up to and including 6.

To visualise these results on song level, we took the log squared error with base 10 for each song per codec. This can be seen in the boxplots in figure 4 and 6. Note that the mean of the log_{10} squared error in equation 2 is different to the log_{10} of the MSE calculated with equation 1.

$$log_{10}((Y_i - \hat{Y}_i)^2) \tag{2}$$

When calculating this measure for BPM, some calculations lead to $log(0)$, which is not possible. To prevent this, these cases have been mapped to 1^{-10}.

3 Results, discussion and conclusion

After running all pipelines the results of the genre classification and BPM or tempo esitmation with Essentia have been evaluated. The results are expressed in the probability per genre and an estimation of the amount of beats per minute. Since this research focusses on the rock genre specifically, we will look at the probability for rock. All tables are ordered from low to high. Higher or lower does not mean better or worse in this context, this is only to clearly show there are differences with each codec.

When drawing conclusions from the presented data, our goal is not to decide which codec gives a better prediction for the rock genre probability or the BPM. This is also something we can not conclude from our data, especially for the rock genre classification since most rock songs are often not purely rock, but also other (sub)genres. This means the algorithm probability will most likely never reach 100 percent certainty of a song being rock. The BPM estimation can be checked on being better or worse among different codecs, since a song is composed at a certain BPM, but this is not within the scope of this research. Our goal is to show the impact of audio quality and bitrates.

3.1 Genre classification

Audio codec	Rock probability
OGG 64 kbps	0.205927
OGG 320 kbps	0.219653
FLAC	0.223278
MP3 320 kbps	0.223490
MP3 256 kbps	0.223890
OGG 128 kbps	0.226068
MP3 128 kbps	0.238958
MP3 96 kbps	0.244565
OGG 96 kbps	0.249704

Table 1: Mean algorithm probability for rock.

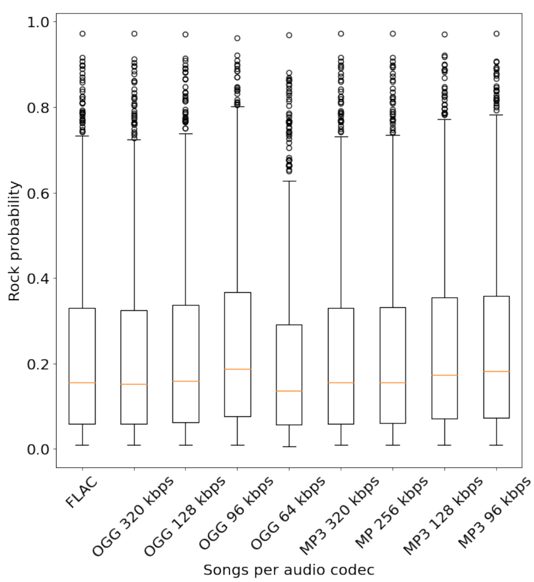

Figure 3: Algorithm rock probability per codec for all songs.

In table 1 we can see the mean of the probability for rock. In figure 3 the rock probability is visible per song, for all used audio codecs. This is shown as a boxplot, which clearly shows the range of the probabilities.

From table 1 it is very clear to see that the three highest quality codecs, FLAC, OGG 320 kbps and MP3 320 kbps, have similar results. With MP3 320 kbps being the most similar to our baseline codec FLAC. With further degradation of the audio into the other codecs, the mean rock probability also keeps changing with a significant amount. There seems to be a consistent change in probability when getting to lower quality audio, however OGG 64 kbps makes a jump to the top of the table. If we also look at figure 3 it is clear there is variation in rock probability when changing to each codec.

Audio codec	MSE rock probability
FLAC	0
MP3 320 kbps	0.000001
MP3 256 kbps	0.000002
OGG 320 kbps	0.000024
OGG 128 kbps	0.000054
MP3 128 kbps	0.000449
MP3 96 kbps	0.001160
OGG 96 kbps	0.001369
OGG 64 kbps	0.001511

Table 2: Mean squared error (MSE) of rock probability with results from FLAC as expected value.

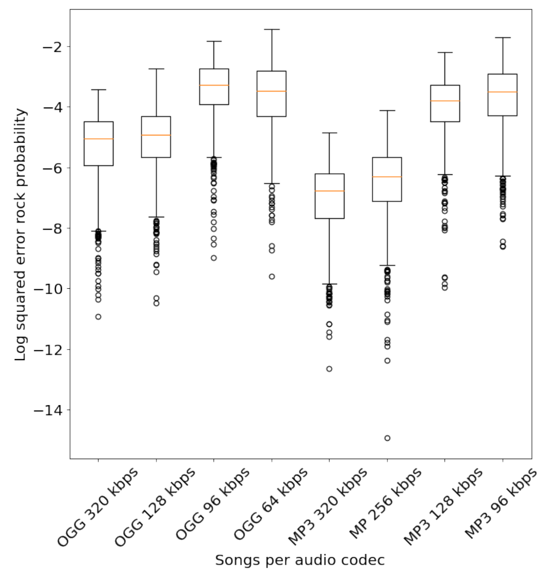

Figure 4: log_{10} squared error of rock probability for all songs compared to FLAC. Logarithmic transformation is only done for visualization purpose. To prevent taking a logarithm of zero, these cases have been mapped to 1^{-10}.

In table 2 we can see the mean squared error (MSE) of the rock probability for all songs for each codec. As explained in the methodlogy section, the MSE is a better measure to show deviations instead of the mean deviation. In this table we can see a strong correlation between the MSE going up when degrading the audio quality. What also stands out is that, although claimed by OGG Vorbis they are the best encoder for mid to high quality (Xiph, 2016), MP3 gets a lower MSE at similar bitrates. However, with this information we can not conclude which codec sounds better for the human listener. A further visualization of this data can be seen in figure 4. As indicated in the methodology, this measure uses the log (base 10) squared error per song. From these tables and figures we can clearly see there is a variation in rock genre probabilty between different quality codecs.

Considering all before mentioned data and findings, we can conclude the audio codec and quality has an impact on the genre classification of rock in Essentia. No conclusion however can be drawn on which codec is best suited for this genre classification task. Especially when looking at the MSE shown in table 2 and figure 4 we can see there is variation in the results of genre classification when using different audio codecs and quality.

3.2 BPM recognition

Audio codec	Mean BPM
MP3 96 kbps	123.225859
OGG 64 kbps	123.506569
MP3 128 kbps	123.594979
OGG 96 kbps	123.606059
OGG 128 kbps	123.678289
MP3 256 kbps	123.815350
MP3 320 kbps	123.846293
FLAC	123.934706
OGG 320 kbps	124.029516

Table 3: Mean BPM per codec.

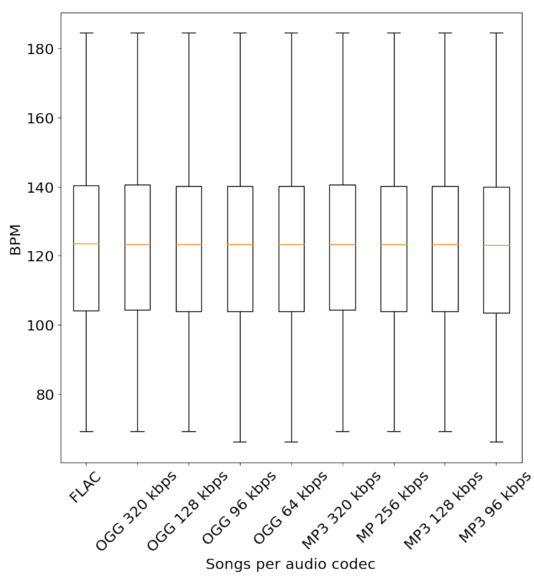

Figure 5: BPM estimation per codec for all songs.

Table 3 shows the mean BPM per codec. This is also visualised as a boxplot in figure 5. Looking at table 3 we can see that, just like with table 1, the highest quality codecs, FLAC, OGG 320 kbps and MP3 320 kbps have similar results in the mean BPM per codec. However, to draw a better conclusion we should once again look at a better measure to spot differences.

In table 4 we can see the MSE of BPM per codec, which is a better measure than the mean BPM. If we look at this table is

Audio codec	MSE BPM
FLAC	0
MP3 320 kbps	18.640363
MP3 256 kbps	27.831408
OGG 320 kbps	29.732891
MP3 128 kbps	37.494367
OGG 128 kbps	66.388747
MP3 96 kbps	68.39757
OGG 96 kbps	75.439119
OGG 64 kbps	112.913352

Table 4: Mean squared error (MSE) of BPM with results from FLAC as expected value.

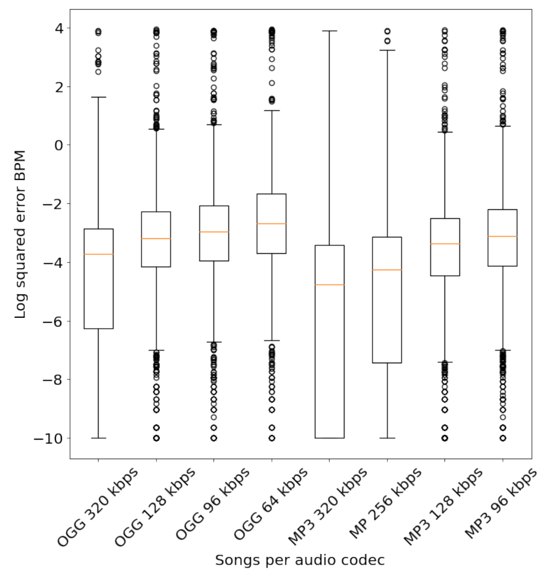

Figure 6: log_{10} squared error of BPM compared to FLAC. Logarithmic transformation is only done for visualization purpose.

visible that the MSE gets large relatively quickly compared to the MSE in table 2. This is mainly caused by BPM being higher numbers than algorithm confidence. Squaring these numbers leads to a larger results quickly. What stands out between these two tables is that the order of codecs from lowest to highest MSE is exactly the same, while the rock classification algorithm and BPM recognition algorithm are two separate algorithms.

As mentioned earlier, these algorithms also run on a different sample rate input. The codecs having the same ”ranks” in these tables could point to a correlation between audio quality and mean squared error compared to FLAC format. This table is visualized in figure 6. As indicated in the methodology, this measure uses the log (base 10) squared error per song and maps cases of $log(0)$ to 1^{-10}. This figure clearly shows the BPM estimation is affected by changes in audio quality and codecs, while figure 5 gives the idea that the BPM estimation is rather stable.

While the mean BPM estimation stays rather stable according to table 3 and figure 5, table 4 and figure 6 gives clear evidence there is a variation in the results for the BPM estimation per song. Therefore we can conclude the audio codec and quality have an impact on BPM recognition in Essentia.

4 Future work

The main take away from this research is that audio codecs and quality should be more explicitly considered in future work. Looking at our results, these factors have a clear influence on the produced results. The least which could be done, is keeping to a single audio codec and bitrate for all audio used in a research. This filters out the influence of different codecs within the results of that particular research. Our results could also indicate audio codecs and quality have an influence on other feature extraction pipelines within the field of MIR. Therefore further exploration of the influence of these factors within MIR can be done.

This research can also be further extended. As mentioned earlier, checking if the BPM estimation gets closer to the actual BPM of a song was not part of this research. This makes for an opportunity for further research, where this dataset can be cross referenced with a dataset that indicates the real BPM of a song. With this data, research can be done on which audio codec and quality give the best results in terms of BPM estimation.

Our dataset can also be extended with more songs or more genres. Growing a dataset often positively influences the accuracy of estimators and classifiers. In combination with this, also more codecs can be researched on their influence on MIR.

5 Responsible Research

To ensure all results of this research are reproducible, all code and metadata used for this project has been uploaded to a GitLab repository. Only the audio files will not be uploaded anywhere. We cannot share these files since they are copyright protected by the artist. Also a non-disclosure/data delivery agreement has been signed with Muziekweb. This means we are legally not allowed to share the music files shared by Muziekweb. This would also be unethical, sharing these files could lead to unwanted sharing on other parts of the internet.

The results of this research have not been affected by my personal preferences. While rock genre is one of my genres of choice when I listen to music, my choice was supported by data which indicated which genre could be most affected by ”irrelevant changes”, as explained in the methodology section. This turned out to be the rock genre. During the album selection procedure, I may have selected artists that I am familiar with. This does not affect the results however, since these albums were never selected outside the rock genre label of Muziekweb. All selected songs are within the rock genre.

Furthermore, there was no personal interest in a certain conclusion for this research. There is no personal advantage for me with any conclusion drawn for this research.

Finally, we have contributed to the field of MIR. We have shown audio quality and codec has an influence on the results obtained in the field of MIR. This supports future research done in this field. Also a bug has been found in Essentia and reported on their Github page[7]. This will help build new and better version of Essentia, which will positively influence all users of the Essentia library.

[7]https://github.com/MTG/essentia/issues/1228

References

Brandenburg, K. (2001). Mp3 and aac explained. *AES 17th International Conference on High Quality Audio Coding*.

Essentia (2019). *Homepage*. Retrieved November 24, 2021, fromhttps://essentia.upf.edu/index.htmll.

FFmpeg (2014). *FFmpeg MP3 Encoding Guide*. Retrieved November 17, 2021, from https://trac.ffmpeg.org/wiki/Encode/MP3.

Hans, M. and Schafer, R. (2001). Lossless compression of digital audio. *IEEE Signal Processing Magazine*, 18(4):21–32.

Jiaaro (2021). *Pydub*. Retrieved January 17, 2022, fromhttps://pydub.com/.

Liem, C. and Kim, J. (2021). Do automated music processing pipelines actually extract musical information? Retrieved November 12, 2021, fromhttps://projectforum.tudelft.nl/course_editions/42/projects/1189.

Liem, C. and Mostert, C. (2020). Can't trust the feeling? how open data reveals unexpected behavior of high-level music descriptors. In *Proceedings of the 21st International Society for Music Information Retrieval Conference*, pages 240–247. Virtual/online event due to COVID-19; 21st International Society for Music Information Retrieval Conference, ISMIR 2020; Conference date: 11-10-2020 Through 15-10-2020.

Muziekweb (2021a). *Colofon - Muziekweb*. Retrieved December 16, 2021, from https://www.muziekweb.nl/Muziekweb/Informatie/Colofon.

Muziekweb (2021b). *Welkom op Muziekweb - Muziekweb*. Retrieved December 16, 2021, from https://www.muziekweb.nl/Muziekweb/Informatie/Welkom.

Sturm, B. L. (2014). A simple method to determine if a music information retrieval system is a 'horse'. *IEEE Transactions on Multimedia*, 16(6):1636–1644.

TensorFlow (2021). *TensorFlow*. Retrieved December 16, 2021, from https://www.tensorflow.org/.

Adobe (2022). *Understanding audio bitrate and audio quality*. Retrieved January 11, 2022, from https://www.adobe.com/creativecloud/video/discover/audio-bitrate.html.

Essentia Labs (2020). *A collection of TensorFlow models for Essentia*. Retrieved December 16, 2021, from https://mtg.github.io/essentia-labs/news/tensorflow/2020/01/16/tensorflow-models-released/.

Hydrogenaudio (2020). *LAME - Hydrogenaudio Knowledgebase*. Retrieved January 11, 2022, from https://wiki.hydrogenaud.io/index.php?title=LAME#VBR_.28variable_bitrate.29_settings.

LAME (2017). *LAME MP3 Encoder*. Retrieved January 11, 2022, from https://lame.sourceforge.io/.

Xiph (2003). *Vorbis.com:FAQ*. Retrieved January 11, 2022, from https://xiph.org/vorbis/faq/#what.

Xiph (2016). *Xiph.org*. RetrievedJanuary11,2022,fromhttps://xiph.org/vorbis/.

Xiph (2019). *FLAC - Free Lossless Audio Codec*. Retrieved January 11, 2022, from https://xiph.org/flac/.

Urbano, J., Bogdanov, D., Herrera, P., Gómez, E., and Serra, X. (2014). What is the effect of audio quality on the robustness of mfccs and chroma features? In *Proc. of the 15th Conference of the International Society for Music Information Retrieval (ISMIR 2014)*, pages 573–578, Taipei, Taiwan.