Master thesis on Sound and Music Computing
Universitat Pompeu Fabra

Automatic Audio Problems Detection

Victor Badenas Crespo

Supervisor: Dmitry Bogdanov

Co-Supervisor: Pablo Alonso

June 2019

Master thesis on Sound and Music Computing
Universitat Pompeu Fabra

Automatic Audio Problems Detection

Victor Badenas Crespo

Supervisor: Dmitry Bogdanov
Co-Supervisor: Pablo Alonso

June 2019

Contents

Acknowledgement

I would like to express my sincere gratitude to:

- My supervisor
- My co-supervisor
- My family

Abstract

The abstract should have at least 200 but not more than 600 words. Placed on a right page next to a blank left page. A list of keywords (approximately 3 to 5) should be just below the abstract, preceded by the word "Keywords". Keywords should be separated by ";".

Keywords: Imaging techniques; Cloud computing; Alzheimer

Chapter 1

Introduction

Audio Problems refer to unwanted or undesired sound contained in an audio file. Many particular issues are considered undesirable, either for mixing purposes or for perceptual quality purposes. Quality of audio recordings can both be subjectively evaluated by percetually detecting audio issues or can be also evaluatted objectively by algorithms detecting patterns or specific features extracted from audio files. This thesis will be focused in the development and evaluation of the objective way to detect audio issues.

1.1 Objectives

The thesis will have three main objectives:

- The first is to evaluate and tune existing algorithms, which are included in the Essentia library. Designed for detecting defects in songs, which its size is much greater than the target audio files of this thesis.

- The second one is to create and develop algorithms to detect other issues common in sound files. Three more issues will be evaluated in this thesis, low SNR, bit depth discrepancies and sample rate discrepancies.
  Low SNR (Signal to Noise Ratio) signals are defined by the SNR equation, which quantifies the relationship between the desired (S) and undesired (N)

part of the waveform. These signals will have a noise (or unwanted content) content higher than desired compared to the desired signal itself.

The other two defects are considered to be in the same group of audio defects which are the resolution issues: bit depth and sample rate discrepancies.

Digital audio samples are coded as bit streams, and the resolution or the available voltage levels that a sound can have is determined by the bit rate. Signals coded as 16 bit signals could be signals coded in 8 bits but in a 16 bit container, thus having unused bits. This kind of sounds have a certain tonality and users might be interested in knowing this features. The main reason for classifying this defect as such is that memory is wasted in containing no relevant information.

When an audio signal is recorded at a lower sample rate than the sample rate of its container, some bandwidth remains unused because of the nyquist limit. An extensive explanation on effective bandwidth, nyquist limit and bit codification can be found on [11]. Some previous work in this can be found on Geoffroy Peeters' paper [6] where a method to calculate the harmonic cutoff frequency which can be extrapolated to the effective bandwidth of a signal.

- The third objective is to evaluate those algorithms in a subset from Freesound.

1.2 Structure of the Report

The Report will be split into The State of the art, where the algorithms of Essentia will be introduced and explained, the methodology where the developed algorithms will be explained in detail and the results and conclusions, where the algorithms will be evaluated and the outcome of the thesis will be explained.

Chapter 2

State of the Art

Previous work on automatic audio problems detection has been done by Pablo Alonso-Jiménez, Lluis Joglar-Ongay, Dmitry Bogdanov, Xavier Serra in their paper Automatic detection of audio problems for quality control in digital music distribution presented to the Audio Engineering Society on March 2019 in Ireland. On this paper, multiple audio defect detections were proposed as well as multiple audio defects were described of which some will be relevant as they will be adapted and evaluated in this paper:

2.0.1 Phase defects

Phase defects correspond to unfavorable relationships between both channels of audio of a stereo file. In this case, the relationship that is interesting for detecting it is the linear correlation between both channels as measured by the pearson correlation coefficient. The pearson correlation is defined to be the division of the covariance of two signals and the product of the standard deviation of them. That means that the range of values that the coefficient can take is [-1,1] where the extremes are caused by the issues that we are tracking.

- False Stereo is caused mainly by the same information being in both channels of an audio file. This can be caused by exporting a mono file as stereo in a Digital Audio Workstation (DAW) for instance and it is detected when the pearson coefficient is close to 1.

- Out of Phase Stereo is caused by a sign shift between both channels of an audio file. This sign shift can be caused by multiple causes, the most common one in recordings is microphone placement. When summed into a single channel audio file (mono) some frequency components cancel due to the issue. The importance of good correlation between both channels on a stereo signal is seen in the amount of articles that explain the importance of phase correlation meters in audio design and audio mixing workflows. Some examples of those articles include [7] [4].

2.0.2 Silence defects

Having too much silence at the start or the end of the file can also be considered a problem as information with no valuable content is being stored. For the detection of this issue the authors propose a frame by frame analysis computation of the rectified envelope to find the last silent frame at the start of the file and the first silent frame at the end of the file.

2.0.3 Noise bursts defects

Noise bursts are characterized by a succession of artifactual samples, typically originated in the codification process of an audio file and are part of the group that the authors denote as digital audio artifacts and that are defined by a succession of samples that do not correspond to a realistic sound.
The detection of this issue is done by thresholding the peaks of the second derivative of the of the signal. The threshold is not fix but rather computed using an exponential moving average (EMA) filter over the quadratic mean value of the second

derivative of the input frame.

2.0.4 Humming defects

Humming is characterized by a low frequency (50-80Hz), steady, narrowband electrical noise added to the signal. This defect is very common in analog recording and mixing gear, as it is in the neighbourhood of the frequency of the electrical supply (50-60Hz). The authors use an algorithm proposed by Brandt and Bitzer in their paper "Automatic detection of hum in audio signals" where they propose an approach based on the computation of measuring the steadiness of the frequency bins on 10-30 seconds audio segments.

2.0.5 Click defects

Click defects are included in the previously mentioned digital audio artifacts. They are characterized by impulsive noise that can come from multiple sources such as plosive sounds or codification errors. To detects this artifacts, after LPC algorithms are used to define the stationary part of the waveform, this is subtracted from the audio file to then compute a peak thresholding of the output of a matched filter of that error signal.

2.0.6 Clipping defects

Clipping is a phenomenon given by an audio waveform exceeding the dynamic range of the media, it being the digital or the analog domain. However, we are focusing on the digital domain, where clipping is perceived when surpassed the [-1, 1] range of the values. This can be originated from DAC not having enough dynamic range to represent the signal that is passed as an input or because of badly executed limiting in mastering.

The method proposed by the authors is detecting streams of adjacent samples that have very little variance between them and computing an energy threshold in order to avoid silence to be detected as clipping.

Chapter 3

Methodology

For the evaluation and creation of the algorithms, the language of python3 was used. For the scope of the project there were 3 algorithms developed.

3.1 Bit depth defects detection

The bit depth detection algorithm proposed works by finding the discrepancy between the value in the header of the Wave container file and the extracted information from the data in it.

The byte rate is obtained from the bytes 28 to 31 of the header and then are converted following 3.1.

$$BR = \frac{b}{8} * fs = B * fs \tag{3.1}$$

Where BR is the bit rate, b is the number of bits with which the signal is coded, B is the number of bytes and fs is the sampling frequency or sampling rate. From there, the first value is obtained.

The second value for the comparison is obtained by computing the used bits in the samples of the audio file.

Samples in a PCM sound format are stored in hexadecimal values, which range from 0xFFFF to 0x0000 in samples coded in 16 bits. Each pair of values correspond to a byte of information and thus each value corresponds to 4 bits.

The first step in the algorithm is to divide the audio file in chunks of N samples and randomly select M of them (N and M being parameters in the algorithm, depending on the degree of confidence desired, they default to 100). Then all those values are converted to binary following 1. So that for example, two samples of 0x0AF0 and 0xFF20 will be translated in 0000 1010 1111 0000 and 1111 1111 0010 0000 respectively.

Table 1: Hexadecimal to binary equivalence

Hex	**Binary**
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

After the hex to bin conversion, an "or" operation is performed to detect which bits are never used. In the example mentioned before, the result of the operation would be 1111 1111 1111 0000. As a little endian configuration is used to read the values, the lower bits are used but from the 16 bits used to code the samples, 4 bits remain unused thus meaning that coding the samples with 12 bits would be such as effective. However, only the higher weight unused bits contribute to the count of unused bits as the bits of lower weight are needed to represent the values of the higher values.

3.2 Bandwidth defect detection

The bandwidth defect detection algorithm proposed is based in the same principle than the bit depth detection: the discrepancy between the information stored in the header and the information extracted from the data stored in the data.

For the header information, is retrieved from the bytes 24 to 27. After this is compared to the extracted frequency from the data.

To extract the information, a frame per frame analysis is done. From each frame, the most likely cut frequency is computed (defining the cut frequency as the last bin with relevant information). From the possible frames, a histogram is computed and the most probable frequency is taken as the result and a confidence value is also computed from the histogram.

The most likely bin per each frame is computed using 3.2:

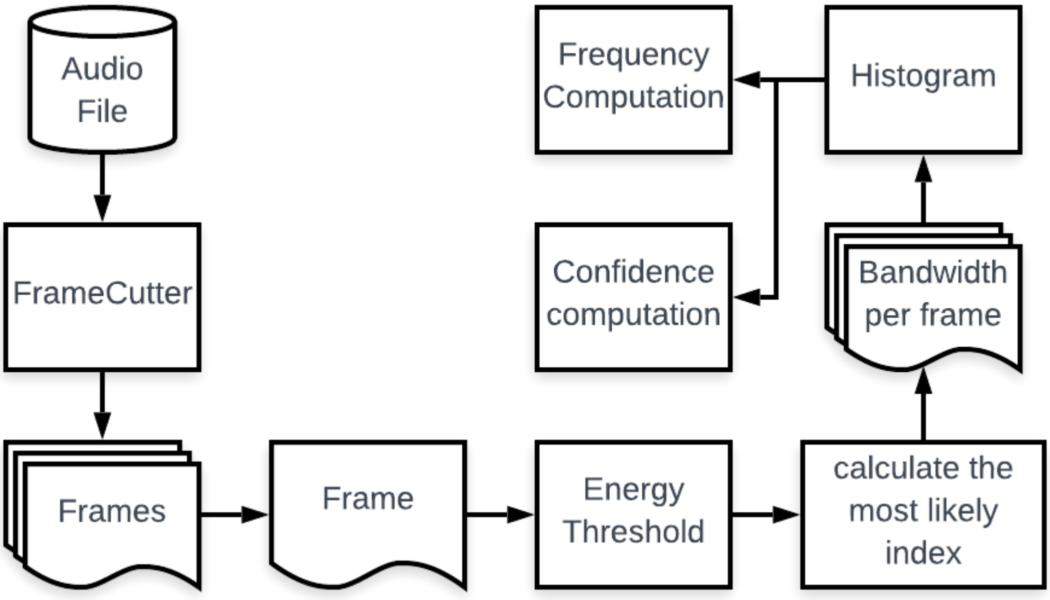

Figure 1: Flowchart for BW detection

$$Fc = max(Fc\ where\ \frac{1}{Fc}\sum_{i=Fc}^{N} FFT(x_{frame} * w_N)[i] > 10^{-5} where Fc = 0, 1, \ldots, N) \tag{3.2}$$

Which value and function was obtained by trying different functions that gave a stable value which to threshold. Once the cut frequency is computed in each frame, a weighted histogram is computed where instead of simply counting the number of occurrences for each frequency bin, the energy of the frame was added so that the equation results in 3.3

$$hist[f] = \sum_{i=0}^{N_{frames}} \sum_{j=0}^{N} (frame_i[j])^2\ for\ i\ so\ that\ f_i = f \tag{3.3}$$

Where Nframes is the number of frames, N is the frame size, f is the cut frequency, fi is the extracted frequency for that frame and hist[f] is the histogram array in the index f. From that weighted histogram, the most likely frequency is taken by extracting the index for which the histogram is largest. Fc=argmax(hist) and the confidence is extracted in two different ways depending on the value of Fc:

From there, if the confidence value is greater than a determined threshold, defaulted to 0.8 and Fc is lower than 0.85*(N/2 +1) it is considered to have a problem. However, this methodology only has been explored with a frame size of 256 samples and a hop size of 128 at the moment.

3.3 Low SNR defect detection

The low SNR detection algorithm is based on the well known SNR relationship of 3.4

$$SNR = \frac{S}{N} \tag{3.4}$$

But as the signal and the noise component of this expression, are mixed in our case, a per frame classification algorithm based on the features of energy and correlation was used with the following conditions: If a frame's energy is less than a 10% of the energy the frame is classified as noise. If it is higher, the next evaluation is performed. When evaluating the correlation, the goal was to measure the resemblance of the frame to white noise. To do this, autocorrelation was used to compute it.

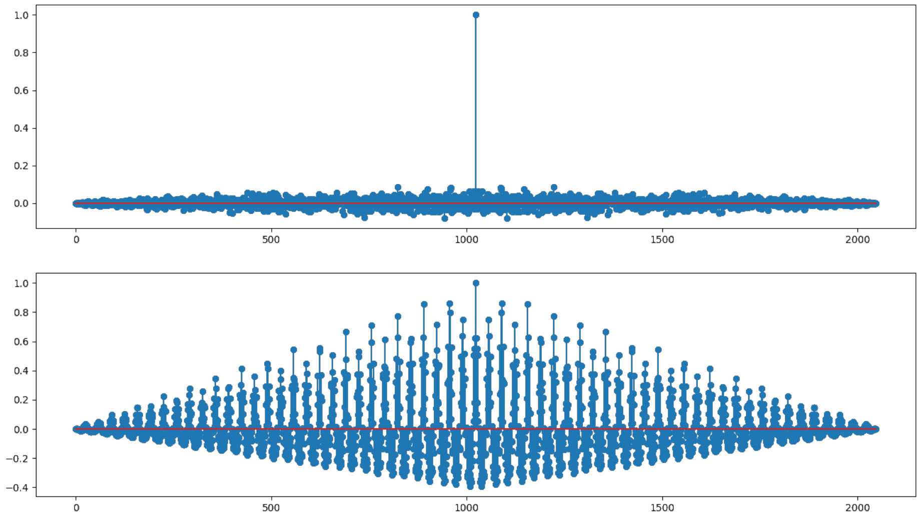

Figure 2: Autocorrelation comparison of N(0,1) noise and a more deterministic signal

Figure 2 represents an autocorrelation plot for white noise N(0,1) and the plot below corresponds to an almost pure harmonic sound with some noise added to it. This raises the question if a measure of which percentage of the values lands in the sample in the middle is a good fit for a measure of similarity, which in further testing it is found that it is. The mathematical expressions for the computation to get the value for deciding on which kind of frame we are looking at is 3.5.

$$\begin{gathered} R_{xx}[l] = \sum_{n=0}^{N-1} x[n]x[n-l] \\ ac[i] = \frac{R_{xx}[0]}{\sum |R_{xx}[l]|} \; for \; i = 0, 1, ..., N_{frames} \\ ac = \frac{ac}{max(|ac|)} \end{gathered} \tag{3.5}$$

This ac vector of values will give us a value between 0 and 1 for each frame which is then thresholded and if the value is lower than the threshold is considered signal, and noise otherwise. From there, once the frames are classified, the power for each signal and noise frames are added and then the SNR is computed. A problem is detected if the SNR is higher than a threshold.

Chapter 4

Results

The algorithms have been evaluated with 1120 mono sounds from the test dataset for the Kaggle Audio tagging competition created by the MTG and Freesound. The test set for the dataset is an accurate representation of the content of the whole dataset for the competition. The dataset has been manually anotated by the author using the script evaluation.py in the Evaluation folder on thew github repository. The evaluation was carried by annotating the existance of the following problems: Hum, Clicks, Noise Bursts, Saturation and Silence. Other algorithms that had to be evaluated were Phase issues. However, as the dataset is formed exclusively of mono files, the algorithms were not properly evaluated.

The parameter values evaluated for each algorithms will be evaluated with 3 measures: precision 4.1, recall 4.2 and $F_{0.5}$ score 4.3. A value of $\beta = 0.5$ was chosen to weight the score in favor of the precision score.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive} \tag{4.1}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative} \tag{4.2}$$

$$F_\beta = (1+\beta^2)\frac{precision \cdot recall}{\beta^2 \cdot precision + recall}, F_{0.5} = 1.25\frac{precision \cdot recall}{0.25 \cdot precision + recall} \tag{4.3}$$

4.1 Hum detection evaluation

Four parameters of the essentia Hum algorithm were considered in the evaluation: TimeWindow, minimumDuration, NumberofHarmonics and timeContinuity. The timeWindow parameter is a measure of the analysis time to use in the Hum estimation in s. The parameter was evaluated for the values: [0.1, 0.3, 0.5, 1, 3, 5] as they seemed a good sweep of values for the parameter. However, as it can be seen in the results obtained in 3, they are not satisfactory, as the values remain constant at 0 and 1 which, inspecting the equations and knowing that a 0/0 indetermination will be considered with a value of 1 as a result, can be extrapolated that: a recall score of 0 means a TruePositive count of 0, and then, a precision value of 1 with a TruePositive count of 0 means that the denominator also is 0 and we can conclude that FalsePositive values are also 0 which means that all audio files were detected as Healthy and by manual revision it can be seen that there are audio files that have hum so this parameter is not relevant to the detection.

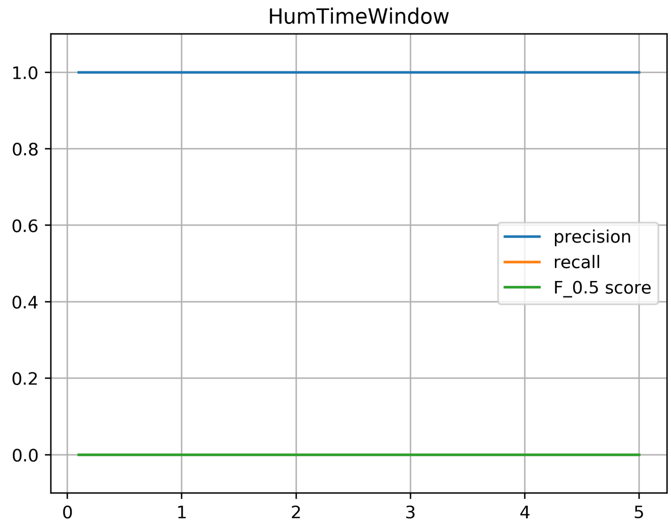

Figure 3: timeWindow parameter sweep results (accuracy, F score and recall)

The next parameter to be evaluated is minimumDuration, which is the minimum time span for which the hum is detected. The parameter was sweeped through the range: [0.01, 0.07, 0.1, 0.3, 0.5, 1, 3, 5]. However, similar results to the previous ones were obtained, as can be seen in 4.

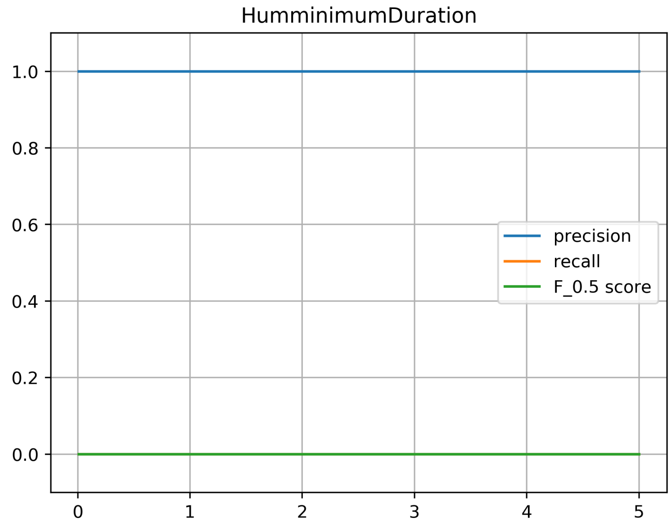

Figure 4: minimumDuration parameter sweep results (accuracy, F score and recall)

The next parameter to be evaluated is NumberofHarmonics, number of harmonic components for which the algorithm will look for it to consider a humming frequency. The parameter was sweeped through the range: [1, 2, 3, 4, 5]. However, similar results to the previous ones were obtained, as can be seen in 5.

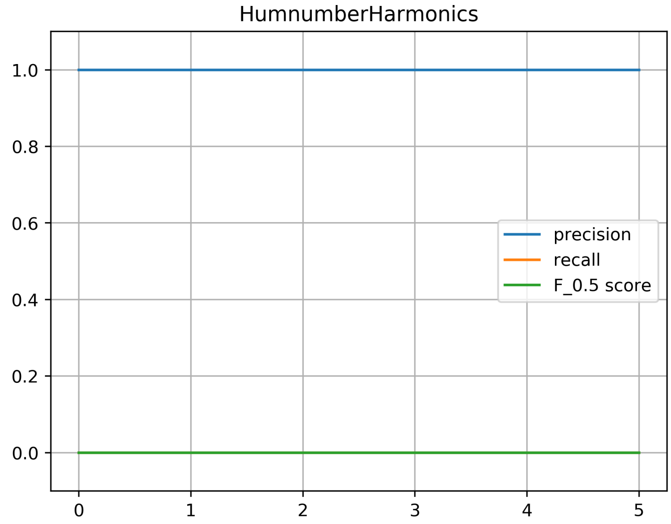

Figure 5: numberHarmonics parameter sweep results (accuracy, F score and recall)

The next parameter to be evaluated is timeContinuity, the minimum time for a humming frequency to be present to be detected. The parameter was sweeped through the range: [0.1, 0.3, 0.5, 1, 3, 5]. However, similar results to the previous ones were obtained, as can be seen in 6.

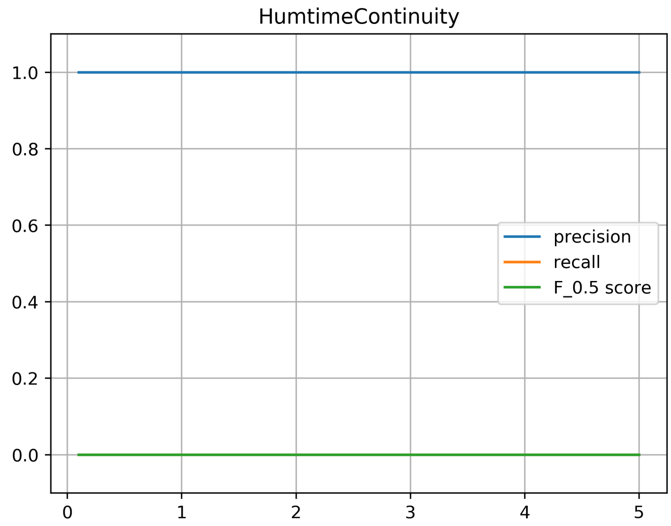

Figure 6: timeContinuity parameter sweep results (accuracy, F score and recall)

4.2 Clicks detection evaluation

Four parameters for the essentia clicks detector algorithm were evaluated: detectionThreshold, order, powerEstimationThreshold, silenceThreshold. The first parameter is detectionThreshold, which is based on the instant power of the noisy excitation signal plus detectionThreshold dBs. detectionThreshold parameter was swept in the range of values: [0, 5, 10, 15, 20, 25, 30, 35]. The results are as seen in 7. As seen in the figure, the best results (maximum of the Fscore result) were obtained for a value of 25dB and the obtained Fscore obtained is about a 0.35.

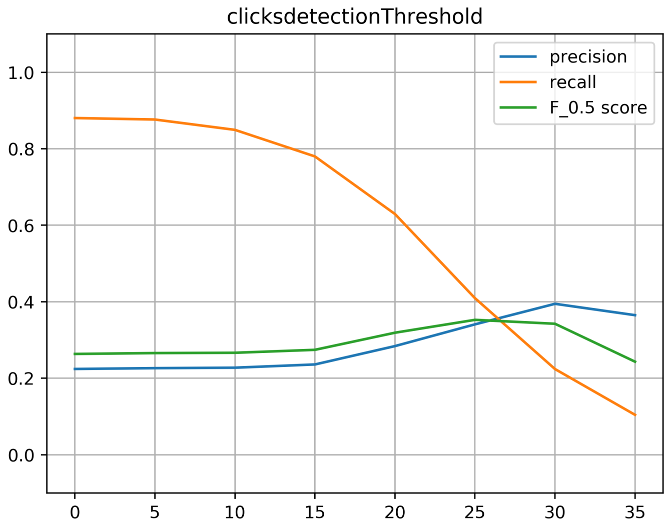

Figure 7: detectionThreshold parameter sweep results (accuracy, F score and recall)

The next parameter to be evaluated is order, which is the number of LPC coefficients to use. The parameter was evaluated for the values: [2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 32, 34, 36, 38]. The maximum Fscore value was obtained for a value of 24 and the result was a value of over 0.4.

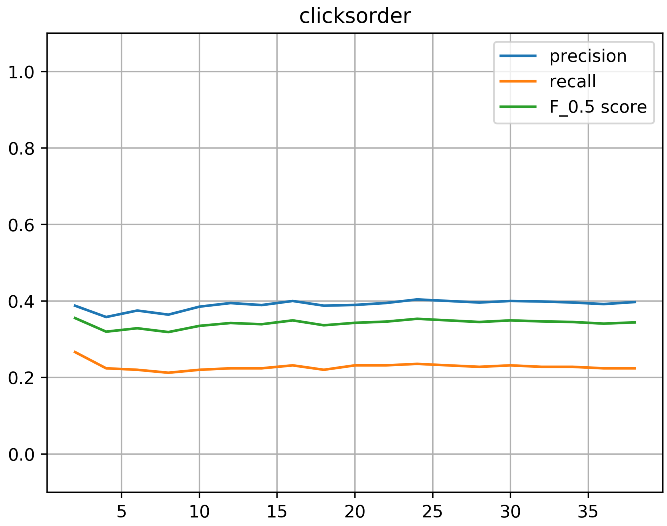

Figure 8: order parameter sweep results (accuracy, F score and recall)

The next parameter to be evaluated is powerEstimationThreshold. In the algorithm, the noisy excitation is clipped to powerEstimationThreshold times its median. The parameter was evaluated for the values: [2, 4, 6, 8, 10, 12, 14]. The maximum Fscore value was obtained for a value of 10 and the result was a value of over 0.35.

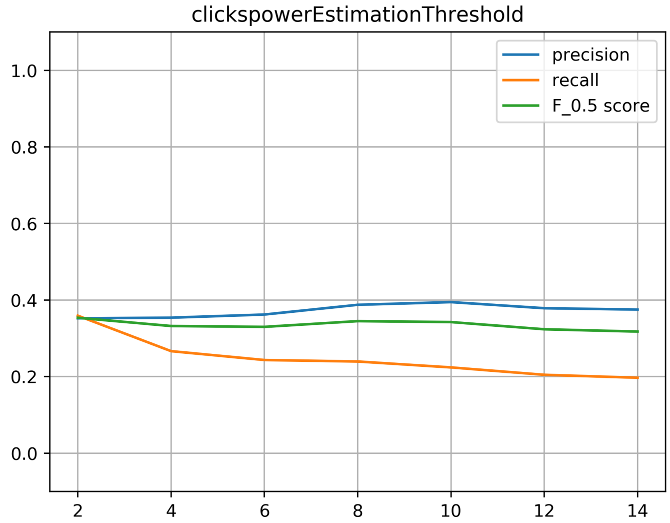

Figure 9: powerEstimationThreshold parameter sweep results (accuracy, F score and recall)

The next parameter to be evaluated is silenceThreshold. The value represents the threshold to skip silent frames. The parameter was evaluated for the values: [-70, -60, -50, -40, -30, -20, -10]. The maximum Fscore value was obtained for a value of -60 and the result was a value of over 0.35.

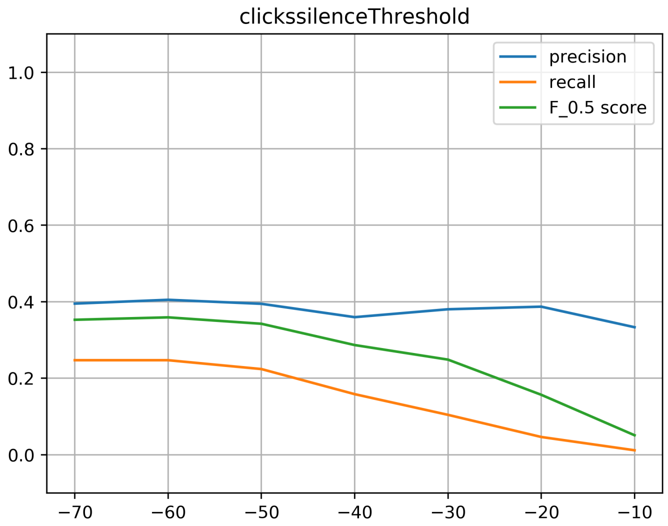

Figure 10: silenceThreshold parameter sweep results (accuracy, F score and recall)

4.3 Noise bursts detection evaluation

For the noise bursts detection algorithm two values were sweeped: alpha value and threshold. The alpha value is the alpha value for Exponential Moving Average threshold estimation. The valuye was sweeped for the range: [0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9] and the best value of Fscore was achieved for 0.1.

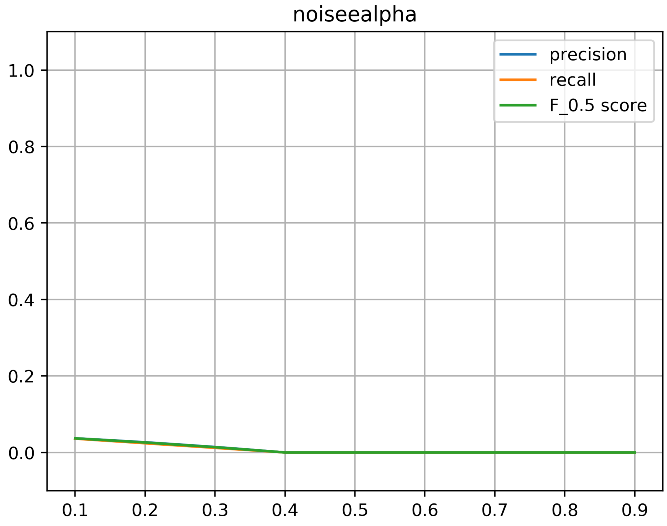

Figure 11: alpha parameter sweep results (accuracy, F score and recall)

The threshold value represent the threshold to skip silent frames. The parameter was swept for the following values: [-10, -8, -6, -4, -2, 0, 2, 4, 6, 8, 10] and the best results were obtained for the value of 4 giving an accuracy of 0.3.

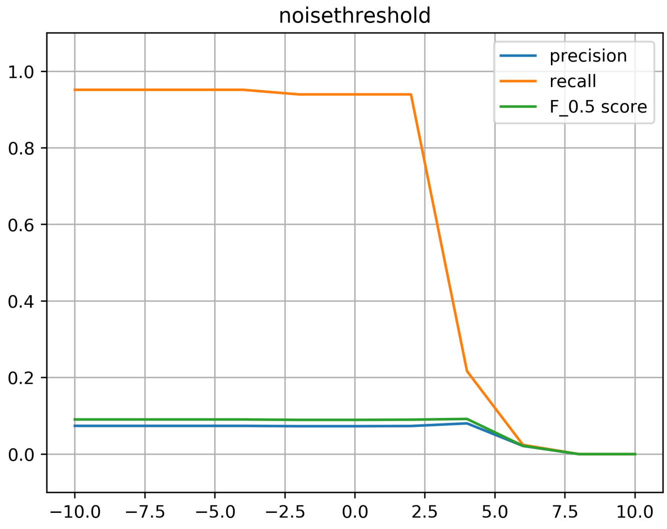

Figure 12: threshold parameter sweep results (accuracy, F score and recall)

4.4 Saturation detection evaluation

For the saturation evaluation, the parameters differentialThreshold, energyThreshold and minimumDuration. The first parameter represents minimum difference between contiguous samples of the salturated regions. The parameter was swept for the values: [0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1, 5, 10]. And the best results in Fscore were obtained for high values such as 5 or 10, but, if precision was to be priotitized, the smallest value wouyld give the best results.

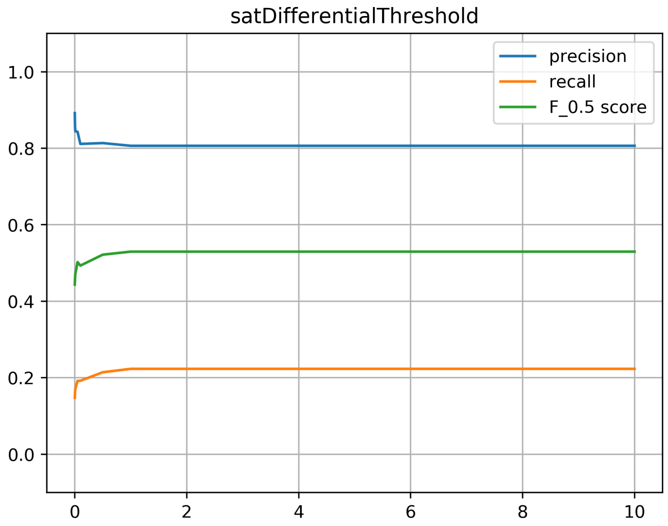

Figure 13: differentialThreshold parameter sweep results (accuracy, F score and recall)

The second parameter is energyThreshold which is the minimum energy for the algorithm to assume a region as saturated. The value was swept for the values: [-30, -20, -10, -7, -5, -3, -1, -0.01] and the best Fscore results were obtainesd for -5 dB.

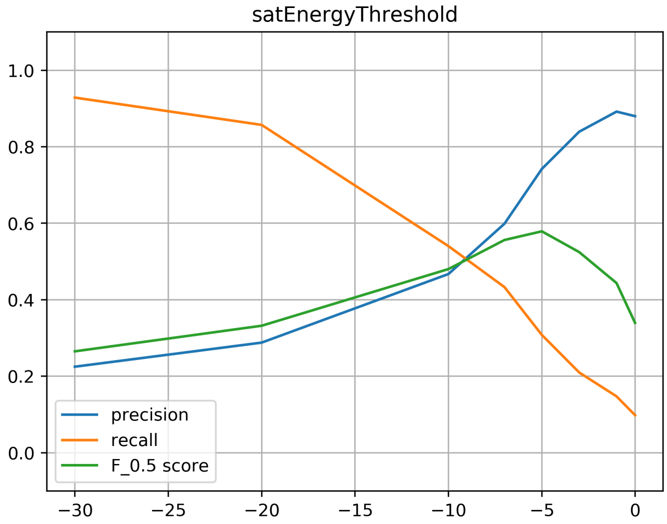

Figure 14: energyThreshold parameter sweep results (accuracy, F score and recall)

The last parameter is minimumDuration, which is the time threshold for the algorithm to detect a saturated region. The parameter was swept for the values: [0.001, 0.005, 0.01, 0.05, 0.1, 0.5, 1] and the best results were obtained for 0.001.

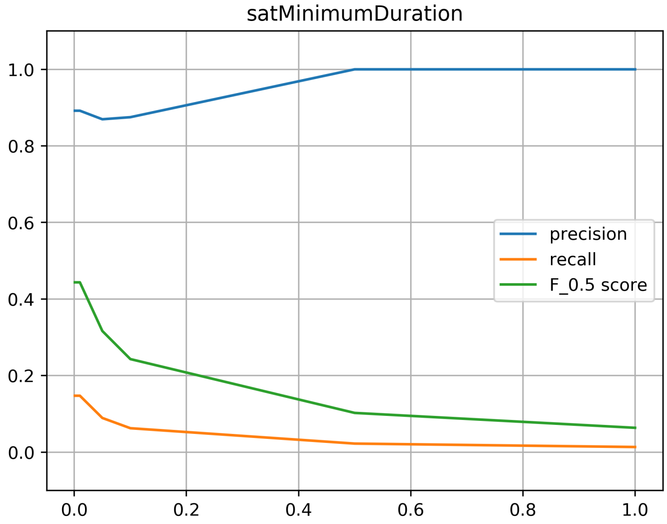

Figure 15: minimumDuration parameter sweep results (accuracy, F score and recall)

4.5 Silence detection evaluation

For the silence detection algorithm, two parameters were evaluated: frameSize and threshold which for both we had simiar results as they were obtained in hum detection meaning that no audios were detected as having an issue.

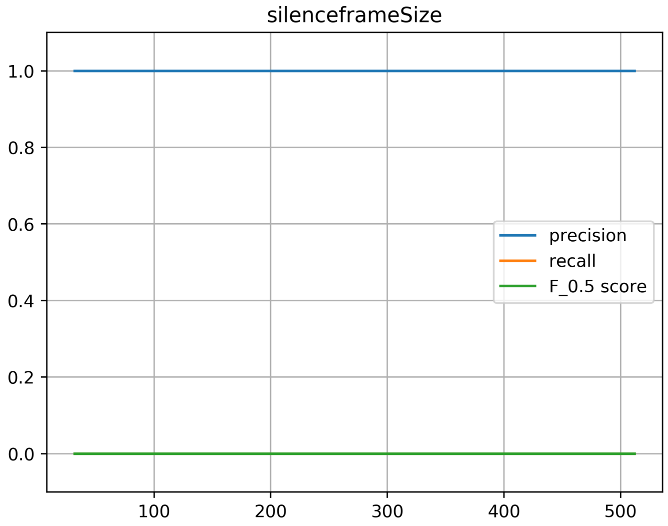

Figure 16: frameSize parameter sweep results (accuracy, F score and recall)

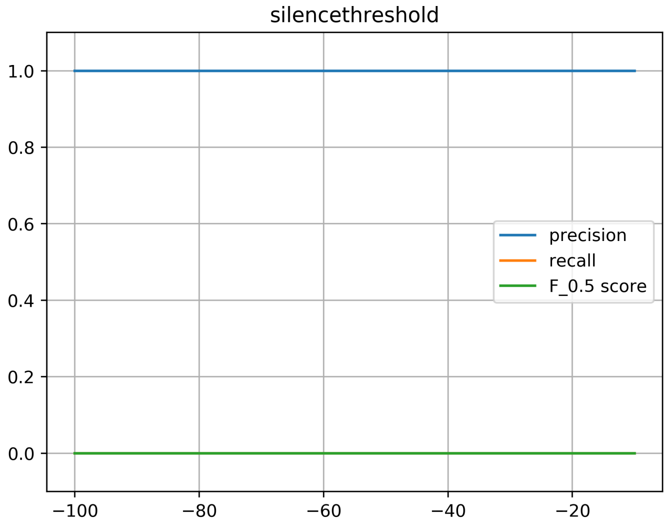

Figure 17: threshold parameter sweep results (accuracy, F score and recall)

Chapter 5

Conclusions

After the grid search results, the project has to conclude with a result that has been not satisfactory and maybe due to the time set to deliver the thesis hasn't been explored as in-depth as it should have been. After seeing the accuracies of all the algorithms being so low and, after a manual re-inspection of the audios that were detected or not as problematic, I can almost certainly state that with a most accurate ground truth (the ground truth was a manual annotation of the author), the results would be more convincing. The algorithms themselves perform fairly well when evaluated in some cherry picked audio files, however, when executed in the whole dataset there were some issues that should be revised in a later revision of the project:

- The whole dataset being mono audio files mean that the phase issues have not been able to be evaluated
- Some algorithms gave unsatifying results such as hum and silence, a further exploration of the algorithms should be done to modify them as the use case for the algorithms and the use case for the project differ considerably
- The annotation for the dataset used in the grid search in the results section did not prove to be a reliable ground truth upon further investigation.

List of Figures

List of Tables

Bibliography