Laughter detection in privacy-sensitive audio

MATTEO FREGONARA
Supervisor(s): HAYLEY HUNG, JOSE VARGAS QUIROS
EEMCS, Delft University of Technology, The Netherlands
20-6-2022

A Dissertation Submitted to EEMCS faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering

Abstract

With the development of new technologies and approaches in the field of social signal processing, concerns regarding privacy around recording conversations have arised. One of the main ways to preserve the privacy of the speakers in recorded conversations consists of decimating said conversations, which consists of reducing the sample frequency and the frequency bandwidth of the audio. This theoretically makes the verbal content of the conversation (the words themselves) unintelligible, while still preserving other useful non-verbal social cues such as laughter, pitch modulation and frequency of speech, amongst others. However, this has not been experimentally verified. This research paper addresses this knowledge gap by exploring the performance of laughter detection machine learning models with decimated audio. An existing pre-trained state-of-the-art laughter detection model was employed and its performance was evaluated for a dataset of decimated audio with sample frequencies ranging from 300Hz to 44100Hz.

1 Introduction

Privacy is starting to become a major concern in the field of social signal processing, especially when it comes to studying audio recordings of conversations, as said conversations might contain sensitive information that the speakers might not want to be made public. Social signal processing refers to the development of automated approaches with the aim of detecting and classifying social cues. Decimation, the process of lowering the sample frequency (also known as the sample rate) and the bandwidth of an audio signal [1, 2], can be used to address the problem of privacy in this field. Figures 1 and 2 respectively illustrate these reductions in sample rate and bandwidth. By using a low enough sample frequency, it is possible to theoretically make speech unintelligible since consonants are not recorded and most words become unidentifiable [3]. In theory, it should still be possible to detect non-verbal social cues such as laughter in the decimated audio. These social cues can in turn help us study the relationship between the speakers while respecting their privacy. However, this has not been experimentally verified. This research paper will specifically focus on the detection of laughter in decimated audio, laughter being an important non-verbal social cue, considered to be "a powerful signal of social acceptance or rejection" [4]. The relevance of studying laughter in the field of social signal processing can further be explained by laughter being an universal [5] and ubiquitous expression of emotion [6, 7]. The research question for this paper is therefore the following: *"How does the reduction in sample frequency hinder the detection of laughter?"*. More concretely, we will aim to answer this question by exploring how the performance of a pre-trained state-of-the-art laughter detection model get affected by decimation, i.e. reductions in the sample frequency. One major point to consider is that laughter detection models are not usually trained on decimated audio.

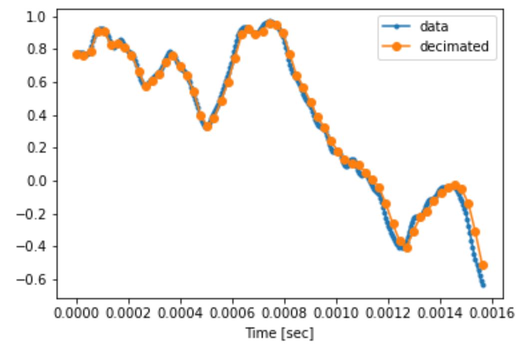

Figure 1: Waveform illustrating the reduction in sample rate which occurs during decimation. The sample frequency was reduced from 44100Hz to 8820Hz (decimation factor of 5).

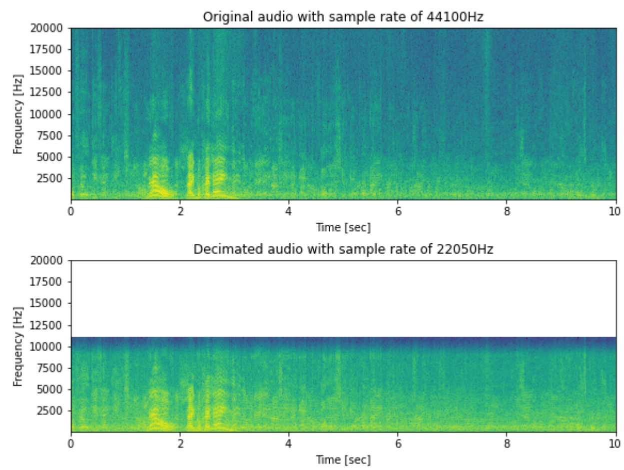

Figure 2: Set of spectograms illustrating the bandwidth reduction which occurs during decimation. The decimated audio is bandlimited at a frequency B, where, for sample rate f_s, $B < \frac{1}{2} f_s$. In this case, $B = 11025Hz$.

The paper is divided in the following sections: section 2 will provide some additional background, while section 3 will describe the methodology employed during the research and the setup of the experiment. Section 4, which will focus on the results, comes next. Section 5 will consist of a discussion of the ethical aspects concerning this research. Section 6 will introduce the main conclusions derived from the experiments. Finally, section 7 will mention what future improvements could be made.

2 Background

This section will strive to provide additional context about recent developments in the field of privacy-sensitive audio recording systems, known as badges, and laughter detection models.

2.1 Privacy-sensitive badges

Efforts have been undertaken in recent years with the goal of developing systems which record audio at a low sample frequency in order to make verbal content unidentifiable while still preserving non-verbal social cues. An example of such a system is MIT media lab's RhythmBadge [8], released back in 2018. By recording audio at a low sampling frequency (700Hz), the badge theoretically does not allow words to be understood. Another such system was developed in 2019, when the TU Delft Socially Perceptive Computing lab carried out a wide scale data collection experiment during the ConfLab [9] conference in Nice, France. A new badge based on the previously developed RhythmBadge was created, named Midge. It recorded audio at a higher sampling frequency than RhythmBadge (1.25KHz instead of 700Hz) in order to compensate for the additional amount of background noise present at the event. However, it has not been experimentally verified with the latest approaches whether using a low sampling frequency would still make the detection of social cues possible.

2.2 Laughter detection models

Laughter detection models can be employed to automatically detect laughter in audio recordings. This automation is of great use when dealing with large amounts of recordings, which is often the case when performing data collection experiments in the field. Much like ASR (Automatic Speech Recognition), the most recently published research about laughter detection employs fully connected deep neural networks (DNN) or convolution neural networks (CNN), as well as Mel-frequency cepstral coefficients (MFCCs) for features. [10–12]. However, there are a few key differences between automatic laughter detection and ASR. Primarily, laughter detection models tend to be more general when compared to speech recognition models and are not concerned with differences in language and accent.

Most approaches when it comes to automatic laughter detection consist of models trained on datasets with a lack of or very little background noise, favouring data extracted from telephone conversations [13] or private meetings [14] with 'clean' audio. Such models undergo a significant drop in performance when attempting to detect laughter in audio featuring a non-negligible amount of background noise [12]. When performing data collection experiments, noise is often present, calling for the need to employ a laughter detection model with some amount of robustness and consistency when presented with noisy input. Fortunately, recent research has produced laughter detection models with robust performance with noisy audio [12]. This will be discussed further in the next section.

3 Methodology

The following section will discuss the methodology of the research. In order to concretely answer the research question, we will begin by exploring one of its main aspect: which laughter detection model to evaluate. Following a discussion on said topic, we will then turn to the problem of choosing a suitable dataset for evaluating the performance of the chosen model. We will then further discuss the evaluation process and the metrics employed to measure performance. We then follow with an argument about why directly decimating the input audio creates complications and we will detail how we instead simulate decimation through low-pass filtering. Finally, we touch upon the various hyper-parameters the model possesses and how we take them into account for our experiment.

3.1 The model

As previously touched upon in section 2, data collection experiments in social signal processing strive to capture human interaction in as much of a natural environment as possible, as opposed to studying said interactions in a controlled environment. This can often result in the presence of background noise. Choosing a laughter detection model with some degree of resilience to noise in the input audio is therefore imperative if we want to achieve the best possible performance. There do not exist many models designed specifically to provide good performance with noisy audio, but Gillick et al. do present us with a robust state-of-the-art model which suits our needs [12]. The model is stated to have achieved a F1-score of 0.75 when evaluated on clean audio, and a score of 0.608 with noisy audio. The source code is available on GitHub [15]. The model itself consists of a residual network employing Mel spectogram features, trained on the Switchboard dataset [13] and featuring dataset augmentations (i.e. noise manually added to the data). A particularity of this system is the reduction of sample frequency it performs on its input audio, reducing the sample rate from the standard 44.1kHz to 8kHz before feeding it to its residual network. The model was also trained with audio at that specific sample frequency.

3.2 LaRed dataset

The LaRed dataset contains a total of 27 hours of conversations, recorded at a networking event. 35 participants were involved, each wearing a custom badge which not only recorded audio but also video and data about the acceleration and motion of the individual wearing it. This data collection experiment was carried out by TU Delft Socially Perceptive Computing lab. We will employ this dataset for evaluating the model. One important point to consider about this dataset is that the conversations were not captured in a privacy-sensitive manner, i.e. the audio was not decimated with the aim to protect the anonymity and privacy of the participants. Instead, the dataset was recorded at a standard sample rate of 44.1KHz. This will be further touched upon in section 5, where the ethical implications of the research are discussed. It is relevant to note that a dataset consisting of privacy-sensitive audio would have been inadequate for the purpose of this research since laughter annotations are required in order to evaluated the performance of the model. Said annotations were made manually, consisting of timestamps indicating the position of laughter in the audio (start time, end time, duration). It is important to observe that what can be classified as laughter can sometimes be considered ambiguous (eg: can a short snicker be considered a laugh?), and an argument can be made about said annotations being subjective. We will elaborate on this particular point in section 7.

3.3 Evaluation process

Having discussed both the model and the dataset, we now proceed to address the evaluation process, where we assess the performance of the model for sample frequencies ranging from 300Hz to 44100Hz. Due to time constraints, for each sample frequency, we only run the model on 10h of audio extracted from the LaRed dataset, from 10 different participants, containing 355 instances of laughter in total. The audio was pre-processed in order to remove head and tail silence. The model returns timestamps detailing the start and end time for each occurrence of laughter detected, which can then be compared to the laughter annotations of the LaRed dataset. An occurrence of laughter is considered a match if the timestamps produced by the model overlap any timestamp present in the annotations. More formally, for a instance of laughter I with start time s and end time e, we consider I to be a true positive if, for the set A of laughter annotations with start times s_a and end times e_a:

$$\exists a \sum_{a}^{A} s_a < s < e_a \vee s_a < e < e_a$$

Due to the ambiguity which comes when trying to detect laughter, it is sometimes difficult to exactly pinpoint the start and end of a laugh. Hence, we make the choice of looking for overlaps between our results and the annotations, instead of trying to precisely match start and end times, since we are more interested in evaluating whether the laugh was detected or not, and not on wheteher the start time and end times match precisely. This relaxation in the precision of the matching between the output of the model and the annotations is reflected in our use of $\vee$ (OR) in the formula above, instead of potentially using $\wedge$ (AND). We then employ various metrics to assess the performance of the model for each sample frequency. These are a standard suite of metrics common to classification problems, consisting of precision, recall and the F1 score, as outlined below:

- $Precision = \frac{TP}{TP+FP}$
- $Recall = \frac{TP}{TP+FN}$
- $F1 = 2 * \frac{Recall*Precision}{Recall+Precision}$

3.4 Simulating decimation through low-pass filtering

Decimating the input audio for each of the sample frequencies we wish to evaluate and feeding it to the model would be the logical approach to take in order to evaluate the performance of the model. However, this results in complications. The model was trained with a dataset of audio with a sample rate of 8kHz, and we cannot use input audio with a different sample rate without suffering the risk of reduced performance due to not fully exploiting the model's residual network, fitted for a sample rate of 8kHz. One possible solution would be to re-train the model for each sample frequency, therefore fitting the residual network to the appropriate sample rate each time, but this solution results in a drastic increase in training time and cannot be achieved in the scope of the research. Instead, we will attempt to simulate the process of decimation by applying a low-pass filter to the input audio and leaving the sample frequency unchanged. Decimation, as stated in section 1, refers to the process of reducing the bandwidth of an audio signal and lowering its sample frequency. By low-passing the input audio for the model, we replicate the bandwidth-reducing aspect of decimation, and therefore obtain an approximation of the process of decimation. The full implications and consequences of this approximation will be discussed in more detail in sections 6 and 7.

In more specific terms, we first filter the input audio using a 8-order Butterworth low-pass filter, implement with the Python scipy library [16]. The choice of this specific filter reflects scipy's own implementation of decimation [17], where an 8-order IIR low-pass filter is employed. The cutoff frequency f_c for the low-pass filter is determined by the use of the Nyquist criterion, where an audio signal at a specific sample frequency f_s is bandlimited at a frequency B, where $f_c = B < \frac{1}{2} f_s$. Once we obtain the low-passed audio signal, we reduce its sample rate from 44.1kHz, the original sample rate of the LaRed dataset, to 8kHz, the sample rate employed by the model. The full process is summarised in Figure 3. Figure 4 shows a comparison of the frequency spectrums between a decimated snippet of audio from the LaRed dataset and the same audio signal with an equivalent low-pass filter applied to it. As can be observed, the spectograms show similarities, but the reduction in samples can be clearly observed in the more coarse-grained representation of the decimated audio.

Figure 3: Full process for simulating decimation and feeding input audio to model

3.5 Model hyper-parameters

The model features a total of two hyper-parameters, both of which are relevant for our experiment: *min_length*, which determines the minimum length in seconds that a laugh needs to last in order to be detected and *threshold*, which "adjusts the minimum probability threshold for classifying a frame as laughter" [15]. The *threshold* parameter can be set between 0 and 1, and a lower value might result in more false positives but also in a higher percentage of laughter being recovered from the audio. In a similar fashion, lowering the value of the *min_length* parameter might result in more short instances of laughter being recovered, at the expense of an increase in false positives.

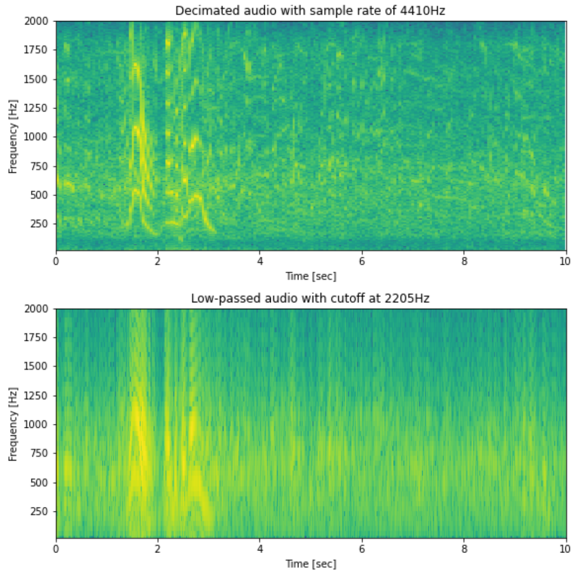

Figure 4: Set of spectograms comparing decimation and low-pass filtering. A 10s snippet of audio from the LaRed dataset was employed, and was decimated to 4410Hz on one hand and low-pass with cutoff frequency 2205Hz on the other.

For the value of *min_length*, we made the choice of 0.5s. Figure 5 illustrates the lengths of laughter instances in the LaRed dataset, where 85% of laughs have a length longer than 0.5s. We deemed this to be a suitable trade-off between recovering less false positives and potentially missing out on detecting the 15% of the dataset consisting of laughter instances with a length shorter than 0.5s.

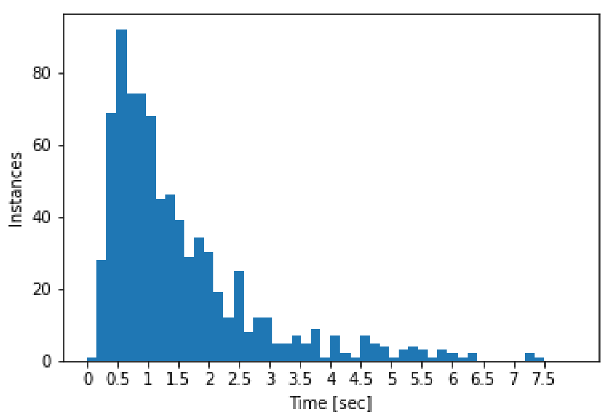

Figure 5: Histogram illustrating the length of laughter instances in the LaRed dataset.

For the *threshold* parameter, we instead chose to evaluate a range of values between 0.5 and 0.8, in order to better explore the aforementioned compromise between false positives and more laughs being recovered. This specific range of values was determined by preliminary tests carried

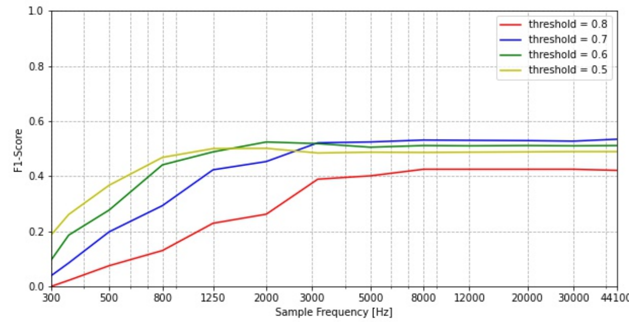

Figure 6: F1-scores of the model for the various values of the *threshold* hyper-parameter.

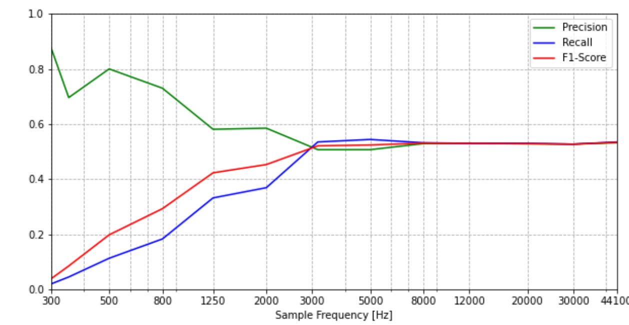

Figure 7: Performance of the laughter detection model on subset of LaRed dataset, using a value of 0.7 for the *threshold* hyper-parameter. Precision, Recall, and F1 scores are given.

out on a limited subset of the LaRed dataset. The data present in this subset does not overlap with the data used for evaluating the model, which help present any biases. These tests explored the impact of using different values for this specific parameter, and showed that this range of values for the *threshold* produced the best results.

4 Results

Overall, the performance of the model was found to be similar to that stated by Gillick et al., achieving a F1-score of 0.534 in the best-case scenario, when using a sample rate of 44100HZ and a *threshold* of 0.7. Figure 6 presents the F1-scores obtained for the various values of the *threshold* parameter. For high sample frequencies, the scores remain virtually unchanged, but as we progress down the sample frequency range, an almost linear drop in performance can be observed from sample frequency f_d. This specific sample frequency appears to vary according to the *threshold* parameter. For a *threshold* of 0.8, $f_d = 8000$Hz, as we first start to observe a decrease in the F1-score around 8000Hz. For a *threshold* of 0.7, $f_d = 3150$Hz. For a *threshold* of 0.6, $f_d = 2000$Hz. And finally, for a *threshold* of 0.5, $f_d = 1250$Hz. This positive relation between the *threshold* value and f_d appears to be caused by a lower *threshold* resulting in more potential laughs being recovered, regardless

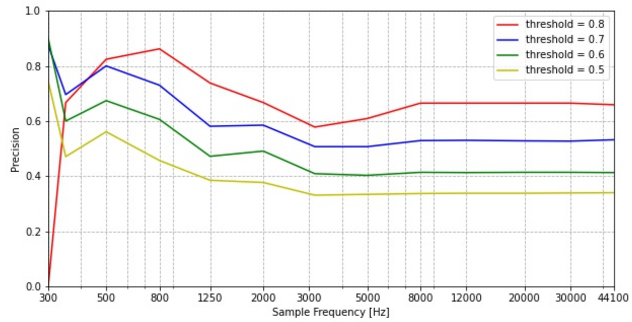

Figure 8: Precision scores of the model for various values for *threshold* hyper-parameter.

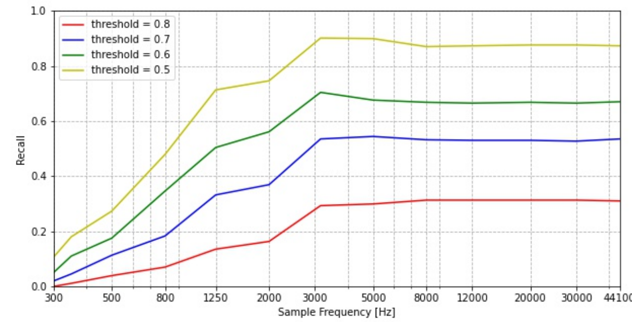

Figure 9: Recall scores of the model for various values for *threshold* hyper-parameter.

of whether laughs are less likely to be true positives. As the sample frequency decreases, laughs which were previously detected by the model might now not be recovered anymore if we use a high *threshold*, while a lower *threshold* value does allow for it. This overall reduction in performance as we lower the sample rate can be said to be an expected outcome, seeing how speech recognition models, which employ similar approaches to laughter detection, are also hindered by a reduction in sample frequency, as discussed in sections 1 and 2. The *threshold* parameter appears to play an important role in how this decrease manifests, and will further be explored later in this section.

It is interesting to observe the effect the reduction of sample frequency has on both precision and recall. Figure 7, where we present the various metrics for a *threshold* value of 0.7, illustrates this point very clearly. While precision and recall are equal for sample rates higher than f_d (for a *threshold* of 0,7, $f_d = 3150$Hz), it is possible to observe an almost linear increase in precision occurring as the sample frequency decreases from f_d downwards, and a corresponding decrease in recall, since the two metrics are inversely related. This can be interpreted as a lower sample rate resulting in less laughs being detected, but the retrieved laughs are more likely to be true positives. This phenomenon occurs for all employed values of the *threshold* parameter, as can be observed in Figure 8, which compares the precision for the different values of the *threshold*, and Figure 9, which presents the recall scores.

Another correlation which can be observed in both figures is the relationship between the *threshold* and the precision (and recall) of the model: a higher value for the *threshold* results in less false positives being recovered, therefore resulting in a higher precision score. This can very clearly be observed in Figure 8. The value for the *threshold* parameter and the precision are therefore positively related (while the *threshold* value and the recall are negatively related). This specific relation has further implications on the distribution of the F1-score over the sample frequency for different values of the *threshold* parameter, as shown in Figure 6. A higher *threshold* value appears to result in a lower F1-score for sample frequencies higher than high, but also in a higher score for frequencies lower than f_d, where the *threshold* appears to affect the gradient of the F1-score curve for a sample rate lower than f_d. This can be explained by the increase in false positives due to a higher *threshold* value for sample frequencies lower than f_d being less significant when compared to sample frequencies higher than f_d. This can clearly be noted with *threshold* values of 0.5, 0.6 and 0.7. However, too high a value seems to result in an overall loss of performance along the whole range of sample frequencies, as shown by the F1-score for a *threshold* value of 0.8.

5 Responsible Research

The are various points to consider when it comes to the ethical implications of this research. First and foremost, the privacy of the participants which contributed to the LaRed dataset must be respected. Secondly, reproducibility should be taken into account, as part of the guidelines of valid academic research.

5.1 Privacy of LaRed dataset contributors

Respecting the privacy of the participants of the LaRed dataset data collection experiment was of upmost importance when carrying out the research, especially since the audio was not recorded in a privacy-sensitive manner. No data pertaining to the dataset was shared in any way or uploaded to the cloud, and all activities relating to processing the data and evaluating the performance of the model were run on a local machine.

5.2 Reproducibility of research

The research was carried out in such a fashion to allow for it to be reproduced. As previously mentioned in section 3, the source code for the laughter detection model employed is available on GitHub [15], and the methodology as well as the setup for the experiment was discussed in a detailed manner: the aforementioned model, its hyper-parameters, the dataset employed and the evaluation process were all discussed and explained.

6 Discussion and Conclusions

The research on the detection of non-verbal social cues such as laughter in privacy-sensitive audio is still ongoing, and it still has not been experimentally verified how decimation affects the detection of laughter. This research paper attempted to address this knowledge gap by exploring how the performance of a pre-trained state-of-the-art laughter detection model get affected by decimation. We simulate decimation with the use of low-pass filtering, and evaluate the performance of the model.

The experiments we carried out showed that decreasing the sample frequency negatively impacts the performance of the model, resulting in a relatively linear decrease in performance after a specific sample frequency f_d. The *threshold* hyper-parameter of the model, which configures the model to recover more laughter instances while taking the risk of increasing the amount of false positives, influences the value of f_d, as well as its performance for sample rates lower than f_d and higher than f_d.

The research featured various limitations which pave the way for further research and experiments, the most important limitation being the use of low-pass filtering to simulate decimation. A drawback of this approximation is that we miss out on the effects that a reduction in sample frequency provides on the detection of laughter, and therefore we do not explore the full impact of decimation on laughter detection. This particular limitation and others will be discussed further in the next section, as well as the future research that could be carried out to address them.

7 Future Work

As mentioned in the previous section, a possible follow-up to this research would be to re-train the model for each sample rate and then evaluate its performance, in order to better explore and understand the full impact of decimation on laughter detection models, rather than only considering the bandwidth reduction aspect of it. The reduction in sample rate might further hinder the detection of laughter, since the presence of less data samples could make laughter detection more difficult, as well as potentially resulting in a higher number of false positives.

Another possible point of improvement would to evaluate the performance of different laughter detection models, especially if future research produces more models with a robust performance with noisy data. Similarly, different datasets could be employed, captured using varied audio recording equipment and with different levels and types of background noise, in order to obtain a more general idea of the performance of laughter detection models with low sample frequency audio, in contrast with the rather preliminary nature of the results obtained in this research.

Additionally, when it comes to selecting the best hyper-parameters for the model, hyper-parameter tuning could have been employed, especially in the case of the *threshold* parameter, where it would have allowed us to more precisely obtain an optimal value for the parameter. Additionally, as shown in section 4, a lower value for the *threshold* resulted in better performance for sample frequencies under f_d, at the cost of performance at higher sample rates. Therefore, one could perform tuning with the goal of maximizing performance with low sample rates rater than maximizing performance over the whole range of sample rates, resulting in a tuned model with a greater performance with low sample frequencies.

Furthermore, an important point to consider about laughter detection in general is that is always ambiguity when it comes to annotating laughter. There are different types of vocalizations (eg: snickers, chuckles, giggles) which some individuals might consider to be a laugh, while other might not. The annotations provided with the LaRed dataset were taken in a way to include all vocalizations mentioned above, classifying them as laughter. A way to obtain more objective annotations would be to have several individuals annotate the dataset and to then derive a final set from their combined annotations.

Finally, fine-tuning can be employed to further attune the laughter detection model to the dataset, initialising a new model with the current model's weights and doing some further training with the current dataset. This might help further improve performance, potentially resulting in more robust laughter detection at low sample frequencies.

References

[1] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, Discrete-time signal processing, 2nd ed. Upper Saddle River, N.J: Prentice Hall, 1999.

[2] EETimes, 'EETimes - Multirate DSP, Part 1: Upsampling and Downsampling', EETimes, Apr. 21, 2008. https://www.eetimes.com/multirate-dsp-part-1-upsampling-and-downsampling/ (accessed Jun. 02, 2022).ation. The term decimation used for the down-sampling process has been accepted and used in many textbooks and fields.

[3] J. Shen, O. Lederman, J. Cao, F. Berg, S. Tang, and A. Pentland, 'GINA: Group Gender Identification Using Privacy-Sensitive Audio Data', in 2018 IEEE International Conference on Data Mining (ICDM), Nov. 2018, pp. 457–466. doi: 10.1109/ICDM.2018.00061.

[4] J. Ritter, C. Brück, H. Jacob, D. Wildgruber, and B. Kreifelts, 'Laughter perception in social anxiety', Journal of Psychiatric Research, vol. 60, pp. 178–184, Jan. 2015, doi: 10.1016/j.jpsychires.2014.09.021.

[5] D. A. Sauter, F. Eisner, P. Ekman, and S. K. Scott, 'Cross-cultural recognition of basic emotions through nonverbal emotional vocalizations', Proceedings of the National Academy of Sciences, vol. 107, no. 6, pp. 2408–2412, Feb. 2010, doi: 10.1073/pnas.0908239106.

[6] R. R. Provine and K. Emmorey, 'Laughter Among Deaf Signers', The Journal of Deaf Studies and Deaf Edu-

cation, vol. 11, no. 4, pp. 403–409, Oct. 2006, doi: 10.1093/deafed/enl008.

[7] R. R. Provine, 'Laughing, Tickling, and the Evolution of Speech and Self', Curr Dir Psychol Sci, vol. 13, no. 6, pp. 215–218, Dec. 2004, doi: 10.1111/j.0963-7214.2004.00311.x.

[8] O. Lederman, A. Mohan, D. Calacci, and A. S. Pentland, 'Rhythm: A Unified Measurement Platform for Human Organizations', IEEE MultiMedia, vol. 25, no. 1, pp. 26–38, Jan. 2018, doi: 10.1109/MMUL.2018.112135958.

[9] 'ConfLab – ACM MM 2019'. https://conflab.ewi.tudelft.nl/ (accessed May 16, 2022).

[10] M. Knox and N. Mirghafori, 'Automatic laughter detection using neural networks', Aug. 2007, pp. 2973–2976. doi: 10.21437/Interspeech.2007-741.

[11] L. Kaushik, A. Sangwan, and J. Hansen, 'Laughter and Filler Detection in Naturalistic Audio', Sep. 2015. doi: 10.21437/Interspeech.2015-541.

[12] J. Gillick, W. Deng, K. Ryokai, and D. Bamman, 'Robust Laughter Detection in Noisy Environments', in Interspeech 2021, Aug. 2021, pp. 2481–2485. doi: 10.21437/Interspeech.2021-353.

[13] Godfrey, John J. and Holliman, Edward, 'Switchboard-1 Release 2'. Linguistic Data Consortium, p. 14610176 KB, 1993. doi: 10.35111/SW3H-RW02.

[14] A. Janin et al., 'The ICSI Meeting Corpus', 2003.

[15] J. Gillick, laughter-detection. 2022. Accessed: Jun. 02, 2022. [Online]. Available: https://github.com/jrgillick/laughter-detection

[16] 'SciPy documentation — SciPy v1.8.1 Manual'. https://docs.scipy.org/doc/scipy/index.html (accessed Jun. 02, 2022).

[17] 'scipy.signal.decimate — SciPy v1.8.1 Manual'. https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.decimate.html (accessed Jun. 02, 2022).