Bachelor's Thesis Computing Science

Radboud University Nijmegen

Filtered Feelings: Investigating Frequency Filters in Speech Emotion Recognition Models

---

Author:
Teun van Gisteren
s1055104

First supervisor/assessor:
dr. L.F.M. ten Bosch

Second assessor:
Prof. T.M. Heskes

July 4, 2024

Abstract

Speech Emotion Recognition (SER) deals with the automatic classification of emotion from the speech signal. SER is a growing field with applications ranging from human-computer interaction to entertainment, and security. Research shows that emotion classification depends on multiple features in the audio signal like pitch and speaking rate, and on the verbal content of the speech signal, the contents of your speech. The question we address is whether different emotions depend in various ways on the features in the audio signal. We investigate the influence of frequency filtering on SER model performance. We propose a method for testing the impact of frequency-filtered audio on an SER model and identify the onset of misclassifications and the boundaries between different emotion classes. We also developed a visualization technique to analyze these results. Our findings reveal that frequency filtering significantly affects SER models, with varying impacts on different emotions.

Contents

Chapter 1

Introduction

Speech Emotion Recognition (SER) is a field focused on recognising emotions conveyed through speech. Recently, this field has increasingly gained more attention due to its numerous potential applications in a variety of different fields such as human-computer interaction, entertainment, or customer service. There are also cultures, such as the Japanese and Korean, where emotion is an important social cue that can force the use of different politeness levels in conversations. In these cultures, it is extremely important to be able to recognise emotions, for humans and machines alike. Understanding the emotions of the user and being able to respond to them appropriately could greatly enhance the capabilities of systems like AI assistants, enabling them to respond appropriately to users' emotional states.

Humans can infer emotions from speech signals. We can identify emotions from speech signals by processing them in two different tiers. The verbal ("what has been said") tier allows us to process the contents of the speech signal. Certain words and phrases can indicate certain emotions, for example, if someone says "I'm so excited" that conveys happiness, while "I'm fuming" is a clear indication of anger. The prosodic tier ("how it was said") takes into account features such as pitch, variation in pitch, speaking rate, etc. Emotions are dependent on specific cues in the audio signal. In this thesis, we address to what extent emotion classification depends on modifications to the audio, which is instrumental for finding out whether different emotions depend on different audio cues.

The suggestion that different emotions differ in the dependence on different cues may become clear when we give an example, like an angry neighbour. If you can hear them through your wall, it is clear that they are angry without being able to recognise any of the verbal contents of their speech. This emotion recognition lies clearly in the prosody of the audio, filtered through the wall, but still providing enough cues to be properly recognised as anger. The question then becomes to what extent other emotions like

happiness or sadness are also robust against filtering techniques, comparable to wall-like damping and filtering effects of audio signals.

Current implementations of SER are often based on end-to-end approaches, where the performance of a model is heavily dependent on the quality of the input data and the processing techniques employed. Datasets such as IEMOCAP [1], a widely used resource in SER research, are recorded with high-quality microphones in controlled, quiet environments. While this results in excellent training data, it does not accurately represent the variability and noise found in real-world audio recordings. In real-world scenarios, audio can also be transformed, such as frequency filtering in telephone calls to save bandwidth. This filtering may inadvertently remove frequencies crucial for accurate emotion classification by SER models. In this thesis, we look at the impact of frequency filtering on the classification performance of SER models. Specifically, we investigate how the application of frequency filters affects classification accuracy when analyzing emotions in speech.

This research aims to understand how limited frequency ranges affect the ability of an SER model to differentiate emotions and to identify the thresholds at which classification errors occur. We explore to what extent emotions rely on different speech signal frequencies, for instance, whether joy is associated with higher frequencies and sadness with lower ones.

We approach this issue by creating a testing pipeline, with which any SER model can be analysed on their performance on frequency-filtered audio. We then create visually informative heat maps to help illustrate performance trends in the model and highlight its limitations.

This thesis is structured as follows: Chapter 2 will start by explaining some of the needed background knowledge to understand this thesis. Chapter 3 will discuss the current state of SER and earlier attempts at improving performance on real-world audio. Chapter 4 will then detail our main testing pipeline and how it works at a high level, after which Chapter 5 will explain in detail the experiments that were performed. Chapter 6 presents the findings of our experiments. In Chapter 7, we will discuss our findings, limitations of this study, and avenues for future research. Finally, Chapter 8 concludes our research and summarize key insights.

Chapter 2

Preliminaries

2.1 Speech Emotion Recognition Models

Speech Emotion Recognition (SER) is a field of study in which the primary goal is to detect and classify the emotions of a speaker through their speech. An SER model typically consists of two distinct parts:

1. **Feature Extraction**: This step involves extracting the relevant features from the speech signal to be able to capture the emotional content of the audio. Commonly extracted features include:

   - Prosodic Features: These features are related to how speech sounds. These include pitch (fundamental frequency), energy (intensity), and speaking rate (tempo).
   - Spectral Features: These features analyse the frequency spectrum, such as formants (frequency peaks in the spectrum with a high degree of energy), mel-frequency cepstral coefficients (MFCCs), and linear predictive coding (LPC) coefficients.
   - Temporal Features: Capture the timing aspects of speech, like pauses and duration.

   Apart from these features, other things like variance and rate of change of these measures are also important.

2. **Emotion Classification**: The extracted features are then to be used to either train an AI model or to be classified by one. Methods to train these models can include more traditional machine learning algorithms like Support Vector Machines, Hidden Markov Models, or k-Nearest Neighbors, but also deep learning models like Convolution or Recurrent Neural Networks.

2.2 Human Speech

Human speech is, just like any sound, an audio signal. Speech signals have various properties that play a role in communication. Some examples of these properties include:

- **Frequency**: According to the source-filter theory of speech production [2], the pitch (also called fundamental frequency) is related to the vibration of the glottis, the center of the voice box, and determines high and low voices, and gender to a large extent. The timbre is related to the shape of the filter, for example, the vocal oral and nasal cavity. The timbre determines the differences between vowel-like speech sounds such as /a/, /i/ and /u/. The timbre is related to the distribution of energies as a function of frequency in the so-called spectrum. A spectrogram is a spectrum varying over time. Pitch is a very useful feature in emotion classification, as it seems to play a big part in conveying emotion [3].
- **Speed, speaking rate**: The rate at which speech is delivered influences its perception and understanding. Variations in speech speed can convey emphasis, urgency, or clarity.
- **Volume**: The intensity or loudness of speech affects its perceived importance or emotion. Volume variations can convey nuances such as anger, excitement, or emphasis.

These features are essential to understanding the meaning and emotional context of spoken language, which is critical for communication, but also for automated speech analysis systems like SER models.

2.3 Audio Filtering

2.3.1 The Basics of Audio Filtering

Audio filtering is a signal processing technique with which you can modify the frequency content of audio signals. It involves amplification and attenuating the signal to achieve the desired characteristics. Audio filters can enhance or suppress certain frequencies, making them very useful for this study.

Some of the most commonly used filters in audio processing are:

- **Low-Pass Filter**: The low-pass filter is a filter that does not affect the lower frequencies of the audio while lessening the higher frequencies past the specified cut-off point.
- **High-Pass Filter**: The high-pass filter is a filter that does the opposite of the low-pass filter, where it reduces the intensity of the lower

frequencies while the higher frequencies above the cut-off point are unaffected.

- **Band-Pass Filter**: A band-pass filter is a combination of both a low-pass and a high-pass filter. It lets the frequencies between two cut-off frequencies pass while lessening the frequencies outside this range.

The plots shown in Figure 2.1 show the different frequency responses of the filters mentioned above. The low-pass filter allows frequencies below 300 Hz to pass through while attenuating higher frequencies. The high-pass filter allows frequencies above 3400 Hz to pass through while attenuating lower frequencies. The band-pass filter allows frequencies between 300 and 3400 Hz to pass through while attenuating frequencies outside this range.

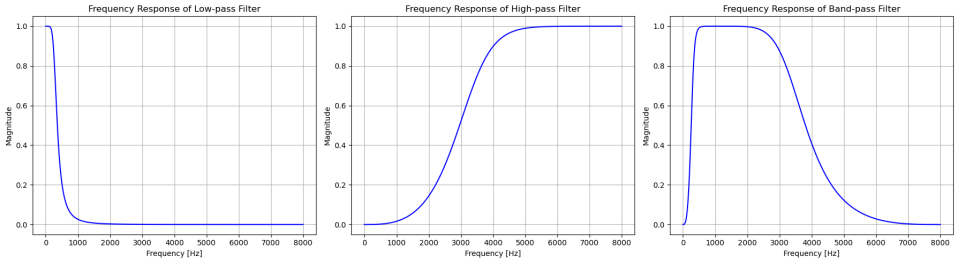

Figure 2.1: Frequency Responses of Low-pass, High-pass, and Band-pass Filters

2.3.2 Audio Filtering in Real-Life Scenarios

There are a multitude of reasons why you would use an audio filter in the real world. Some of these are:

- **Noise Reduction**: A low-pass filter could for instance be used to eliminate some high-frequency noise that is present in your audio, as described in [4].
- **Communication Systems**: When using any form of telecommunications, audio filtering is often used to optimize bandwidth usage. Tradition telephones use the narrowband frequency range of 300 Hz to 3400 Hz, as specified in the G.711 standard [5]. Newer systems like Voice over Internet Protocol (VoIP) can transmit a significantly larger range, 50 Hz to 7000 Hz, as specified in the G.722 standard [6]

There are a lot of different effects and types of noise that can be applied to audio to alter it, but frequency filtering specifically is interesting because it is a well-known technique applied in a multitude of different fields and the ability of filters to emulate real-world acoustic effects. For example, applying a low-pass filter can make it sound like audio is coming from another

room. While audio filtering is very useful in those cases, when it comes to communication systems, the filtered frequencies could have an impact on the performance of an SER model. Filtering alters the spectral characteristics of speech signals, potentially affecting extracted features used in SER. For instance, a band-pass filter could emphasize or suppress specific frequency bands relevant to emotion recognition, which could then enhance or degrade classification accuracy.

Chapter 3

Related Work

The field of SER encompasses a wide range of studies focusing on developing new models, feature sets, and reviewing existing methodologies to identify gaps in knowledge. While SER as a field has existed for multiple decades, deep learning has significantly advanced SER. This has enabled the development of end-to-end AI models that automatically process and classify emotional states from speech.

Morais et al. (2022), for example, introduced a novel end-to-end system based on a modular upstream and downstream architecture paradigm [7]. They use self-supervised learning for feature extraction followed by simple aggregators and classifiers to identify emotion. Using the IEMOCAP dataset, they managed state-of-the-art performance with their model.

Although no studies directly investigate the impact of frequency filtering on SER models, having real-world audio quality possibly interfere with the performance of an SER model is not a new idea. Tawari, A., & Trivedi, M. (2010). proposed a framework for adaptive noise cancellation as preprocessing for SER models and a new feature set to improve performance [8]. Using their own recorded dataset LISA-AVDB, which includes noise, and the EMO-DB database [9] they show promising results for improving emotion recognition in noisy conditions.

Wani et al. (2021) gave a comprehensive review of the state of SER that covers datasets, methodologies, and other state-of-the-art insights [10]. It also provides some knowledge into challenges that the field faces like defining the meanings of each emotion, recording authentic emotional speech, labeling integrity, but also handling noise involving convolute distortion and meddling speakers. They show that the issue of unreliable data, like frequency-filtered audio, is still very much a problem to be solved.

While progress has been made in the development of SER models and handling noisy environments, the specific impact of frequency filtering on SER performance remains underexplored. This thesis aims to fill this gap by investigating how frequency filtering affects SER model performance.

Chapter 4

Methodology

Here we outline the general methodology for evaluating the performance of an SER model given a dataset.

4.1 Overview of the Testing Pipeline

The testing pipeline is designed to systematically determine how the performance of an SER model is impacted by filtering out different frequency ranges. This is achieved by applying various band-pass filters to the audio data and observing the impact on the model's performance.

The pipeline begins with the selection of an appropriate SER model and dataset for testing. Once the model and dataset are selected, the next step is to define the range for the band-pass filters. ss filters. Three parameters need to be set: the minimum and maximum frequency for the band-pass filter and the step size.

After the band-pass filter settings have been determined, We systematically filter the audio for each frequency range and then the model classifies this filtered data. The results of each test are stored separately for later analysis.

The final step involves generating heat maps to visualize the performance of the SER model for each frequency range. This allows for the identification of trends or specific emotions that the model struggles to recognize when a certain frequency filter is applied.

4.2 Step-by-Step Procedure

4.2.1 Step 1: Model and Dataset Selection

The first step is to select the SER model that you want to use for testing. Then it is important to select one or more appropriate dataset(s), for example, the one it was trained on, to perform the tests on. Applying this

process to multiple datasets can help verify that the results are specific to the model and not the datasets chosen. If necessary, ensure that the dataset is preprocessed to suit the needs of your model.

4.2.2 Step 2: Determining Band-Pass Filter Settings

Then, the next step is defining the range of the band-pass filters by specifying the minimum and maximum band sizes. It is important that while making this decision to be aware of what you are trying to accomplish. If you want to know what the overall performance is, you can use the whole available frequency range. If you only need to know up to specific ranges, it is useful to limit your minimum and maximum to not waste resources on unnecessary tests.

Determining the step size is also an important part of the process. The bigger the step size, the less precise you can interpret your final performance data. Choosing a step size that is too small will increase the amount of tests that need to be performed, but give more precise detail on specific ranges. We recommend starting with a step size of around 200 Hz. It is always possible to later run more tests in a more specific range at a higher fidelity to get a clearer view of the performance trend.

4.2.3 Step 3: Applying Band-Pass Filters & Running the Tests

After determining the band-pass filter settings, the next step is to apply these filters to the dataset(s). For each combination of minimum and maximum frequencies, as determined by the step sizes, filter the audio data accordingly. Each filtered version of the audio data is then to be used to test the SER model. Store the results of each test in a separate file for easy access later.

4.2.4 Step 4: Visualize the Results and Analyze the Performance Trends

The final step is to generate heat maps to visualize the performance of the SER for every frequency range. For every pair of emotions that the model can classify, a heat map is created. Every different frequency range will be represented as a point on the graph, with the colour representing how often it misclassified one emotion for the other.

From this, we can then identify certain trends or emotions that the model is having difficulty recognizing when a frequency filter of a certain size is applied.

Chapter 5

Experiments

5.1 Setup

For the following experiments, we used a variety of software tools to preprocess the data, train the model, and evaluate its performance. All the coding and tests were done within Jupyter Notebook and written in Python. The following external libraries were used:

- **SciPy** [11]: Used for the filtering of the audio.
- **NumPy** [12]: For mathematical operations during the filtering.
- **scikit-learn** [13]: Used for generating confusion matrices and calculating the accuracy score of the model.
- **matplotlib** [14]: Used for plotting heat maps among other graphs.

The notebooks containing the experiments can be found in Appendix A.

5.2 The SER Model

The model that was used to perform the experiments is the emotion-recognition-wav2vec2-IEMOCAP model by SpeechBrain [15]. SpeechBrain is an open-source toolkit for conversational AI made for Python. This specific model was made by the SpeechBrain team as an example of how to create an AI model with their tools.

Their model is based on the wav2vec2 [16] model. wav2vec2 is a model developed by Facebook AI for automatic speech recognition purposes. It is capable of processing raw waveform files directly, instead of relying on extracted features like the Mel-frequency cepstral coefficients to extract meaningful representations of speech. This allows wav2vec2 to learn directly from the raw audio data, employing self-supervised learning techniques allowing it to encode high-level features that are useful for speech recognition tasks.

As input, the model takes WAV files. These are then processed using wav2vec2, after which the embeddings, numerical representations of complex objects, are extracted using attentive statistical pooling. These embeddings are then passed through a linear layer that consolidates them into four neurons. Following this, a softmax function generates a probability distribution across the model's four potential outcomes: 'hap', 'sad', 'neu', and 'ang'. These outcomes are subsequently encoded into more interpretable labels. The output of the model becomes the probabilities for each outcome, a score, the index of the best class, and the text labels of the outcomes.

The model is trained on recordings with a sampling rate of 16 kHz in a single channel, which is the file size it accepts as input. Any audio input will automatically be normalized and converted to these parameters if needed.

5.3 The Dataset

The primary dataset used during these experiments was the IEMOCAP dataset [1]. This dataset was chosen because it was used to train the model that was being tested and is one of the most used datasets for SER currently.

# of Total Utterances	10039	# of Annotators	3
# of 'sad' Utterances	1103	# of Speakers	10
# of 'happy' Utterances	595	Total Length (hrs)	∼12
# of 'angry' Utterances	1708	Sample Rate	16 kHz
# of 'neutral' Utterances	1084	File Type	WAV

Table 5.1: Summary of the IEMOCAP dataset, only including the audio part

The IEMOCAP dataset contains audio, transcriptions, video, and motion capture recordings of dyadic mixed-gender pairs of actors. There are five different sessions, with ten different actors in total. The recordings are a mix of improvisations of affective scenarios or performances of theatrical scripts. These are separated by hand into utterances. Each utterance has been labeled categorically over the following emotions: angry, happy, sad, neutral, frustrated, excited, fearful, surprised, by at least three different annotators, and dimensionally over the axes of valence (positive vs. negative); activation (calm vs. excited); and dominance (passive vs. aggressive) by at least two different annotators.

For our purposes we are only interested in the audio part of the dataset, so just that part of the dataset will be considered from here on out.

5.4 Preprocessing Dataset

The model used for testing purposes was only trained to recognize the emotions 'anger', 'sadness', 'happy', and 'neutral'. This meant that not all the utterances that are in the IEMOCAP dataset were able to be properly detected by the chosen model and thus all the utterances with emotions that it was not capable of recognizing had to be filtered out. The exclusion criteria precisely were any utterances that were classified as emotions other than 'neu', 'hap', 'ang', and 'sad'. Any utterances which had an undetermined emotion class were also excluded. This was done by checking if an utterance was of one of the four possible emotions, and if not then it would not be considered for the test set.

5.5 Baseline Experiment

To start the experiments, it was important to set a baseline performance for the model. There are multiple reasons for doing this. The first is that we can ensure the model performs as expected on the data given the accuracy claim by the author of the model. Secondly, it provides us with a reference point that we can use to measure the impact of frequency filtering. Any performance decrease can then only be attributed to the filtering and not the model or dataset.

Testing the baseline performance of the model is straightforward. We took the preprocessed dataset and ran the classifier on all of the utterances. This then resulted in a CSV file that contained the utterance file name, the emotion label, and the emotion as classified by the model. An example can be seen in Table 5.2.

Sentence	Emotion	Emotion_Guess
Ses01F_impro01_F000.wav	neu	neu
Ses01F_impro01_F001.wav	neu	neu
Ses01F_impro01_F002.wav	neu	neu
Ses01F_impro01_F005.wav	neu	neu
Ses01F_impro01_F012.wav	ang	ang

Table 5.2: A small excerpt of the CSV file that was created during this experiment

With this data, we could then compare the classifier results against the true labels to determine the accuracy of the model. This resulted in an accuracy of 92,85%, which was significantly higher than the advertised accuracy of 78,7% (Avg: 75,3%) [17].

5.6 Audio Filtering Experiments

5.6.1 Filter Design

After ensuring that the baseline performance of the model was correct, we started with the frequency filtering phase of the research. We first had to select the type of audio filter to apply to our dataset. For our experiments, a band-pass filter was chosen. The band-pass filter allows us to specify a frequency range and only allow frequencies in that range to pass through, which is ideal for our needs.

The Butterworth filter [18] was selected due to its maximally flat frequency response, ensuring minimal signal distortion. This feature prevents the creation of ripples in the signal and results in a smooth, gradual decrease in frequency response [19]. Consequently, the audio in the transition regions remains unaffected, which ensures that there are no undesirable audio signals generated by the filter.

Initially, we employed the `lfilter` [20] function, which applies a filter causally along one dimension. This approach presented issues due to its tendency to introduce lag and phase distortion, which impacted the accuracy of our results.

To combat this issue, we ended up using the `filtfilt` function [21] instead of `lfilter`. This applies the filter both forward and backward, which eliminates any lag and results in zero phase shift. This ensured that the audio signal would be compromised as little as possible.

The order of the filter was also a setting that we experimented with. Through trial and error, it was observed that an order that was too high caused artifacts in the audio signal, where the original audio was lost and all that remained was a high-pitched noise. It was eventually decided that an order of 3 was our best option. Using this value, our audio was free of any artifacts while isolating the desired frequency range as much as possible.

5.6.2 Baseline Filter Experiment

To ensure the filter application did not introduce unintended issues, we conducted initial experiments using a filter spanning the entire frequency range of the audio files. The filter was applied to every file in the dataset and evaluated by the model to determine the impact of the filter.

The results indicated a slight decrease in model accuracy to 92.72%, which is only marginally lower than our baseline experiment at 92.85%. This outcome suggests that applying the filter did not significantly alter performance compared to using unfiltered audio.

Having established that the application of a filter did not have an impact on the classification accuracy of the model, the next experiment that was done was to filter the audio to different frequency ranges to test our theory. The first frequency range we picked was the narrowband frequency range.

This range, typically 300 Hz to 3400 Hz, is commonly used for telephone calls, making it a realistic scenario for our research. This was done to ensure that the filter had the intended result and we could immediately see what the impact of phone call quality audio was on the ability of the model to classify the emotions correctly.

First, we applied the filter to just a single utterance to make sure that the filter was applied correctly. Figure 5.1 shows the spectrogram of this utterance WAV file with the full frequency range until 8000 Hz, while Figure 5.2 displays the same file constrained between 300 Hz and 3400 Hz. Something to note here is that the frequency does not have a hard cut-off point. The cause for this is the relatively low order of the filter we used, which resulted in a gradual drop-off in the effectiveness of the band-pass filter. Increasing the filter's order would typically address this issue by creating a steeper cutoff. However, in our case, doing so introduced artifacts that prevented us from using a higher filter order.

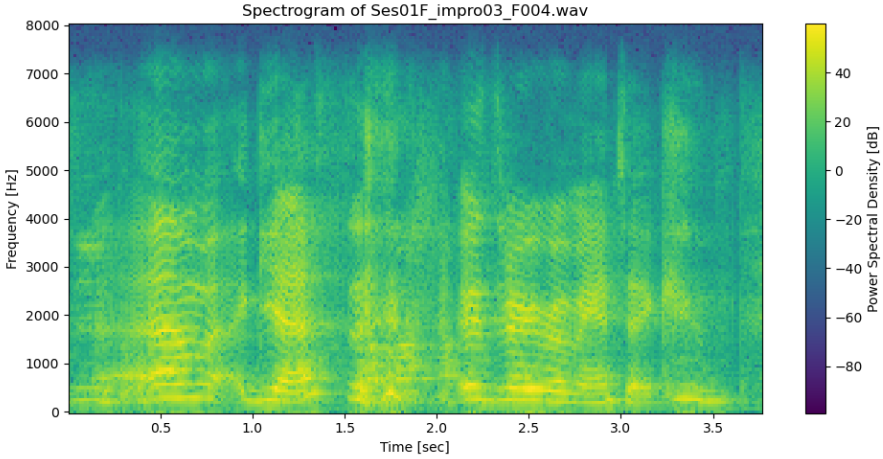

Figure 5.1: Spectrogram of the file SES01F_impro3_F004.wav from the IEMOCAP dataset

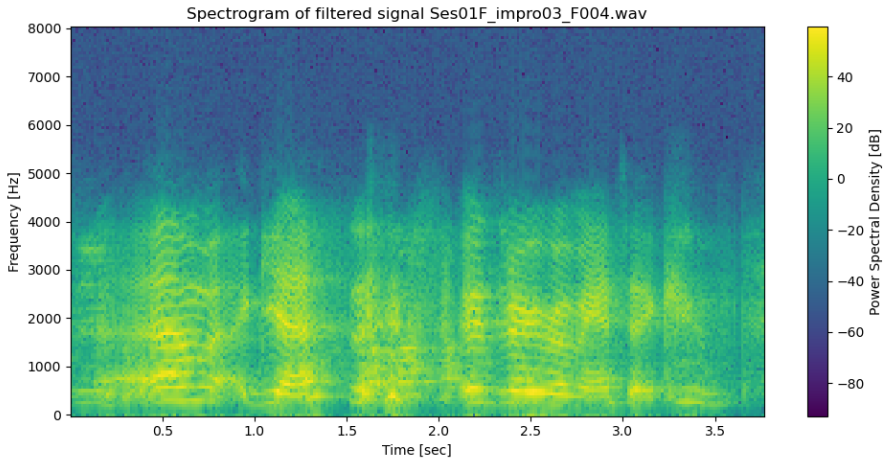

Figure 5.2: Spectrogram of the file SES01F_impro3_F004.wav from the IEMOCAP dataset with a band-filter applied with a range between 300 Hz and 3400 Hz

We then filtered the whole dataset with a band-pass filter between 300 and 3400 Hz and classified all the filtered files. We got an accuracy of 74.45%, which is significantly lower than our original accuracy of 92.72% with a filter over the whole frequency range. This indicated that there was some performance impact that could be interesting to assess and find any trends in.

The other tests were run on other reduced frequency ranges to determine if we could observe any other significant differences in performance compared to the baseline experiment.

5.7 Systematically Altering The Frequency Range on Audio Filters

In the final experiment during the testing phase, we implemented the system to systematically evaluate every frequency range and apply it to the dataset. This involved automating tests for every combination of minimum and maximum frequencies, as determined by predefined step sizes.

To achieve this, we created a loop that iterated over the specified ranges of minimum and maximum frequencies. For each combination, we applied corresponding band-pass filters to the audio data and conducted evaluations to assess performance metrics or criteria of interest.

Following each evaluation, the file name, the true label, and the prediction were recorded and systematically stored in individual CSV files designated for each frequency range with the same format as the baseline experiment. This approach allowed us to analyze and compare the impact of

different frequency bands on our testing outcomes effectively later.

5.8 Visualisation Methods

To visualise the performance of a classification model, a confusion matrix can be used. This is a table that visualises the predictions of the model against the actual labels. In a confusion matrix, each row represents a true label, with each column representing a predicted label. The cells within the matrix indicate the frequency with which instances are classified into each category. An example can be seen in Figure 5.3.

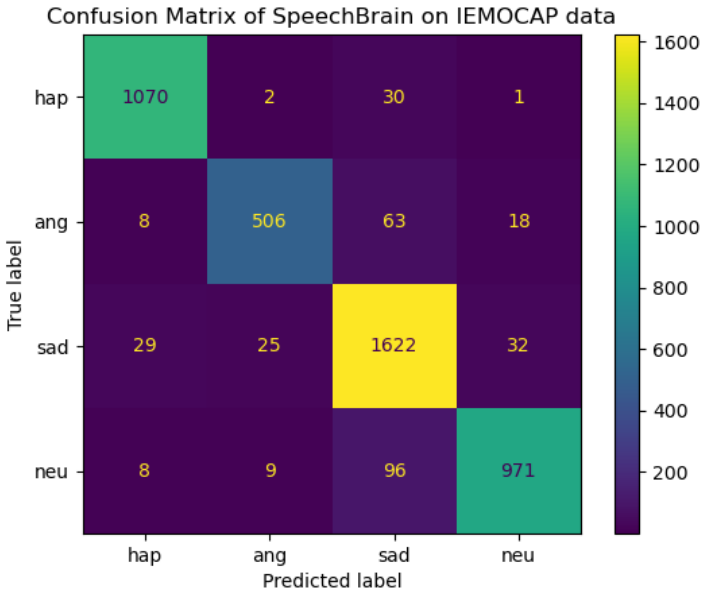

Figure 5.3: Example of a confusion matrix of the predictions on the unfiltered dataset.

Now, we were interested in determining how the ability of the model to classify the correct emotion changed as the frequency range was adjusted. Through visual inspection of the confusion matrices, we noticed that as we narrowed the frequency range, the model started getting confused and misclassifying emotions. It is however quite a slow and tedious process to have to go through each confusion matrix and inspect them to spot a performance trend. This then prompted us to come up with a method for effectively presenting this data intuitively and understandably. Given the amount of data points we were working with, we could potentially highlight a trend or pattern across the frequency ranges.

We came up with the idea of displaying the collapse of the emotion classi-

fication in a dendrogram. A dendrogram is a type of diagram representing the distances of attributes between pairs of merging classes. Since the confusion matrices seemed to present a pattern of certain emotions becoming indistinguishable from others, a dendrogram seemed like it would be a good fit. The idea was that when the model became confused between the two emotion classes and started misclassifying them as each other the distance would become less. This would then show on the dendrogram as the two emotion pairs having collapsed into one branch of the tree. This way of visualising was not successful, however, as this gave us the same type of issue as with the confusion matrices. Each frequency range would generate a dendrogram, leaving us with the same amount of graphs as before.

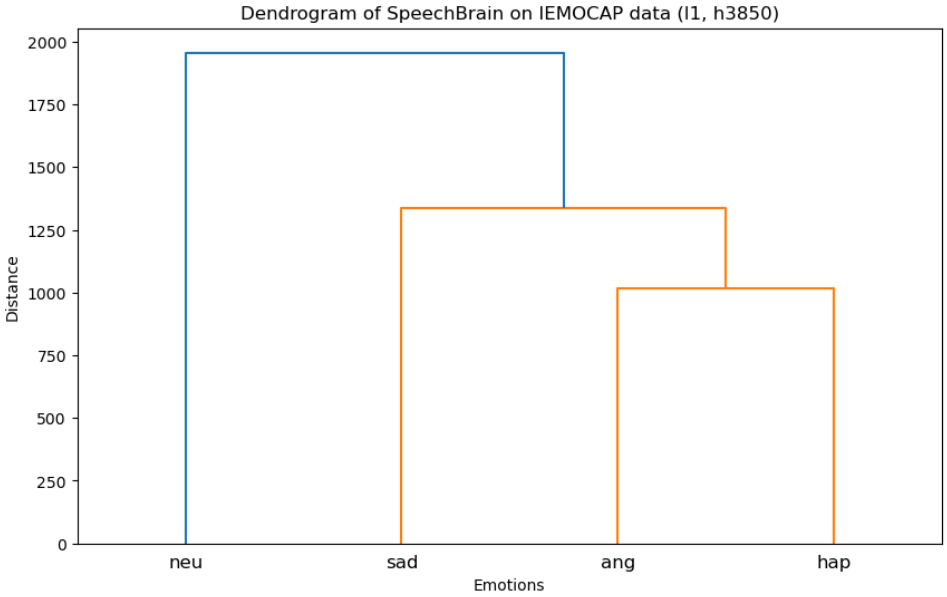

Figure 5.4: Example of a dendrogram that shows the distance between the different emotions for audio with a band-pass filter in the range of 21 Hz to 950 Hz.

The final idea for data visualization was realized through heat maps. The idea was that we could then represent each data point as a point on a grid, with the x-axis representing the lower cutoff boundary of the frequency range and the y-axis representing the higher cutoff boundary for the frequency range. The colour gradient would indicate the extent to which a certain emotion is being misclassified as another. The way the colour of the data point would be determined is as follows:

$$\text{accuracy} = \frac{\text{correct}}{\text{correct}+\text{wrong}} \textbf{ if } \text{correct} + \text{wrong} > 0 \textbf{ else } 0$$

Where 'correct' denotes the number of classifications that are correctly predicted and wrong is the number of classifications where the predicted label was equal to the emotion we are comparing the actual emotion to.

We then normalise this value to give us a normalised ratio of correct classifications to misclassifications for each emotional pair. Normalising ensures that we have consistent color mapping which allows us to better visually distinguish the accuracy values on the heat map. We then map this to a color range to indicate the value of this normalised ratio. Refer to Section 6.1 for a more thorough explanation of how to interpret the graphs.

With this approach, we could easily identify misclassifications by observing how each emotion was categorized, determining which specific emotion it was misclassified as.

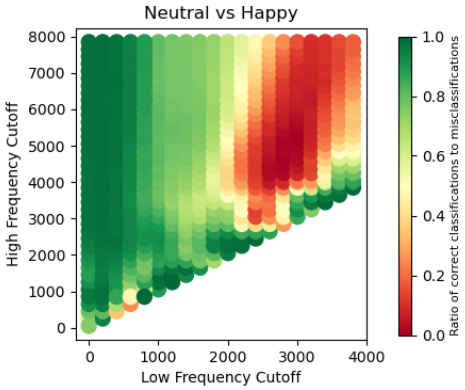

Figure 5.5: Example of a heat map that shows the ratio of correct classifications and incorrect ones between neutral and happy for the IEMOCAP dataset.

5.9 Testing on the MELD Dataset

In addition to the IEMOCAP dataset, we also conducted tests on the MELD dataset. The MELD dataset is a multimodal dataset for emotion recognition and sentiment analysis, which includes audio, visual, and textual modalities taken from the TV show 'Friends' [22]. Further relevant details of this dataset can be found in Table 5.3. Performing tests on the MELD dataset allows us to evaluate the generalisability of our model. If the performance of the model is similar across the board for both the IEMOCAP and MELD datasets, that would suggest that our findings are not specific to one dataset but apply more broadly to the model.

Since the MELD dataset contains the emotions anger, disgust, fear, joy, neutral, sadness, and surprise we, similarly to our exclusion criteria for the IEMOCAP dataset, excluded any utterance from the tests that were classified as emotions other than 'neutral', 'joy', 'anger', and 'sadness'. Another

excluded utterance was the utterance in the file 'dia38_utt4.wav'. This utterance caused the model to run out of memory and crash each time we tried to classify it, which meant it had to be excluded from being used for testing.

# of Total Utterances	13708	# of Annotators	5
# of 'sad' Utterances	1002	# of Speakers	407
# of 'happy' Utterances	2308	Total Length (hrs)	∼14
# of 'angry' Utterances	1607	Sample Rate	44.1 kHz
# of 'neutral' Utterances	6436	File Type	WAV

Table 5.3: Summary of the MELD dataset, only including the audio part

From this experiment, it was found that the baseline performance of the model on the MELD dataset was significantly worse. The accuracy of the model on the MELD dataset was 48.98%, nearly half the accuracy of the dataset that the model was trained on. Several factors may contribute to this discrepancy, including dataset characteristics, the complexity of emotional expressions captured in the MELD dataset, or simply that the model was overfitted on the training dataset.

Chapter 6

Results

In this section, we will share the different results that were determined using the methodology as described, on the emotion-recognition-wav2vec2-IEMOCAP model using both the IEMOCAP and MELD dataset.

6.1 Interpreting Heat Map Results

The series of heat maps in Figures 6.1 and 6.2 illustrate how varying constraining frequency ranges affect the classification performance of the model across different emotion pairs.

Each heat map compares two specific emotions, as indicated in the title of each plot. Each point on a heat map corresponds to a specific frequency range. The x-axis indicates the minimum frequency of a band-pass filter, while the y-axis shows the maximum frequency. The color of each point on the heat map represents how often the emotion on the left is correctly classified versus how often it is incorrectly classified as the emotion on the right. As previously mentioned, the formula for the colouring is as follows:

$$\text{accuracy} = \frac{\text{correct}}{\text{correct+wrong}} \textbf{ if } \text{correct} + \text{wrong} > 0 \textbf{ else } 0$$

Where 'correct' denotes the number of classifications that are correctly predicted and wrong is the number of classifications where the predicted label was equal to the emotion we are comparing the actual emotion to.

To give an example of how the colours work, we can see in the 'Sad vs Neutral' graph of Figure 6.1 that sad is classified correctly when the audio has a frequency range of 50 Hz to 5000 Hz. This is indicated by a green point on the heat map. This green point indicates that the emotion on the left is not being misclassified as the emotion on the right. If we look at another point in the same graph, when the frequency range is between 3000 Hz and 5000 Hz, we observe that that point has a red colour. This suggests that sadness is being misclassified nearly all of the time and that at least some of

these incorrect identifications are being classified as neutral. So, the color gradient from green to red indicates the number of misclassifications, with green representing no misclassifications of the emotion on the left, and red representing that the emotion is always misclassified and at least sometimes as the emotion on the right.

6.2 IEMOCAP

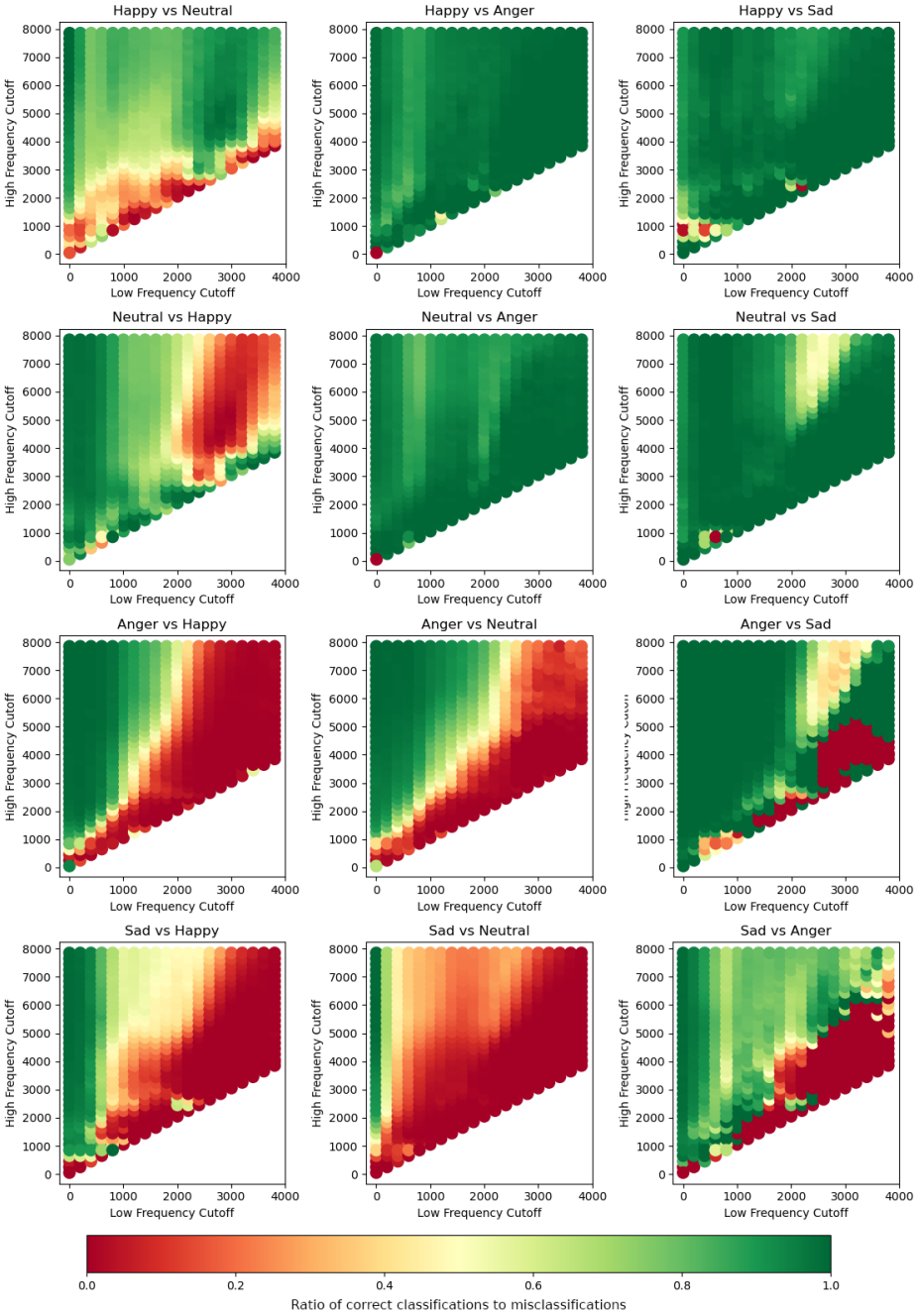

Figure 6.1: Heat maps of emotion pairs showing SpeechBrain model classification accuracy on the IEMOCAP dataset with color indicating correctness of emotion classification.

6.3 MELD

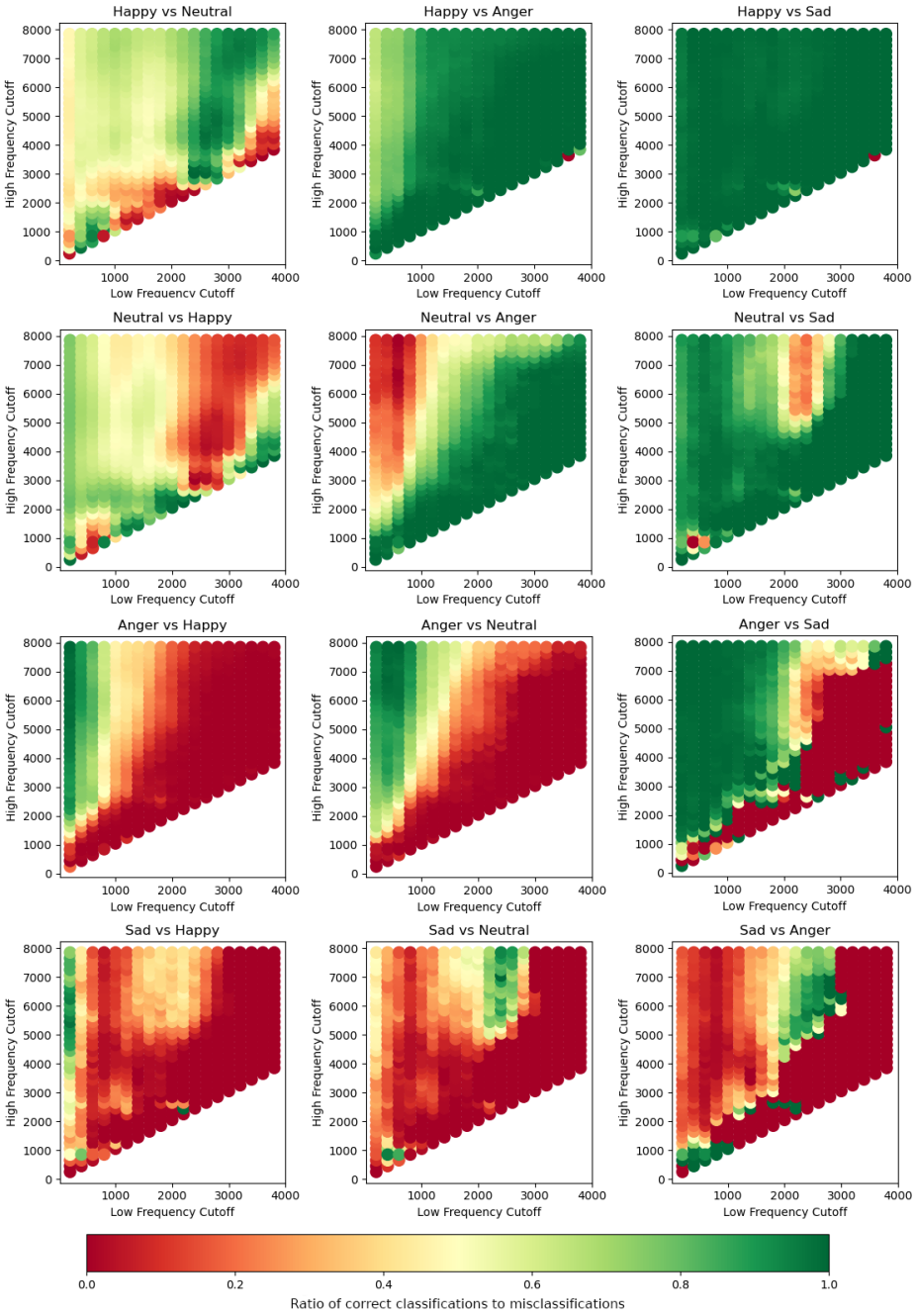

Figure 6.2: Heat maps of emotion pairs showing SpeechBrain model classification accuracy on the MELD dataset with color indicating correctness of emotion classification.

Chapter 7

Discussion

In this thesis, we set out to investigate the influence of frequency filtering on SER model performance. We carried out a multitude of experiments, created a testing pipeline and a useful way of visualizing the performance data.

Let us first look at Figure 6.1. From the results, some performance trends start to appear. Both the happiness and neutral emotions seem to exhibit stability when they are being compared to anger and sadness. Neutral does seem to get confused after we filter out the lowest 2000 Hz of our audio, while happiness gets misclassified as neutral to some degree in almost all the frequency ranges except some of the ranges where no low end is cut off.

As we begin to filter out a significant portion of the lower end of the frequency range, anger is frequently misclassified as either happiness or neutral, especially after the 2000 Hz lower frequency cutoff point. The model does seem more resilient with anger being classified as sadness, as for most of the frequencies with a cutoff point below 2000 Hz it does not get these emotions mixed up.

Sadness appears to be the most unstable emotion, particularly when contrasted with the neutral emotion. Once we filter out the first 500 Hz of the lower frequencies, sadness gets misclassified as neutral nearly all the time. Beyond the 1000 Hz mark, the model also starts to struggle with distinguishing sadness from happiness. Similarly when comparing anger and sadness, sadness and anger also exhibits the same behaviour that it gets confused between the two after we cut off the first 2000 Hz.

These trends seem to be mirrored in the MELD dataset, as depicted in Figure 6.2. While the overall performance of the dataset is always lower, we can visually identify similar trends as discussed for the IEMOCAP dataset. This tells us that the performance is not just based on the dataset, but that it is based on the model.

A small difference between the IEMOCAP and MELD datasets is that with the MELD dataset neutrality and anger get confused, but when we cut

off around 1000 Hz of the low end, performance seems to improve here.

A limitation of this method is that it could potentially become more unclear to visualise performance trends as you use models that can classify more emotions. Generating a graph for every emotion pair means you have $2^n - n$ graphs, where n is the amount of emotions a model can detect. This is an exponential limit, and can thus grow to unwieldy big sizes quite quickly.

Another limitation is that these graphs do not indicate the amount of misclassifications that occur. A lower ratio indicates that there are more misclassifications of the emotion on the left, and that at least some of those are misclassifications to the emotion on the right. Then you can manually look up what the actual numbers are for these misclassifications, so this graph only gives an indication of misclassification.

Some of these limitations could be addressed in future research. A better way to visualise this performance which creates fewer graphs could be interesting, as well as a way to colour the graphs such that it is clear how often an emotion is misclassified as the other. Future research could also explore incorporating additional real-world factors, like distortion, noise, and various forms of artifacts, and the impact these have on the performance of SER models.

When analysing our results, it becomes evident that applying frequency filters plays a critical role in the robustness of SER models. Previous studies, as discussed by Wani et al. (2010), have found that altering audio signals can significantly affect the impact of emotion recognition accuracy. The results of this study align with these studies, showing that there is a need for robust preprocessing methods to mitigate the influence of this noise.

Another conclusion we could make is that the way we classify emotions is different from what emotions are. When we classify emotions, they are put in a collection of unordered labels (anger, sadness, happiness, etc.), yet when we more closely inspect the data we found here it seems like, acoustically speaking, a hierarchy with a specific structure. Emotions seem to be based on specific audio cues and these cues have differing levels of robustness. This indicates that some emotions share similar acoustic features, creating a spectrum rather than distinct, separate categories.

Chapter 8

Conclusions

In this thesis, we explored the impact of frequency filtering on the performance of Speech Emotion Recognition models. We established a testing pipeline that can be used for assessing the performance of an SER model with a given dataset. We also created a novel way of visualizing the performance to make it easier to detect performance trends.

Our research shows that the performance of an SER model is significantly influenced by the frequency components in speech. This highlights the importance of frequencies in SER and to consider the frequency characteristics of the data you are trying to classify.

Moving forward, future research could expand by including other audio noise elements or enhancing the current method of visualising the data. By continuing to refine our understanding of these nuances, we can enhance the capabilities of SER technology in diverse applications, making them more robust and effective across various real-world applications.

Bibliography

[1] C. Busso, M. Bulut, C. Lee, A. Kazemzadeh, E. Mower, S. Kim, J. Chang, S. Lee, and S. Narayanan, "Iemocap: Interactive emotional dyadic motion capture database," *Journal of Language Resources and Evaluation*, vol. 42, pp. 335–359, 2008.

[2] I. Tokuda, "The source–filter theory of speech," Nov. 2021.

[3] E. Rodero, "Intonation and emotion: Influence of pitch levels and contour type on creating emotions," *Journal of Voice*, vol. 25, no. 1, pp. e25–e34, 2011.

[4] O. Ali and S. Mohammed, "Audio noise reduction using low pass filters," *Open Access Library Journal*, vol. 04, Nov 2017.

[5] International Telecommunication Union, "G.711: Pulse code modulation (pcm) of voice frequencies," tech. rep., ITU-T, 1988.

[6] International Telecommunication Union, "G.722: 7 khz audio-coding within 64 kbit/s," tech. rep., ITU-T, 2012.

[7] E. da Silva Morais, R. Hoory, W. Zhu, I. Gat, M. Damasceno, and H. Aronowitz, "Speech emotion recognition using self-supervised features," *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6922–6926, 2022.

[8] A. Tawari and M. M. Trivedi, "Speech emotion analysis in noisy real-world environment," in *2010 20th International Conference on Pattern Recognition*, pp. 4605–4608, 2010.

[9] F. Burkhardt, A. Paeschke, M. Rolfes, W. Sendlmeier, and B. Weiss, "A database of german emotional speech," in *Proc. Interspeech*, pp. 1517–1520, 2005.

[10] T. M. Wani, T. S. Gunawan, S. A. A. Qadri, M. Kartiwi, and E. Ambikairajah, "A comprehensive review of speech emotion recognition systems," *IEEE Access*, vol. 9, pp. 47795–47814, 2021.

[11] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[12] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, and T. E. Oliphant, "Array programming with NumPy," *Nature*, vol. 585, pp. 357–362, Sept. 2020.

[13] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[14] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[15] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. D. Mori, and Y. Bengio, "SpeechBrain: A general-purpose speech toolkit," 2021. arXiv:2106.04624.

[16] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *CoRR*, vol. abs/2006.11477, 2020.

[17] SpeechBrain, "Emotion recognition with wav2vec2 base on iemocap." `https://huggingface.co/speechbrain/emotion-recognition-wav2vec2-IEMOCAP`, Accessed: 2024-06-24.

[18] S. Butterworth, "On the Theory of Filter Amplifiers," *Experimental Wireless & the Wireless Engineer*, vol. 7, pp. 536–541, Oct. 1930.

[19] L. D. Paarmann, *Butterworth Filters*, pp. 113–130. Boston, MA: Springer US, 2001.

[20] S. Contributors, "scipy.signal.lfilter," 2023. Accessed: 2024-06-30.

[21] S. Contributors, "scipy.signal.filtfilt," 2023. Accessed: 2024-06-30.

[22] S. Poria, D. Hazarika, N. Majumder, G. Naik, E. Cambria, and R. Mihalcea, "MELD: A multimodal multi-party dataset for emotion recognition in conversations," *CoRR*, vol. abs/1810.02508, 2018.

Appendix A

Appendix

A.1 GitHub Repository

All of the experiments that were conducted and the data produced during this research can be found at this repository: `https://github.com/teunvgisteren/bachelorthesis`