



School of Engineering

InIT Institute of Applied
Information Technology

Project Work Computer Science

Audio Processing for the Radiosands Art Project

Author

Tobias Schlatter
Daniel Wassmer

Main supervisor

Prof. Dr. Thilo Stadelmann

Sub supervisor

Prof. Dr. Sven Hirsch

Date

21.12.2018



DECLARATION OF ORIGINALITY

Project Work at the School of Engineering

By submitting this project work, the undersigned student confirm that this work is his/her own work and was written without the help of a third party. (Group works: the performance of the other group members are not considered as third party).

The student declares that all sources in the text (including Internet pages) and appendices have been correctly disclosed. This means that there has been no plagiarism, i.e. no sections of the project work have been partially or wholly taken from other texts and represented as the student's own work or included without being correctly referenced.

Any misconduct will be dealt with according to paragraphs 39 and 40 of the General Academic Regulations for Bachelor's and Master's Degree courses at the Zurich University of Applied Sciences (Rahmenprüfungsordnung ZHAW (RPO)) and subject to the provisions for disciplinary action stipulated in the University regulations.

City, Date:

Signature:

.....

.....

.....

.....

The original signed and dated document (no copies) must be included after the title sheet in the ZHAW version of all project works submitted.

Zusammenfassung

Radiosands ist eine Installation mehrerer ferngesteuerter Radiogeräte. Mit einer Vielzahl von Audiomeerkmalen und dem Transkript vieler Live-Radiosignale möchte der Künstler den Eindruck einer sich selbst entwickelnden, manipulativen Intelligenz im Radiosignal erzeugen, indem er den Sender oder die Lautstärke des einzelnen Radios ändert. Dafür ist es natürlich unerlässlich, mit einer Reihe interessanter Funktionen arbeiten zu können. In diesem Projekt untersuchen wir, welche Merkmale das Radiomedium ausmachen und wie diese dem Projektteam zur Verfügung gestellt werden können.

Es gibt einige Audioverarbeitungslösungen, die bereits Möglichkeiten bieten, diese Merkmale zu extrahieren. Aber keine davon basiert auf dem gleichen Medientyp. Daher wird das Testen und Validieren der vielversprechendsten Lösungen anhand der tatsächlichen Radiodaten zu einem entscheidenden Bestandteil unserer Arbeit.

Die Segmentierung in Sprache und Musik ist eine Aufgabe, die wir mit „inaSpeechSegmenter“ gelöst haben. Dieses wurde auf den Inhalten von französischen TV-Sendern trainiert und hat eine Genauigkeit von 97% erreicht. Die gleiche Lösung wenden wir für die Geschlechtererkennung in der Sprache an, bei der wir eine Genauigkeit von 92% auf unserem Testsatz erreichen.

In Sachen Musik interessiert sich der Künstler vor allem für die Genreerkennung. Wir haben keine passende Lösung gefunden, die Genres klassifiziert. Daher schlagen wir eine Alternative vor, die keine diskreten Klassen verwendet, sondern eine Skala, um die spektrale Gleichförmigkeit und den Perkussionsanteil eines Audiosignals zu beschreiben.

Obwohl unsere annotierten Daten zeigen, dass die Emotionen der Sprecher nur wenig variieren, wäre es interessant, Muster bezüglich Emotionen und Prosodie zu finden. Wir stellen eine Basis für die teilweise Extraktion dieser Merkmale in Form von Valenz- und Erregungsskalen bereit. Die Ergebnisse zeigen, dass sich an den Extremen der Skala einige Unterschiede in der Emotion finden lassen. Den Rest des Bewertungsprozesses geben wir jedoch an das Künstlerteam weiter. Es ist ein Thema, das sich für eine weitere Forschung in einer anderen Arbeit eignet.

Wir schliessen unsere Projektarbeit ab, indem wir aus allen bewerteten Lösungen eine einzige, gut dokumentierte Toolbox zusammenstellen. Die Python-Bibliothek kann problemlos in die vorhandene Pipeline des Projekts implementiert werden. Die Gesamtleistung wird höchstwahrscheinlich verbessert, sobald die Toolbox mit einer Sprach-zu-Text-Lösung gekoppelt wird.

Abstract

The Radiosands exhibition piece is an installation of multiple, remotely controlled radio sets. Using a multitude of audio features and the transcript of many live radio signals, the artist aims to create the impression of a self-developing, manipulative intelligence within the radio signal, by changing the tune or the volume of individual radios. Having a set of interesting features to work with is, of course, vital for the artist and his team. In this project thesis, we investigate what features the media radio has and how to make them readily available to the project team.

There are numerous solutions in audio processing that already provide the means to extract these features, but none of them are based on this exact type of media according to our research. Therefore testing and validating the most promising ones of them against the actual radio data becomes a crucial part of our work.

Segmentation into speech and music is a task that we solved using ‘inaSpeechSegmenter’ – which was trained on French TV broadcast content – arriving at an accuracy of 97%. Using the same library, we tackle gender recognition in speech where we achieve an accuracy of 92% on our test set.

Regarding music, the artist is mainly interested in genre recognition. We did not find a suitable solution that classifies into genres. So we propose an alternative that does not use discrete classes, but a scale to describe the flatness and percussiveness of a piece of audio.

Even though our annotated data shows that there is only little variation in the emotion of the speakers, it would be interesting to find patterns regarding emotion and prosody. We provide a basis to partially extract these features in the form of valence and arousal scales. The results show that on the extremes of the scale some emotional differences can be found. We cede the rest of the evaluation process to the artist’s team. This task is a separate topic that could be investigated further in another research project.

We then conclude our project work by compiling one single, well-documented toolbox from all evaluated solutions. The resulting Python library can be easily implemented in the existing pipeline of the project. Overall performance will most likely improve further once the toolbox is coupled with a speech-to-text solution.

Contents

1	Introduction	7
1.1	Background	7
1.2	Problem Statement	8
1.3	Contribution	9
1.4	Further structure	10
2	Foundations	11
2.1	Related Works	11
2.2	Audio Processing	12
2.2.1	Short Time Fourier Transformation	12
2.2.2	Spectral flatness	12
2.2.3	Percussive and Harmonic components	13
2.3	Method of Measurement	14
2.3.1	Confusion Matrix	14
2.3.2	Accuracy, Precision, Recall and F1-Score	15
2.4	Existing Software	16
2.4.1	Librosa	16
2.4.2	PyAudioAnalysis	16
2.4.3	InaSpeechSegmenter	18
3	Methodical Approach	19
3.1	Efforts and Priorities	19
3.2	Integration	19
3.3	Performance	20
4	Implementation	21
4.1	Evaluation Pipeline	21
5	Results	23
5.1	Features and Priorities	23

Contents

5.2	Test Set	23
5.2.1	Conclusion on Test Set	26
5.3	Module Evaluation	28
5.3.1	Segmentation	28
5.3.2	Gender recognition on speech	31
5.3.3	Genre Recognition in Music	33
5.3.4	Emotion recognition in speech	37
5.3.5	Content recognition on Speech	41
5.4	Creation of the Toolbox	41
6	Conclusion and Outlook	43
6.1	Future Work	43
	Bibliography	45
	List of Figures	49
	List of Tables	50
	Glossary	51
A	Appendix	52
A.1	Installation Instructions	52
A.2	Complementary Data on USB Drive	52
A.3	Task Definition	54

1 Introduction

High technology has become an essential part of many aspects of everyday life, no matter if we watch movies online, listen to music, read the news or browse our social media. Most of the time, what we see or hear has been tailored to our own preferences. Today, the art and quality of profiling users, and providing them their preferred content, is a critical factor of success.

The term ‘filter bubble’ has only recently been coined by Eli Pariser [10] and discussed in the media. The term describes a state of intellectual isolation resulting from algorithms selectively assuming the information a user would want to see and then showing them only this content without their awareness. Accordingly, users find themselves in a bubble of potential ignorance – missing the exposure to different and contrasting views. People not knowing about the fact that this is happening may fall victim to targeted manipulation.

The filter bubble is a concept that can be hard to comprehend for non-professionals. Especially now with the boom of technologies such as big data and artificial intelligence that are redefining the way we interact with the aforementioned platforms.

Furthermore, our fast-paced, multimedia-based culture leads to the consumption of only small chunks of content, be it the switching back and forth between apps on our mobile devices, watching a movie and being on social media, or answering emails and talking to someone in person at the same time.

Within the Radiosands exhibition piece, you can experience these phenomena with old-fashioned, trusted, analogue radio in place of high-tech devices. In the following sections, we go into more detail about what Radiosands is and put our thesis into context.

1.1 Background

Radiosands is an exhibition piece by Swiss-German artist Thom Kubli. It is an art installation that features more than a dozen radio sets distributed in the exhibition space (see figure 1.1). The

1 Introduction

volume, tuning and other audio characteristics of each radio can be remotely controlled to manipulate the signal. The visitor should be able to feel a manipulative intelligence within the radio that changes the signal based on the context of what is playing.

Imagine one radio set playing news on some disaster. As a reaction, all the other radios could tune to a station playing classical music. In another scenario, the bad news could be drowned out by playing positively framed music at a high volume.



Figure 1.1: Montage of the Radiosands exhibition piece [30]

On a more technical basis, the Radiosands team has already established a working prototype where they focus on speech recognition. Apart from the remotely controlled radio devices, there is a master node responsible for the analysis of the streams and the orchestration.

1.2 Problem Statement

Radio offers different types of content with equally different features. We differentiate between the two main content types music and speech, which may, however, overlap.

As for speech, this thesis focuses on paralinguistic analysis [13]. It comprises non-textual features like gender, emotional state or the identity of a speaker. Regarding music, we find non-textual features as well, which for example describe the atmosphere, tempo and genre.

1 Introduction

Evidently, both feature enumerations are not exhaustive. Indeed, finding such features is precisely one of the tasks at hand. As a simplification, we allow ourselves to use the term ‘paralinguistic’ not only for the aforementioned properties on speech but also for the ones on music.

Once the artist can conveniently extract these features from the radio streams, it will open up additional possibilities to achieve the appropriate experience .

Therefore, this thesis revolves around the central question: Is it possible to provide an interface to the paralinguistic information on radio broadcast using easily implementable solutions in audio processing.

This leads to the three sub-topics we want to investigate in this thesis.

Features on Radio Media We investigate what kind of features can be extracted from radio media using readily available solutions.

Eligibility for Radiosands We further condense this list of features considering and comparing their eligibility for the Radiosands project. We then rate individual performance to ascertain it can be deemed good enough for Radiosands.

Consolidated Solution We combine the selected solutions so that the Radiosands project team can quickly implement them into the existing and yet growing setup.

The scope of this thesis focuses on the non-textual features. Therefore, speech recognition – which is handled by another team of the Radiosands project – will not be considered. The project should be as independent from online connectivity as possible. Online solutions were, therefore, only considered as a last resort. This thesis does not aim to beat the state-of-the-art, but transfer existing solutions with an acceptable loss in accuracy.

1.3 Contribution

The main contribution of our work is a Python toolbox to extract paralinguistic information on radio broadcast, enabling easy gathering of semantic meaning from it. To achieve this, a multitude of libraries and machine learning models are evaluated. If no suitable implementation exists we suggest alternatives based on low-level feature extraction.

1.4 Further structure

In chapter 2 we show the current state of research on the topics discussed in this thesis. We also explain some of the key principles and libraries we use. Our methodical approach is delineated in chapter 3, while the implementation of the evaluation pipeline is described in chapter 4. Next, we come to the results in chapter 5 which includes the analysis of radio media, our test set and the evaluations we ran on different modules. Alternatives are presented in the same chapter. Lastly, we summarise our work and give a brief outlook in chapter 6.

2 Foundations

In section 2.1 we first give an overview on the current state of research on the overall topic but on the sub-challenges as well. We then describe some of the key principles in audio processing in section 2.2 and also explain the different methods of measurement in 2.3. Lastly, in section 2.4, we describe a choice of suitable software packages in our thesis' context.

2.1 Related Works

An overview of computational paralinguistics is provided by [13]. An overview of the state-of-the-art approaches and challenges are examined in [14]. To the best of our knowledge, there is no work concerning paralinguistic feature extraction on radio broadcast as a whole. Nonetheless, there are works related to the sub-challenges of it. The following enumeration is therefore grouped by segmentation, gender recognition on speech, emotion recognition on speech, genre recognition on music and content recognition.

Segmentation into speech and music is required to solve the identified subsequent challenges. In literature segmentation of music is prominent. Content-based segmentation with a support vector machine (SVM)¹ is addressed by [3]. An open source solution is provided by [26], it uses a convolutional neural network (CNN)² to tackle segmentation on TV broadcast. It does also address **gender recognition on speech**. The INTERSPEECH 2010 paralinguistic challenge [7] released a training corpus which made gender recognition gain momentum in this field of research. For example [12] – a Gaussian mixture model (GMM)³ based model. Experiments and evaluations on multiple machine learning models are conducted by [20].

1 Further reading on SVMs:

<https://medium.com/machine-learning-101/chapter-2-svm-support-vector-machine-theory-f0812effc72>

2 Further reading on CNNs: <http://cs231n.github.io/convolutional-networks/>

3 Further reading on GMMs: <https://brilliant.org/wiki/gaussian-mixture-model/>

An overview about the state of **emotion recognition on speech** is given by [9]. It examines available test sets and compares the state-of-the-art. [15] shows a holistic approach on emotion in speech and music. A prominent dataset for **genre recognition on music** is GTZAN [2]. CNNs are a prominent approach to solve genre recognition as seen in [17], [6] and [8]. Literature about **content recognition** on radio broadcast are scarce. Some approaches on TV broadcast can be found in [21]. Ultimately, most of the solutions are based on fingerprinting approaches.

2.2 Audio Processing

This section explains some of the audio processing methods and features used in the course of our thesis.

An audio signal can be observed in either time or frequency domain. From the time domain, features like amplitude and tempo can be extracted. A conversion to the frequency domain – where features like pitch are found – can be achieved by using a Fourier transformation.

2.2.1 Short Time Fourier Transformation

The short-time Fourier transform (STFT) is a variation of a Fourier analysis. The Fourier transform itself does not keep any temporal information. However, a STFT does and is, therefore, also applicable for non-stationary signals. To compute an STFT a signal is divided into shorter segments of equal length. Each segment is then transformed to the frequency domain. A STFT is often represented by a spectrogram. A more in-depth look at how the STFT works can be found on the web⁴.

2.2.2 Spectral flatness

The spectral flatness does quantify how much noise-like a sound is, as opposed to being tone-like [4]. It measures how equally the frequency is distributed over the whole spectrum. A signal where frequencies are all present at equal strength (white noise see figure 2.1) would approach one. A pure tone (see 2.2) on the other hand approaches zero. We used this measure in 5.3.3.

⁴ https://ccrma.stanford.edu/~jos/sasp/Short_Time_Fourier_Transform.html

2 Foundations

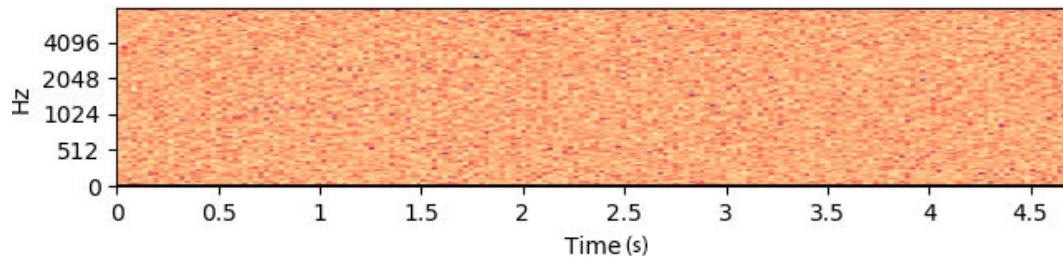


Figure 2.1: Spectrogram of white noise

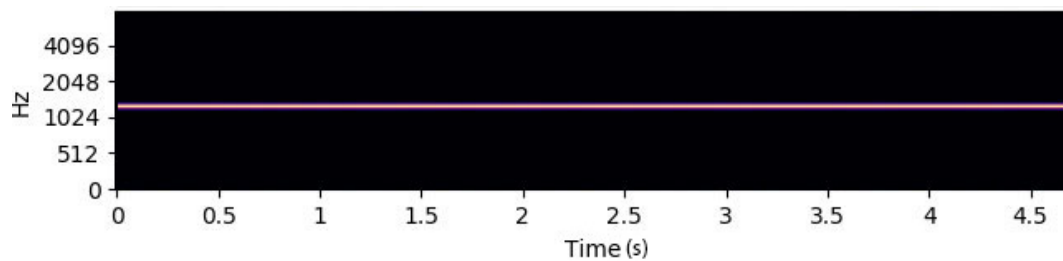


Figure 2.2: Spectrogram of a pure tone

2.2.3 Percussive and Harmonic components

An audio signal can be split into percussive and harmonic components. We used this in 5.3.3. First, a signal is converted to frequency space. Figure 2.3 shows the spectrogram representation of an audio source before the split.

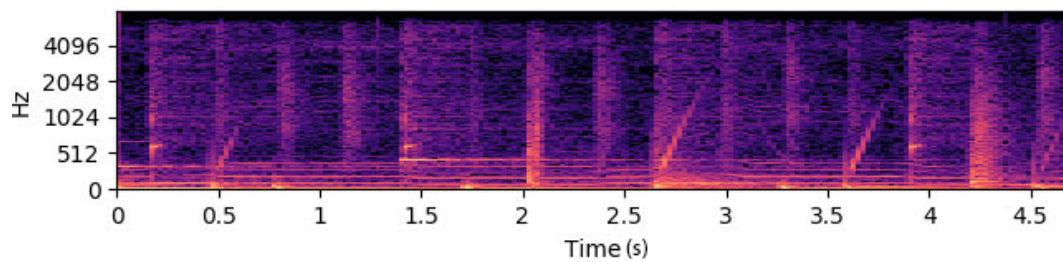


Figure 2.3: Spectrogram of an audio source

In a spectrogram representation, percussive sounds can be identified by their vertical structure (see 2.4). Harmonic sounds, on the other hand, have a horizontal structure (see 2.5).

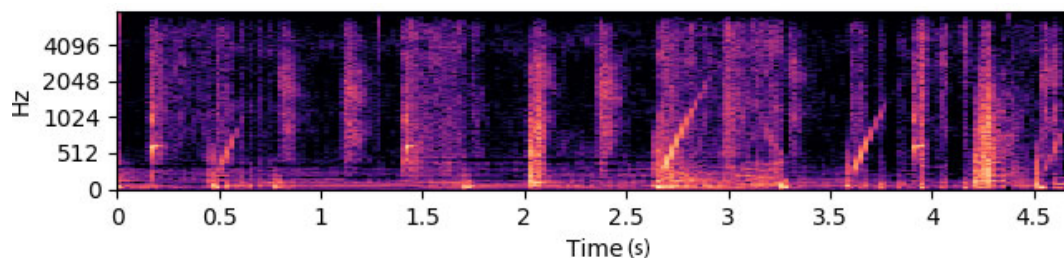


Figure 2.4: Percussive component of an audio source

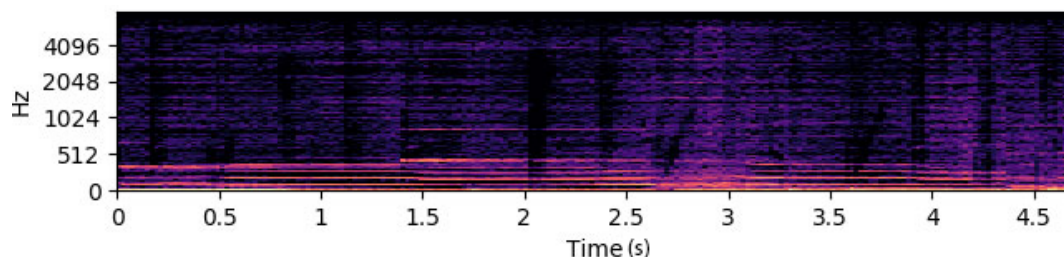


Figure 2.5: Harmonic component of source

The algorithm of [16] is used to conduct the split. In contrast to other algorithms, it can handle a certain amount of noise. After the split, the components can be transformed back to time domain using the inverse STFT .

2.3 Method of Measurement

To score the results of our experiments, we will use the same method of measurement whenever applicable. As we will do classification tasks, we need to use confusion matrices 2.3.1 to visualise the performance of the selected solutions. Based on it, we calculate a final score as described in section 2.3.2.

2.3.1 Confusion Matrix

A confusion matrix is a table that shows correct and incorrect classifications of a classifier. It makes it easy to see if the system is ‘confusing’ a particular class with another. Each row corresponds to an actual class and a column to the predicted class. True positives (TPs), false negatives (FNs), false positives (FPs) and true negatives (TNs) can easily be read from it (see figure 2.6).

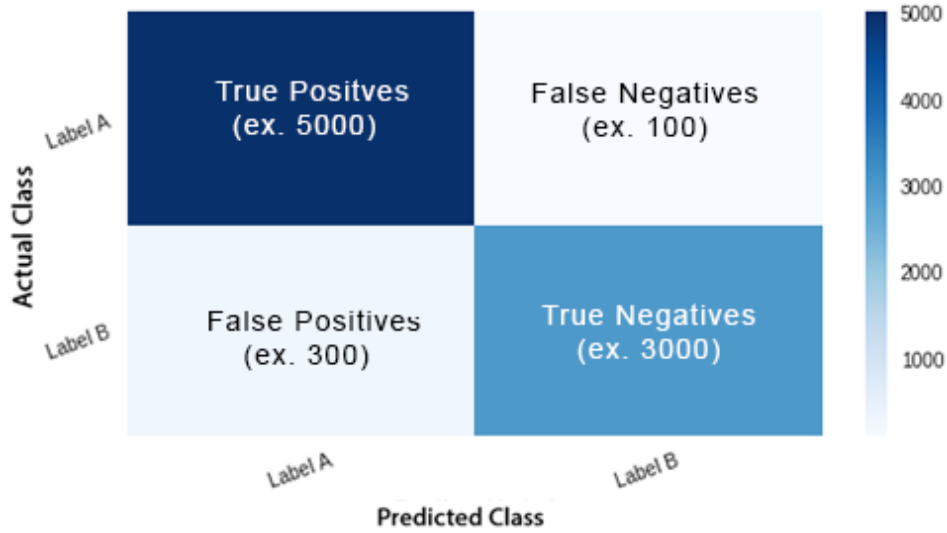


Figure 2.6: Confusion Matrix

In this thesis, the number represents the amount of data points that were classified as the corresponding class. Each data point is a 0.1-second chunk of an audio source.

2.3.2 Accuracy, Precision, Recall and F1-Score

The overall accuracy of a classifier is defined by the formula in 2.1.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.1)$$

If a data set is unbalanced, the overall accuracy will yield misleading results. Thus it is not always a reliable metric. For further judgement of a classifiers performance, precision (see 2.2), recall (see 2.3) and F1-score (see 2.4) are calculated.

$$Precision = \frac{TP}{(TP + FP)} \quad (2.2)$$

$$Recall = \frac{TP}{(TP + FN)} \quad (2.3)$$

$$F1\text{-score} = 2 * \frac{Precision * Recall}{(Precision + Recall)} \quad (2.4)$$

Precision indicates that proportion of positive identification that was actually correct. Recall indicates what proportion of actual positives was identified correctly. The F1-score is a weighted harmonic mean of precision and recall.

2.4 Existing Software

During our work, we focused on existing solutions in the field of audio processing. There are a lot tools available. Some of them are very specific to a certain problem, and others offer a variety of features for different applications. In this section, we would like to go into detail on three of them.

2.4.1 Librosa

Librosa [19] is a popular and actively maintained collection of audio processing tools written in Python. It provides not only a great number of feature extraction methods, but also many helpers and converters that enabled us to do quick prototyping throughout our work.

We used Librosa to perform most of the low-level feature extractions like STFT, root mean square (RMS)⁵ energy or spectral flatness (see 2.2.2). It is also used to calculate the separation of harmonic and percussive elements of a track, which is what we used directly for our alternatives for perceived intensity on music and speech.

2.4.2 PyAudioAnalysis

‘PyAudioAnalysis’ [18] is a Python library developed to support researchers on a variety of audio analysis tasks. It provides tools for feature extraction, classification, regression, training and segmentation tasks. It also includes a set of pre-trained models to be used as a starting point.

A list of extracted features to use with the provided models is shown in table 2.1.

This library is promising due to its wide range of functionality. We used its speech emotion model to classify audio on the valence and arousal scale (see 5.3.4).

⁵ Further reading on RMS: <http://mathworld.wolfram.com/Root-Mean-Square.html>

Feature Name	Description
Zero Crossing Rate	The rate of sign-changes of the signal during the duration of a particular frame.
Energy	The sum of squares of the signal values, normalized by the respective frame length.
Entropy of Energy	The entropy of sub-frames' normalized energies. It can be interpreted as a measure of abrupt changes.
Spectral Centroid	The center of gravity of the spectrum.
Spectral Spread	The second central moment of the spectrum.
Spectral Entropy	Entropy of the normalized spectral energies for a set of sub-frames.
Spectral Flux	The squared difference between the normalized magnitudes of the spectra of the two successive frames.
Spectral Rolloff	The frequency below which 90% of the magnitude distribution of the spectrum is concentrated.
MFCC	Mel frequency cepstral coefficient (MFCC) representation where the frequency bands are not linear but distributed according to the mel-scale.
Chroma Vector	A 12-element representation of the spectral energy where the bins represent the 12 equal-tempered pitch classes of western-type music (semitone spacing).
Chroma Deviation	The standard deviation of the 12 chroma coefficients.

Table 2.1: PyAudioAnalysis features, taken from [18]

2.4.3 InaSpeechSegmenter

The ‘InaSpeechSegmenter’ [26] is capable of performing a segmentation into speech, music and silence. Furthermore, it recognises the speakers gender [27] and classifies speech segments with either male or female.

The original aim of this library is to segment french TV broadcast based on audio. To do so, the training set consists of recordings from French speakers. It is optimised for French language. The segmentation is based on a CNN model and reaches an overall accuracy of 97.42% on the original test set. This library is promising since it was trained on a similar domain. After evaluation, it was chosen for segmentation as well as gender recognition (see 5.3.1 and 5.3.2).

3 Methodical Approach

At first, we explore the possibilities on the radio media. It is vital that we know what project-beneficial features we can obtain from it.

Ideally, we can identify libraries that offer straight-forward means to extract information from the audio signal. Whenever they do not, we think of an alternative approach using the available data. We aim towards fast and easy to implement solutions, in order to cover a wide range of features throughout our project work.

We move on to the prioritisation which is described in the following section 3.1. Next, we find available modules which we can use for Radiosands. The selection will happen with the constraints set by 3.2. Based on this we continue to test the performance of each selection in 3.3.

3.1 Efforts and Priorities

We evaluate implementation efforts for each potential module. The estimate is based on the yield of our research into the respective topic. In communication with the Radiosands team, we establish a priority ranking of the selected features. Using it as a foundation, we start working on the implementation and evaluation of possible solutions.

3.2 Integration

The project team is working on other parts of Radiosands in parallel. Therefore we choose solutions that build on their code base. This narrows down our choices to code written purely in Python or behind a Python wrapper. The code base is using a Python 2.6 environment. We do not restrict our choices to this version and look at Python 3 implementations as well. If reasonable we adapt the code of the chosen library to be in accordance with Python 2.6.

3.3 Performance

We test each implementation against the annotated data described in 5.2. Calculating the performance of each with the metrics described in 2.3. If there is more than one module candidate, we choose the one with better performance. We also note that within this project there is no need to optimise for top accuracy. Reaching an accuracy that is no more than 5% worse than the original is sufficient for the needs of the project.

4 Implementation

In section 4.1 we present how the evaluation pipeline was set up, that we used to assess the performance of a piece of audio processing software against our test set.

4.1 Evaluation Pipeline

For every evaluation, one pipeline was set up. While they differ depending on the module under evaluation, the basic structure was kept the same. A pipeline reads one audio file after another from a specified array of folders. A comma-separated values (CSV) table was created with the name of the audio file containing the same columns used for the annotation in the test set (see 5.2). Each audio file was split into chunks of usually three seconds and saved as a temporary .wav file. Depending on the module, a set of transformations (see 5.3) was applied to extract the desired classes or continuous values. The extracted classes or values were written into the corresponding column of the CSV file.

The newly created CSV files were then combined into an overview. The overview provides confusion matrices 2.3.1 and a set of key figures (precision, recall, accuracy and F1-score – see 2.3.2) per audio file, and overall. Additionally, a ‘D3.js’-application¹ was built to visualise the performance of the analysed modules.

¹ D3.js is a Javascript library to visualise data. Further information on D3.js: <https://d3js.org/>

4 Implementation

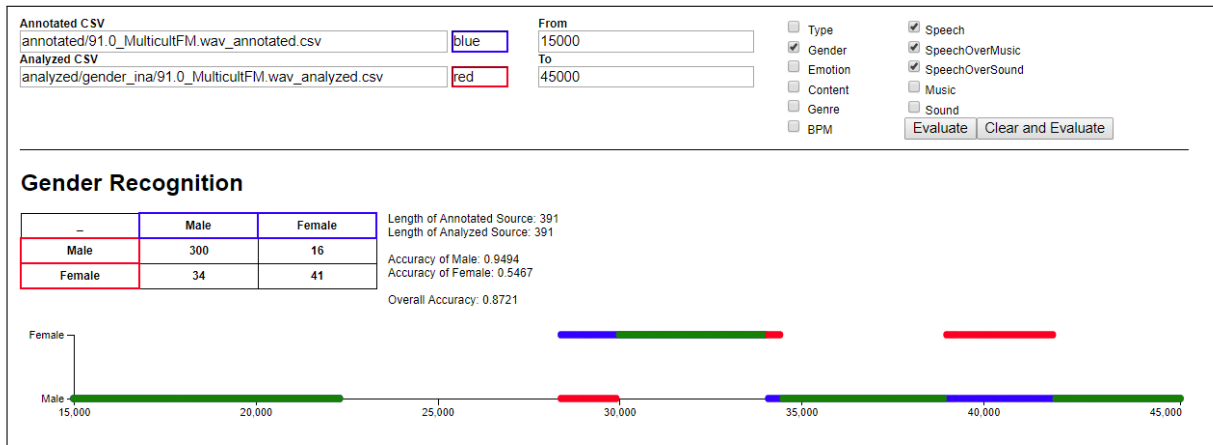


Figure 4.1: Screenshot of D3.js data visualisation. Here an evaluation on gender recognition is performed. Red data points = predicted class, Blue data points = actual class, Green data points = predicted and actual class match

5 Results

All of our results are presented in this chapter. In the beginning, we describe how the features to pursue were chosen. We go on to describe the structure and analysis of our test set in section 5.2. Next, we discuss the results for each individual module in the section Module Evaluation. Finally, in section 5.4, we summarise which modules were consolidated into one toolkit.

5.1 Features and Priorities

We defined a collection of possibilities based on our own observations while listening to radio. Our choices were also influenced by some of the discussions we had with the project team. Initial research into all the topics revealed that they have high differences in yield.

The priorities for the Radiosands project were determined by the artist and his team. Their primary focus was on speech and not on music. Accordingly, the priorities in table 5.1 have been established. We structured our approach respectively.

5.2 Test Set

In this section, we exemplify the created test set which was used for the evaluation. Initially, multiple radio streams were captured on the actual Radiosands infrastructure. The recordings were annotated by hand in the categories type, content, gender, emotion, genre. In total, the final test set consists of five hours of radio broadcast recordings. Thereof, three hours were captured from four different Spanish radio stations. The remaining two hours were taken from eight different German radio stations. Each recording was annotated with an accuracy of a tenth of a second.

The categories split up as described in the following sub-sections.

5 Results

Feature	Effort Estimation	Priority
Speech/music Segmentation	Medium	High
Genre on music	Medium	High
Emotion on speech	High	High
Energy on speech	Low	Medium
Gender on speech	Medium	Medium
Emotion on music	High	Medium
Pitch on speech	Low	Low
Content on speech	Very high	Low
Tempo on speech	Medium	Low
BPM on music	Low	Low
Pitch on music	Low	Low
Scale/key of music	Medium	Low
Instrument on music	Very high	Low

Table 5.1: Radiosands project priorities (sorted by priority)

Type

In table 5.2, classes used for the category type are listed with corresponding duration and distribution over the whole test set. The classes music and sound are distinguished by the following characteristics: music describes segments containing parts of a song or instrumental (with a genre, title and artist). Segments classified as sound contain all sorts of sounds, noise, or audio clips (such as in radio jingles or background music). The amount of ‘speech over sound’ is surprisingly high. This is because on some stations news were entirely broadcast over background music.

Class	Duration [min]	Ratio
Speech	94.01	31.34%
Speech over music	8.42	2.81%
Speech over sound	43.51	14.50%
Music	145.03	48.34%
Sound	8.18	2.73%
Silence	0.87	0.29%

Table 5.2: Type classes with distribution

Gender

Table 5.3 shows how the parts containing speech are divided according to gender. These classes were only annotated if the audio segment was of type speech, ‘speech over music’ or ‘speech

5 Results

over sound’. These three types add up to a total length of 141.99 minutes, which is 47.33% of the whole test set.

Class	Duration [min]	Ratio
Male	83.84	59.05%
Female	58.15	40.95%

Table 5.3: Gender classes with distribution

Emotion

The annotated emotions of the parts containing speech are listed in table 5.4. These classes were as well only annotated if the audio segment was of type speech, ‘speech over music’ or ‘speech over sound’. Neutral and happy emotions are abundantly present in on our test set, while the other emotions are almost non-existent.

Class	Duration [min]	Ratio
Happy	28.90	20.36%
Sad	0.00	0%
Angry	0.03	0.02%
Neutral	112.78	79.43%
Fear	0.22	0.16%
Disgust	0.06	0.04%

Table 5.4: Emotion classes with distribution

Content

A listing of all content types and its distribution can be found in table 5.5. Only segments of type speech, ‘speech over music’, ‘speech over sound’ or sound were annotated. These four types add up to a total length of 154.12 minutes which is 51.37% of the whole test set.

Class	Duration [min]	Ratio
Moderation	39.53	34.93%
News	34.46	30.44%
Commercial	1.34	1.18%
Jingle	5.83	5.15%
Report	32.01	28.29%

Table 5.5: Content classes with distribution

Genre

The genres are annotated by the same ten classes as in the GTZAN test set [41]. It is a challenge to select an appropriate set of genres and analyse music correspondingly. In the process of annotation compromises had to be made to fit music into one of the genres. For example, electronica, which is played quite frequently nowadays, is not one of the genres. Additionally, a genre can comprise a high number of sub-genres. Moreover, these sub-genres sometimes differ a lot in how they ‘feel’ towards a listener. For example, rock ranges from rock ballads and classic rock to hard rock which sound fairly different. We address this issue in 5.3.3.

Listing 5.6 shows how the classes are distributed within the test set. To annotate the genre, only the types music and ‘speech over music’ were considered. These two types add up to a total length of 153.45 minutes which is 51.15% of the whole test set. The test set does not contain any ‘hip hop’ nor reggae and only very little disco – this is not representative for the music played on radio, but rather an artefact of the recorded radio stations.

Class	Duration [min]	Ratio
Blues	2.63	1.72%
Classical	37.14	24.20%
Country	24.7	16.10%
Disco	0.84	0.55%
Hip Hop	0	0%
Jazz	3.74	2.44%
Metal	7.58	4.94%
Pop	47.80	31.15%
Reggae	0	0%
Rock	29.02	18.91%

Table 5.6: Genre classes with distribution

In addition, beats per minute (BPM), artist and song title were annotated for every music sequence. The BPM values were collected from songbpm.com, bpmdatabase.com and tunebat.com. It ranges from 70 to 175 BPM. The artist and song title were gathered with the help of the Shazam App.

5.2.1 Conclusion on Test Set

The data set is not big enough to train a machine learning model and is only used for evaluation purposes. Additionally, we note that our test set has quite some unbalanced data. Still, by

5 Results

running existing models against our test set, we are able to extract valuable information about the model's performance for our specific case.

5.3 Module Evaluation

This section details our findings during the evaluation of the different modules according to the Radiosands project priorities. We start with the segmentation into speech and music in 5.3.1, which builds the foundation for the rest of the modules. We move on to the gender, genre and emotion recognition in sections 5.3.2, 5.3.3 and 5.3.4 respectively. We close with the attempt of recognising content on radio broadcast in 5.3.5.

5.3.1 Segmentation

An accurate segmentation into speech, music and silence segments is critical for an accurate system as a whole. Being at the root of other modules, errors have a big impact (subsequent errors) and, therefore, have to be minimised. In the context of the Radiosands art project, it is desirable to classify segments with someone talking, as speech. All segmentation-modules were therefore instructed to classify ‘speech over music’ and ‘speech over sound’ as speech.

Segmentation with a SVM

A first approach to tackle the segmentation was a pre-trained SVM provided by pyAudioAnalysis [18]. The model is not supposed to be state-of-the-art but with pyAudioAnalysis it is straightforward to implement. It can segment audio into speech and music. It is not apparent what data it was trained on. To train it, however, the feature set of pyAudioAnalysis (see 2.1) was used. Therefore, these features also had to be extracted for segmentation.

To evaluate its performance, we ran the segmentation against our test set. Figure 5.1 and table 5.7 respectively show how it performed. Speech was misclassified as music quite often and had a precision of only 86.09%. A closer look at the affected data revealed that misclassification most often took place on ‘speech over music’ or ‘speech over sound’. Silence was not recognised at all and classified as either speech or music. It is left in the chart for reasons of comparison. The majority of silence classified as speech were pauses between speech and, therefore, considered as part of a speech segment.

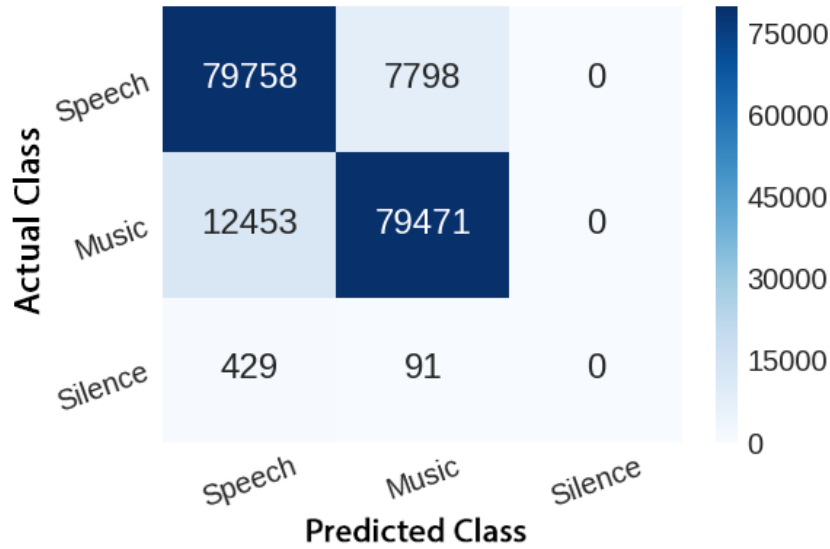


Figure 5.1: SVM segmentation confusion matrix

	precision	recall	F1-score
Speech	86.09%	91.09%	88.52%
Music	90.97%	86.45%	88.65%
Silence	0%	0%	0%

Table 5.7: SVM segmentation scores per class

The overall accuracy of the SVM is 88.46%. To further improve these results we were looking for a model that is trained on known material and preferably on a similar domain.

Segmentation with a CNN

The InaSpeechSegmenter is trained on TV broadcast audio. We, therefore, expected a better performance due to the similarity of the domain. Moreover, the InaSpeechSegmenter originally aims to classify ‘speech over music’ and ‘speech over sound’ as speech—which is what we want as well. In addition, it also recognises silence. Mel¹ scaled filter banks² are directly fed to the CNN-model. Therefore the audio signal was split into 25 ms sliding windows with 10 ms overlap.

1 Mel is an experimentally determined [1] psychoacoustic measure of how differences between sounds are perceived by listeners. With Mel scale applied to a signal lower frequencies are elevated, and only frequencies perceived by listeners are taken into account.

2 A filter bank is an array of separate frequency sub-bands of the original signal.

5 Results

To evaluate the performance of this model we ran it against our test set. Figure 5.2 shows that speech and music are classified with a significantly better accuracy. Speech is misclassified as music more than ten times less (compare 5.1). We observed an increase in accuracy for all classes. Table 5.8 shows an increase in F1-Score to about 97% for the classes music and speech.

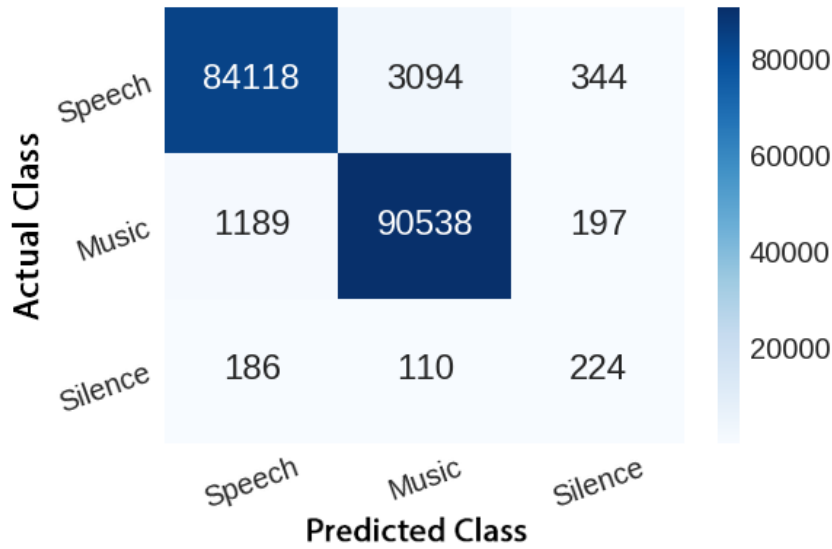


Figure 5.2: InaSpeechSegmenter segmentation confusion matrix

	precision	recall	F1-score
Speech	98.39%	96.07%	97.21%
Music	96.58%	98.49%	97.53%
Silence	29.28%	43.08%	34.86%

Table 5.8: InaSpeechSegmenter segmentation scores per class

Still, some classification errors were made on segments containing ‘speech over music’. Also, the classification of silence does not perform well and results in a low F1-score. It turns out that the library classifies short segments of silence, where on the other hand our annotation considered it as part of speech. Another downside of this library is its use of more computational power than the SVM approach. It is left to the project team to decide whether they accept this in favour of a better segmentation. However, not only segmentation is handled by this model but gender discrimination as well (see 5.3.2).

An evaluation of the original performance is not available as it is only given in combination with gender recognition. After consultation with the project team, we considered this library as suitable for this purpose. Therefore it was added to the final toolkit.

Further improvement could be reached with a system currently developed at the ZHAW [29]. It is trained on radio broadcast and specifically aiming for very high precision on speech.

5.3.2 Gender recognition on speech

The ability to differentiate between genders was a high priority to the project team. It opens up a great number of possibilities for the Radiosands installation.

Gender recognition with a GMM

First, the GMM approach from [36] was implemented. It is trained on a subset (only audio clips with male or female speech) of the AudioSet [23]. It contains 558 female-only and 546 male-only speech utterances. Each clip is ten seconds long. The training set is a diverse mixture of interviews, moderation, speeches and other audio clips. We assumed that this diversity represents the multitude of radio contents.

To use this model, the fundamental frequency³ and the MFCCs⁴ (see 2.1) were extracted from the audio source. The original experiment [36] reported an overall accuracy of 94%. On our test set, however, this model failed and is not suitable for our purpose. Actually, figure 5.3 and table 5.9 show it performed only slightly better than a random classifier. We suspect that the GMM for one had a substantial overfit on the data used for training and secondly the data seems to be less comparable to radio broadcast than we thought.

³ defined as the lowest frequency of a periodic waveform

⁴ MFCCs are a compact representation of mel scaled frequency bands. Further reading: http://ismir2000.ismir.net/papers/logan_paper.pdf

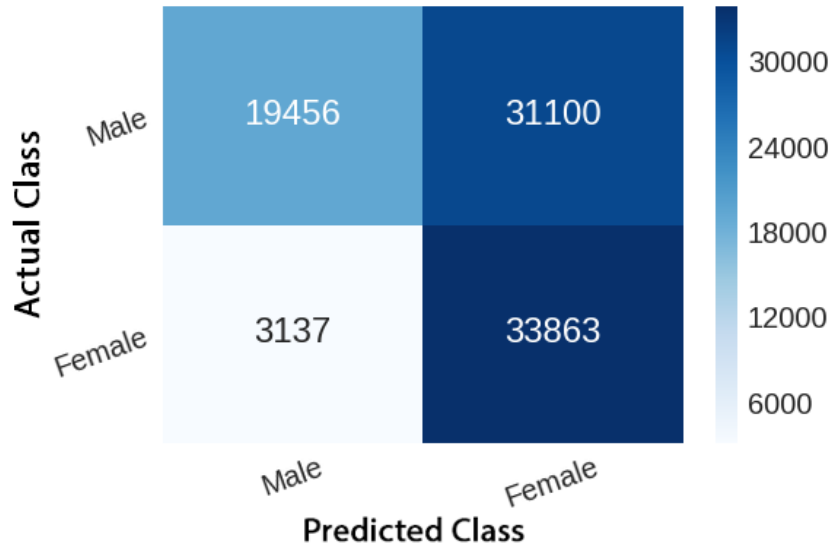


Figure 5.3: GMM gender recognition confusion matrix

	precision	recall	F1-score
Male	86.12%	38.48%	53.20%
Female	52.13%	91.52%	66.42%

Table 5.9: GMM gender recognition scores per class

Gender recognition with a CNN

The InaSpeechSegmenter that is used for segmentation (see 5.3.1) is also used for gender recognition. On the original test set, gender recognition reached an overall accuracy of 97.42%. Table 5.10 also shows the male and female recall of the original test. The original test set consisted of more male than female speakers. F1-scores for each class however are not provided.

male recall	female recall	overall accuracy
98.04%	95.05%	97.42%

Table 5.10: InaSpeechSegmenter original scores [26]

To evaluate gender recognition of the InaSpeechSegmenter we ran it against our test set. The overall accuracy reached 92.18%. On a per class basis an F1-score of 93.34% and 90.54% respectively was reached. (see figure 5.4 and table 5.11)

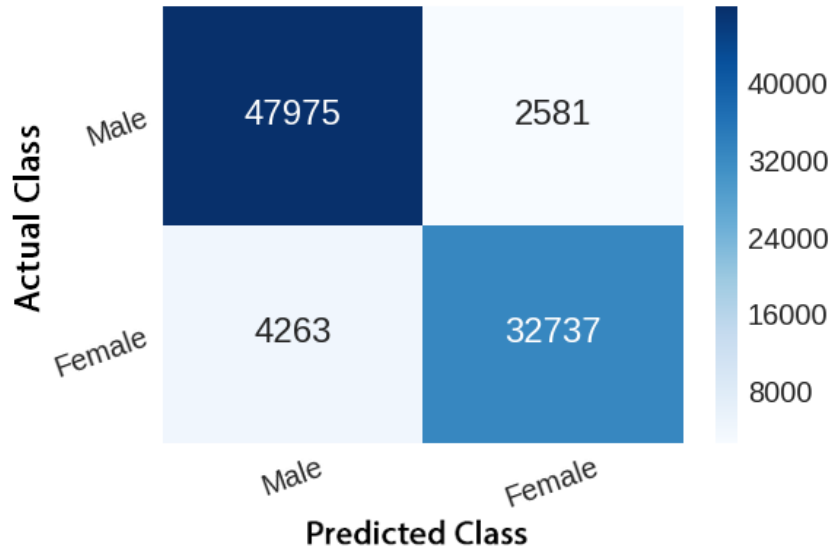


Figure 5.4: InaSpeechSegmenter gender recognition confusion matrix

	precision	recall	F1-score
Male	91.84%	94.89%	93.34%
Female	92.69%	88.48%	90.54%

Table 5.11: InaSpeechSegmenter gender recognition scores per class

Most often, a misclassification occurs for one of the two reasons: Either a male speaking over audio is predicted as female or a relatively low female voice is predicted as male. The documentation of the InaSpeechSegmenter states that acoustic correlations of speaker gender are language dependent. Our test set consists of German and Spanish speakers, the library however, is trained on French speakers. Keeping that in mind, a drop of about 5% compared to the original result is within the expected deviation.

As a further improvement, a model could be trained to take multiple languages into account. Other interesting models are provided by Kory Becker [20]. Though to use it in Python a porting from R is needed.

5.3.3 Genre Recognition in Music

In this section, we investigate the possibilities to recognise the genre in music. To evaluate genre recognition models we were only looking at pure music segments. Because if it were ‘speech over music’ our system would consider it as speech, and therefore a genre would not be needed.

Genre Recognition with a CNN

Two different CNN models were evaluated against our test set. None of them were selected for the final toolkit due to poor accuracy. Both models were trained on the GTZAN [2] data set and a mel scaled spectrogram representation is used as input. The ‘Deep Music Genre Classification’-model [40] is based on [17], [42] and [37]. Figure 5.5 shows the confusion matrix of this model. It reached an overall accuracy of 21.69% on our test set.

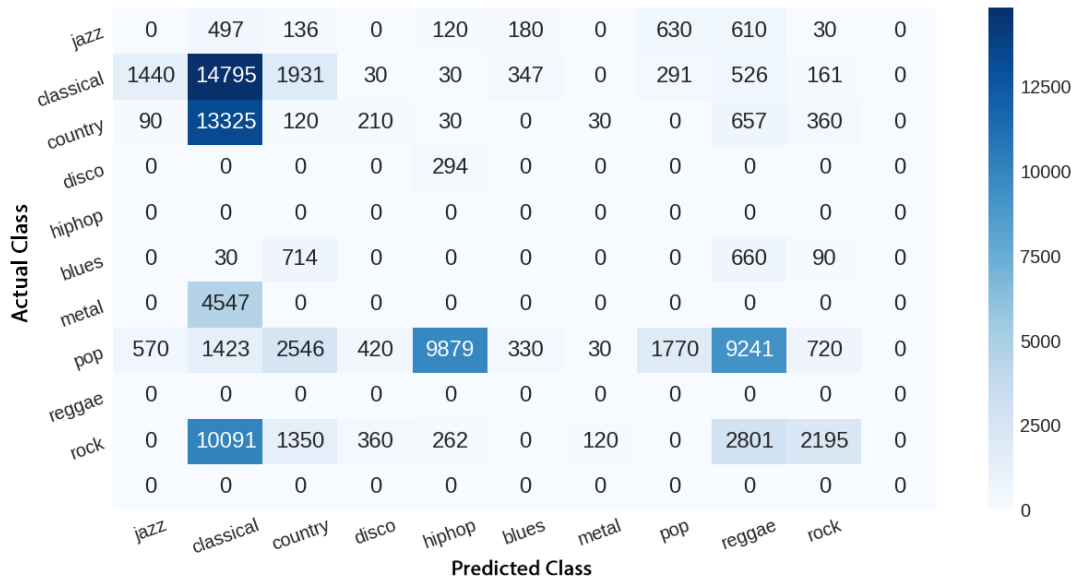


Figure 5.5: Confusion matrix of the ‘Deep Music Genre Classification’-model

Using CNNs for genre recognition seems to be the state-of-the-art technology [22]. We wanted to make sure the first approach did not fail due to a defective model and evaluated another similar CNN-model from a different source [41]. Figure 5.6 indicates that its performance is comparable to the first approach. It reached an overall accuracy of 24.64%. This suggests that it was not because of a defective model, but rather because the GTZAN training set is not similar enough to our data. Additionally, genre usually applies for the whole song while the atmosphere can change fundamentally within a song (also see 5.6).

5 Results

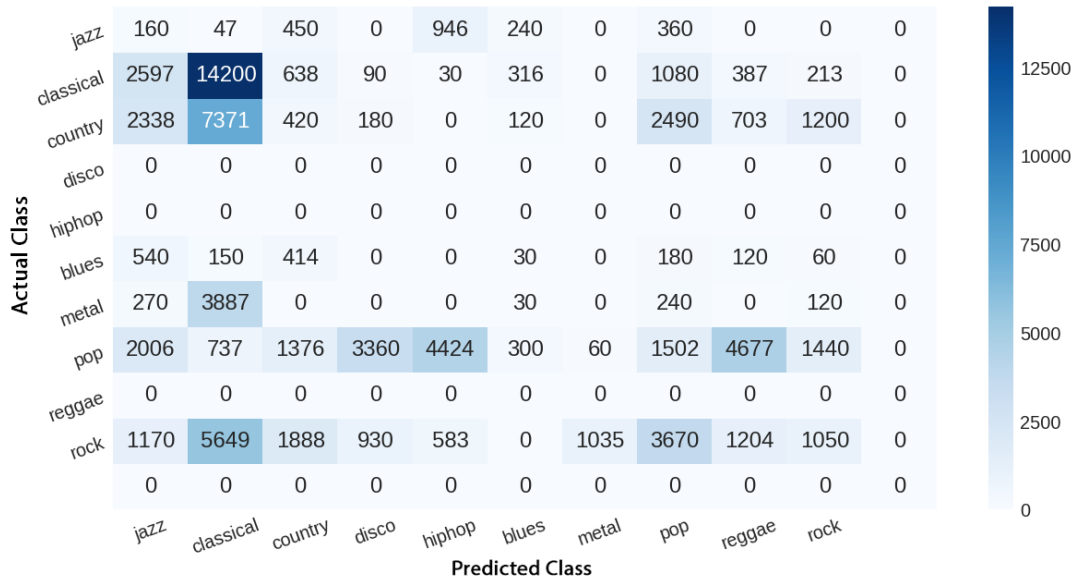


Figure 5.6: CNN genre recognition confusion matrix

Perceived intensity on music

The team’s interest in genre recognition originates from the fact that music carries a certain atmosphere and emotion which is often associated with genre. But ultimately it is the information within the genre that they are interested in. Classifying music into discrete genres does not benefit this goal. This insight and the insufficient performance of the classifiers led us to the conclusion to not follow this approach any further. We were looking for a solution focusing on how the music is perceived by a listener.

To find a better representation of how the intensity of music is perceived, we analysed the radio streams by plotting them against several low-level features. The best correlation was identified with spectral flatness (see 2.2.2). To extract the flatness, multiple steps were executed. First, a three-second audio chunk was transformed into frequency space using an STFT. The amplitude range over all frequencies within the audio chunk was then normalised so that it lies between zero and one. The flatness of a frame containing silence is approaching one because the frequency distribution is flat. Intensity of such audio, however, is of course low. To avoid peaks on silent audio, frames with energy below a certain threshold were ignored. Furthermore, the spectral flatness was calculated on a frame level basis. Finally, a running average was applied to suppress high impact of outliers.

For verification, we cut together a set of songs from multiple genres with different ‘intensities’. We sorted the songs based on their spectral flatness. As a result, a new piece of audio ranging

5 Results

from low to high spectral flatness was generated. The order of the new piece of audio was then verified to be according to our expectations. The focus was set to both ends of the scale, i.e. the extremes. Gentle songs are indeed found at the beginning (low flatness) while fast-paced, intensive songs are found at the end (high flatness).

Still, the position of some songs seemed wrong. For example, slow electronica and classical music were not distinguishable by their spectral flatness. To address this issue, a second value was introduced to the scale. In [17] a percussion/harmonics separation (see 2.2.3) was suggested. We figured that a great number of percussive elements are perceived as more intensive. We, therefore, measured how much RMS energy is present at a specific frame on the percussive component.

We illustrated the behaviour of selected genre classes on the two scales in figure 5.7 and 5.8. There is not a one-to-one correlation of how music is perceived on the chosen scales and what genre it is. Nonetheless, differences are clearly visible.

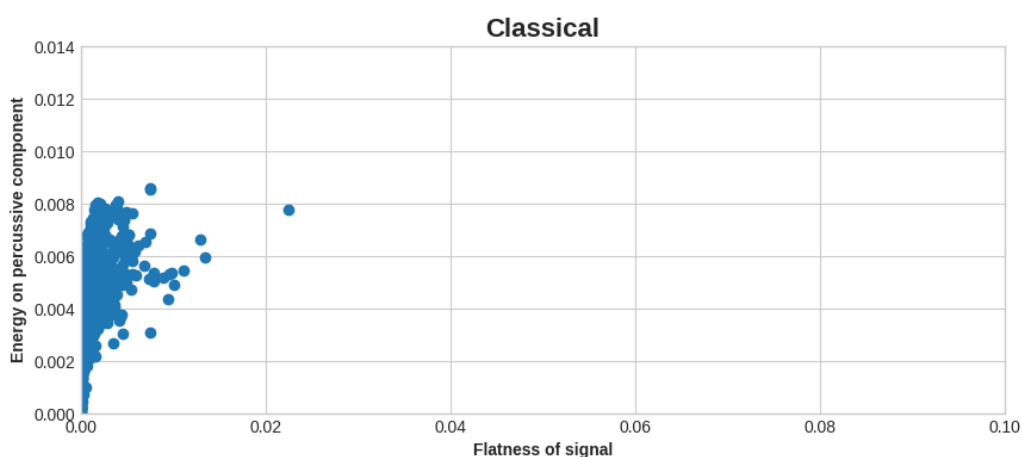


Figure 5.7: Classical songs from our test set plotted on our flatness-percussiveness scale

5 Results

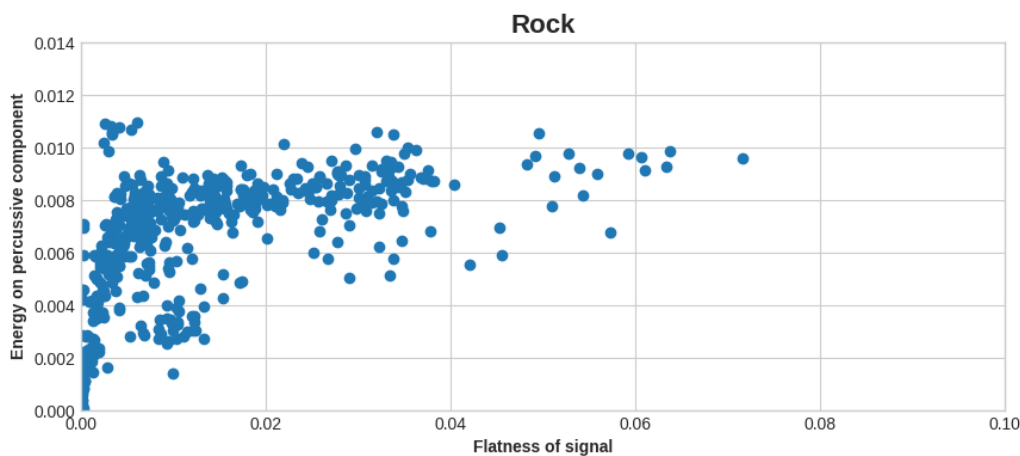


Figure 5.8: Rock songs from our test set plotted on our flatness-percussiveness scale

Alternatives

An alternative approach to the problem of genre recognition is the use of third-party services to identify a piece of music and gather data on it. It may yield more precise results than the aforementioned solutions.

One would first create a fingerprint of the song that is playing. For this step, a library called Chromaprint [39] could be used. Next, the fingerprint data could be sent to an audio identification service like ‘Acoustid’ [38] which would map the fingerprint to a description of the song (artist, album, track name etc.).

Using the information retrieved from the audio identification service, finding more information on a song becomes a task of querying yet another web service that provides such data. We would like to mention one example in particular, namely the ‘Spotify’ API [34]. Besides ways to fetch artist-based genre information [32] it also provides its own track features like danceability, instrumentality, valence and others [33].

However, this solution requires an internet connection and might necessitate a paid subscription for using the service.

5.3.4 Emotion recognition in speech

We found that the process of classifying emotions in speech is not straight forward. This was mostly due to the one-sided emotions on the radio data. Most of the solutions on hand needed

5 Results

training on the actual data [18] [25], which was impossible as our whole set did not contain all the possible classes and was very small.

More straight forward approaches were chosen. One describes the emotion in speech in the form of scalar values instead of definite classes [11]. Another tries to detect the perceived intensity on speech using scales for spectral flatness and percussiveness. We go into more detail in the following two sections.

Valence and Arousal

We used two of the pretrained SVM-regression-models of pyAudioAnalysis [18] [28] to find the valence and arousal values of our data. It comes with the training data that represents a subset of EmoDB [5] as well as ground truth data for valence and arousal.

We do not have annotated values for valence and arousal in our test set. An actual test result is, therefore, not provided. We did conduct some secondary tests, however.

Firstly, as we have access to the complete set of the EmoDB corpus, it made sense to run the regression with regard to its whole. It also offers data of a human perception test against all the recordings. We selected only the ones where 90% of the testers recognised the same emotion, and 70% rated the acted emotion as being natural.

By creating an arousal and valence scatter plot for each emotion class in EmoDB we were able to identify promising and problematic cases. Joy and sadness seemingly split up well into two groups as depicted in figure 5.9. Anger partially coincides with the bounds of the joy class but also separates itself from the centre.

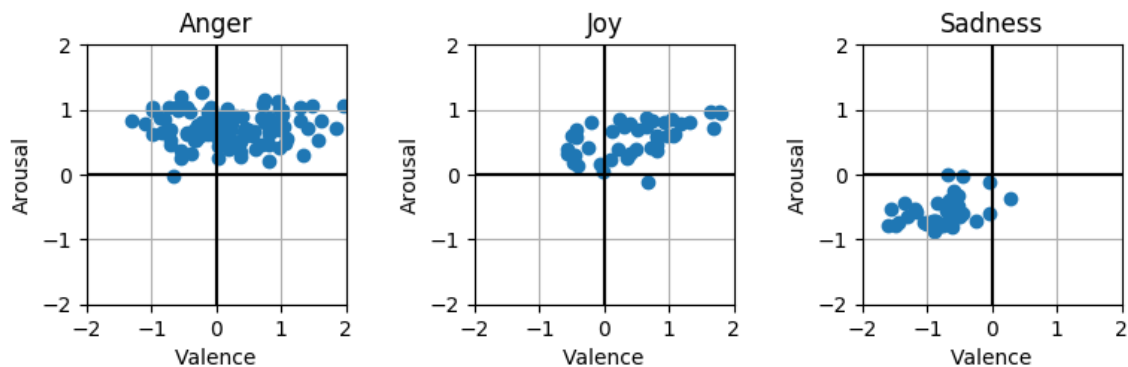


Figure 5.9: Test of the pyAudioAnalysis speechEmotion model against the EmoDB joy, anger and sadness classes

5 Results

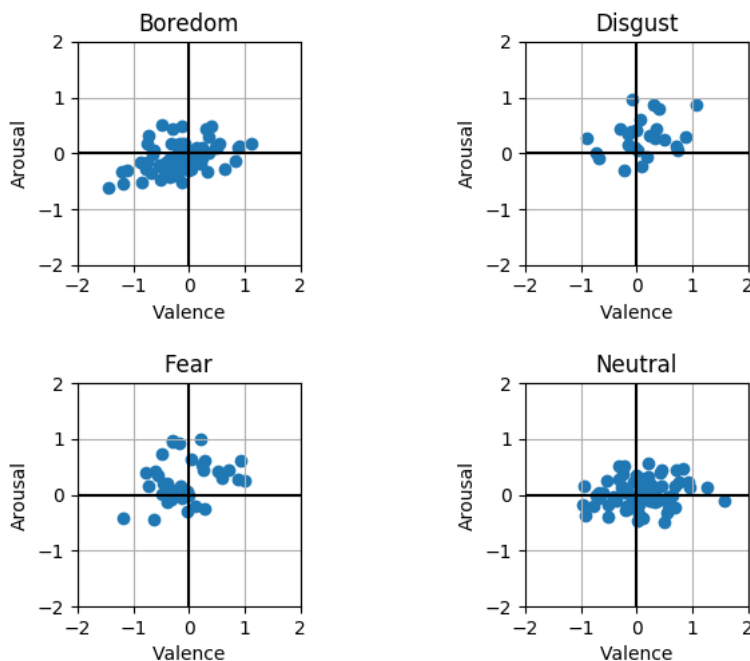


Figure 5.10: Test of the pyAudioAnalysis speechEmotion model against the EmoDB boredom, disgust, fear and neutral classes

Whereas the rest of the classes more or less fall into the same area in the plot (see figure 5.10), there is only little chance of being able to identify them reliably. This also impacts the promising cases negatively, as they share much space on the scale with the indiffereniable ones.

The same test was performed on the radio test set, where we extracted individual audio files for each emotion class. We observed no split over the two relevant⁵ emotions. In fact, the neutral class covers a significant area with its wide spread. This means either that our labelled emotions are incorrect or that the training data is not a good fit for our set.

⁵ We only use emotions with more than 1% overall share. See 5.4.

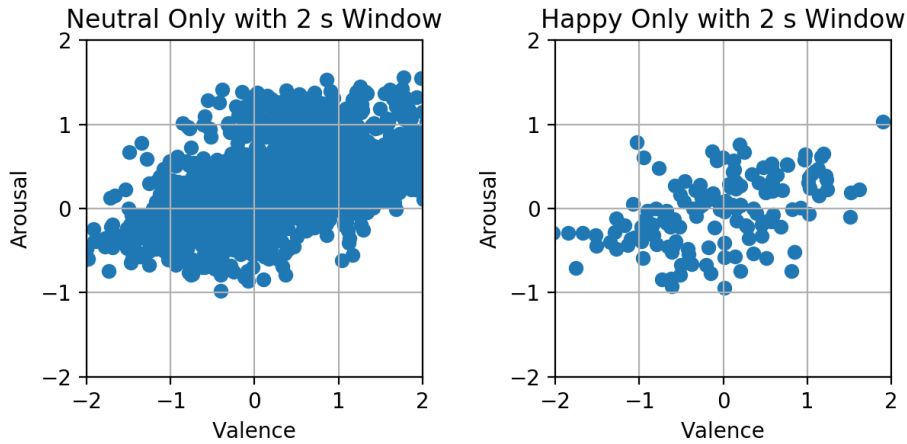


Figure 5.11: Test of the pyAudioAnalysis speechEmotion model against the radio test set happy and neutral classes

We conducted a human perception test on upper and lower bound values for valence. A total of 10 high and low valence clips were evaluated. Judging the mere sound of the recordings, leaving the content unheard, we determined that the high valence ones sound indeed more pleasant. The clips from the lower bound sound more calm but definitely not unpleasant. We also note that they mostly consist of male, German language, while the high valence audio is only in Spanish.

We conclude that the speech emotion model does not perform satisfactory on our data and even not on the EmoDB data, which was used for model training. Some use might be found in the extreme values. We see most of its potential in a strong combination with speech-to-text technologies. This topic lends itself for further research and experimentation.

Perceived intensity on Speech

The intensity scale approach we proposed for music (see 5.3.3) can also be applied to speech. We used the same technique to cut together two mixes of speech clips sorted by spectral flatness and percussiveness.

The differences on both ends of the scale were not as strong as on music. Nonetheless, on the low percussion end, we found slow and calm examples. On the opposite side, fast and strong speech – as it can be found in commercials – was playing.

5.3.5 Content recognition on Speech

No readily available solutions to differentiate between a multitude of content types have been identified. The project team emphasised that recognising commercials would be of special value. We, therefore, investigated if it is possible to use an ad blocking solution for this purpose.

Adblock Radio

Adblock Radio [35] is an ad blocker for online radio streams. The creators combine a machine learning approach and acoustic fingerprinting to detect commercials and other unwanted content. While the way of how it works is promising, the solution itself was not usable for Radiosands. One limitation was the fact that the models do not work on FM radio according to [31]. Each radio channel uses a different model, and a different fingerprint database – only a small number of German and Spanish channel are supported out-of-the-box [24]. They plan opening up the flagging of ads to the community, but this feature remained a work-in-progress at the time of writing [24]. We argue that training the models on our limited amount of data will not be of any use for the Radiosands project.

Alternatives

Often commercials are intentionally noisy to catch the listeners attention. To have a clue, whether one is playing, the spectral flatness used in 5.3.3 can also be applied to speech segments. A high value would indicate a potential commercial. 5.3.4 confirmed this approach as well.

Another possibility is to use a fingerprinting solution similar to the one discussed in 5.3.3. Most likely only particular ads are of interest. A database of acoustic fingerprints of all wanted commercials could be created, and their occurrence be checked continuously. An advantage certainly is that individual ads can be detected and not just ads in general.

5.4 Creation of the Toolbox

We progressively added the most promising findings from the previous section to the toolbox, which we called ‘Rabio’. It is a simple Python 2.6 module, which contains the segmentation of

5 Results

speech and music as well as gender recognition, highlighted in 5.3.1 and 5.3.2. Additionally, it includes the alternative approaches of flatness and percussiveness on music and speech discussed in 5.3.3 and 5.3.4. Last but not least the toolbox features the model for arousal and valence extraction from 5.3.4.

6 Conclusion and Outlook

Our aim with this thesis has been to provide an interface to paralinguistic information on radio broadcast. In chapter 5 we evaluated if existing machine learning solutions can be used for the extraction. If no suitable solution existed, alternatives based on low-level audio features are suggested. We use the library `InaSpeechSegmenter` to solve the challenge of segmentation and gender recognition. For emotion and genre, we propose alternative approaches that do not use fixed classes, but scalar values to express the mood and perceived intensity within the audio signal. Both of them rely solely on low-level features.

Returning to our initial question in 1.2 we conclude that it is indeed possible to create an interface to paralinguistics of radio media using solutions within reach. We combined the most useful modules in a toolbox¹. It enables the artists to extract semantic meaning from these paralinguistic features on a higher abstraction level.

A reason why the selected approaches for genre and emotion recognition are of use for the Radiosands project is because the extremes are of special value. A discrimination of small changes was not of great interest considering the relatively uniform radio media.

6.1 Future Work

We see much potential in combining our paralinguistic information with the information gathered from speech-to-text technologies. The Radiosands project team already has a working prototype featuring speech-to-text, and we recommend harvesting this potential.

The same applies to content recognition on radio which may profit greatly from speech-to-text, but following up on the approach used by `AdblockRadio` [35] would be an interesting next step as well. Most likely the approach could be reused for other classes like news, weather reports etc.

¹ To be found at <https://github.com/tobiasschlatter/radio> (private repository)

6 Conclusion and Outlook

We also remark that a limitation of our work was the small and unbalanced test set. A prerequisite for any future work is certainly the extension and revision of the annotations.

Bibliography

- [1] Stanley Smith Stevens, John Volkman and Edwin B Newman. ‘A scale for the measurement of the psychological magnitude pitch’. In: *The Journal of the Acoustical Society of America* 8.3 (1937), pp. 185–190.
- [2] George Tzanetakis and Perry Cook. ‘Musical genre classification of audio signals’. In: *IEEE Transactions on speech and audio processing* 10.5 (2002), pp. 293–302.
- [3] Lie Lu, Hong-Jiang Zhang and Stan Z Li. ‘Content-based audio classification and segmentation by using support vector machines’. In: *Multimedia systems* 8.6 (2003), pp. 482–492.
- [4] Shlomo Dubnov. ‘Generalization of spectral flatness measure for non-gaussian linear processes’. In: *IEEE Signal Processing Letters* 11.8 (2004), pp. 698–701.
- [5] Felix Burkhardt et al. ‘A database of German emotional speech’. In: *9th European Conference on Speech Communication and Technology*. Vol. 5. Jan. 2005, pp. 1517–1520.
- [6] Tom LH Li, Antoni B Chan and A Chun. ‘Automatic musical pattern feature extraction using convolutional neural network’. In: *Proc. Int. Conf. Data Mining and Applications*. sn. 2010.
- [7] Björn Schuller et al. ‘The INTERSPEECH 2010 paralinguistic challenge’. In: *Proc. INTERSPEECH 2010, Makuhari, Japan*. 2010, pp. 2794–2797.
- [8] Sander Dieleman, Philémon Brakel and Benjamin Schrauwen. ‘Audio-based music classification with a pretrained convolutional network’. In: *12th International Society for Music Information Retrieval Conference (ISMIR-2011)*. University of Miami. 2011, pp. 669–674.
- [9] Moataz El Ayadi, Mohamed S Kamel and Fakhri Karray. ‘Survey on speech emotion recognition: Features, classification schemes, and databases’. In: *Pattern Recognition* 44.3 (2011), pp. 572–587.
- [10] Eli Pariser. *The filter bubble: what the Internet is hiding from you*. London: Viking/Penguin Press, 2011.

Bibliography

- [11] Marcello Mortillaro, Ben Meuleman and Klaus R. Scherer. ‘Advocating a Componential Appraisal Model to Guide Emotion Recognition’. In: *International Journal of Synthetic Emotions (IJSE)* 3.1 (2012), pp. 18–32. (Visited on 20/12/2018).
- [12] Ming Li, Kyu J Han and Shrikanth Narayanan. ‘Automatic speaker age and gender recognition using acoustic and prosodic level information fusion’. In: *Computer Speech & Language* 27.1 (2013), pp. 151–167.
- [13] Björn Schuller and Anton Batliner. *Computational paralinguistics: emotion, affect and personality in speech and language processing*. John Wiley & Sons, 2013.
- [14] Björn Schuller et al. ‘Paralinguistics in speech and language—State-of-the-art and the challenge’. In: *Computer Speech & Language* 27.1 (2013). Special issue on Paralinguistics in Naturalistic Speech and Language, pp. 4–39. ISSN: 0885-2308. DOI: <https://doi.org/10.1016/j.csl.2012.02.005>. URL: <http://www.sciencedirect.com/science/article/pii/S0885230812000162>.
- [15] Felix Weninger et al. ‘On the Acoustics of Emotion in Audio: What Speech, Music, and Sound have in Common’. In: *Frontiers in Psychology* 4 (2013), p. 292. ISSN: 1664-1078. DOI: 10.3389/fpsyg.2013.00292. URL: <https://www.frontiersin.org/article/10.3389/fpsyg.2013.00292>.
- [16] Jonathan Driedger, Meinard Müller and Sascha Disch. ‘Extending Harmonic-Percussive Separation of Audio Signals.’ In: *ISMIR*. 2014, pp. 611–616.
- [17] Daniel Grzywczak and Grzegorz Gwardys. ‘Deep Image Features in Music Information Retrieval’. In: vol. 60. Aug. 2014, pp. 187–199. DOI: 10.1007/978-3-319-09912-5_16.
- [18] Theodoros Giannakopoulos. ‘pyAudioAnalysis: An Open-Source Python Library for Audio Signal Analysis’. In: *PloS one* 10.12 (2015).
- [19] Brian McFee et al. ‘librosa: Audio and Music Signal Analysis in Python’. In: *Proceedings of the 14th Python in Science Conference, 2015* (2015).
- [20] Kory Becker. *Identifying the Gender of a Voice using Machine Learning*. June 2016. URL: <http://www.primaryobjects.com/2016/06/22/identifying-the-gender-of-a-voice-using-machine-learning/> (visited on 24/09/2018).
- [21] Sergiy Bilobrov and Andres Hernandez Schafhauser. *Continuous content identification of broadcast content*. US Patent 9,703,932. July 2017.
- [22] Yandre MG Costa, Luiz S Oliveira and Carlos N Silla Jr. ‘An evaluation of convolutional neural networks for music classification using spectrograms’. In: *Applied soft computing* 52 (2017), pp. 28–38.

Bibliography

- [23] Jort F. Gemmeke et al. ‘Audio Set: An ontology and human-labeled dataset for audio events’. In: *Proc. IEEE ICASSP 2017*. New Orleans, LA, 2017.
- [24] Adblockradio. *adblockradio/available-models*. Nov. 2018. URL: <https://github.com/adblockradio/available-models> (visited on 18/12/2018).
- [25] M. Chen et al. ‘3-D Convolutional Recurrent Neural Networks With Attention Model for Speech Emotion Recognition’. In: *IEEE Signal Processing Letters* 25.10 (Oct. 2018), pp. 1440–1444. ISSN: 1070-9908. DOI: 10.1109/LSP.2018.2860246.
- [26] David Doukhan et al. ‘An Open-Source Speaker Gender Detection Framework for Monitoring Gender Equality’. In: *Acoustics Speech and Signal Processing (ICASSP), 2018 IEEE International Conference on*. IEEE. 2018.
- [27] David Doukhan et al. ‘INA’S MIREX 2018 MUSIC AND SPEECH DETECTION SYSTEM’. In: *Music Information Retrieval Evaluation eXchange (MIREX 2018)*. 2018.
- [28] Theodoros Giannakopoulos. *4. Classification and Regression*. Sept. 2018. URL: <https://github.com/tyiannak/pyAudioAnalysis/wiki/4.-Classification-and-Regression> (visited on 19/12/2018).
- [29] Zaniyar Jahany Hans-Peter Hutter Matthias Büchi. ‘KWS Key-Word-Spider: System zur Unterstützung der Segmentierung, Inhaltsanalyse und Codierung von audiovisuellen Medienbeiträgen’. In: 2018. URL: https://www.zhaw.ch/no_cache/de/forschung/forschungsdatenbank/projektdetail/projektid/1988/ (visited on 22/11/2018).
- [30] Thom Kubli. *Montage of the Radiosands exhibition piece*. received by email from the author on 11.12.2018. 2018.
- [31] Alexandre Storelli. *Designing an audio adblocker for radio and podcasts*. Nov. 2018. URL: <https://www.adblockradio.com/blog/2018/11/15/designing-audio-ad-block-radio-podcast/> (visited on 18/12/2018).
- [32] Spotify AB. *Get an Artist*. URL: <https://developer.spotify.com/documentation/web-api/reference/artists/get-artist/> (visited on 08/12/2018).
- [33] Spotify AB. *Get Audio Features for a Track*. URL: <https://developer.spotify.com/documentation/web-api/reference/tracks/get-audio-features/> (visited on 08/12/2018).
- [34] Spotify AB. *Spotify Web API*. URL: <https://developer.spotify.com/documentation/web-api/> (visited on 08/12/2018).
- [35] *Adblock Radio*. URL: <https://www.adblockradio.com/en/> (visited on 18/12/2018).

Bibliography

- [36] ML bot1. *Voice Gender Detection using GMMs: A Python Primer – Machine Learning in Action*. URL: <https://appliedmachinelearning.blog/2017/06/14/voice-gender-detection-using-gmms-a-python-primer/> (visited on 25/09/2018).
- [37] Sander Dieleman. *Recommending music on Spotify with deep learning*. URL: <http://benanne.github.io/2014/08/05/spotify-cnns.html> (visited on 10/10/2018).
- [38] Lukáš Lalinský. *AcoustID*. URL: <https://acoustid.biz/> (visited on 08/12/2018).
- [39] Lukáš Lalinský. *Chromaprint*. URL: <https://acoustid.org/chromaprint> (visited on 08/12/2018).
- [40] Evan Otero. *Deep Music Genre Classification*. URL: <https://github.com/evanotero/deep-music-genre-classification> (visited on 10/10/2018).
- [41] Evan Otero. *gtzan.keras*. URL: <https://github.com/Hguimaraes/gtzan.keras> (visited on 20/10/2018).
- [42] Bartosz Michalak Piotr Kozakowski. *Music Genre Recognition*. URL: http://deepsound.io/music_genre_recognition.html (visited on 12/10/2018).

List of Figures

1.1	Montage of the Radiosands exhibition piece [30]	8
2.1	Spectrogram of white noise	13
2.2	Spectrogram of a pure tone	13
2.3	Spectrogram of an audio source	13
2.4	Percussive component of an audio source	14
2.5	Harmonic component of source	14
2.6	Confusion Matrix	15
4.1	Screenshot of D3.js data visualisation. Here an evaluation on gender recognition is performed. Red data points = predicted class, Blue data points = actual class, Green data points = predicted and actual class match	22
5.1	SVM segmentation confusion matrix	29
5.2	InaSpeechSegmenter segmentation confusion matrix	30
5.3	GMM gender recognition confusion matrix	32
5.4	InaSpeechSegmenter gender recognition confusion matrix	33
5.5	Confusion matrix of the ‘Deep Music Genre Classification’-model	34
5.6	CNN genre recognition confusion matrix	35
5.7	Classical songs from our test set plotted on our flatness-percussiveness scale	36
5.8	Rock songs from our test set plotted on our flatness-percussiveness scale	37
5.9	Test of the pyAudioAnalysis speechEmotion model against the EmoDB joy, anger and sadness classes	38
5.10	Test of the pyAudioAnalysis speechEmotion model against the EmoDB boredom, disgust, fear and neutral classes	39
5.11	Test of the pyAudioAnalysis speechEmotion model against the radio test set happy and neutral classes	40

List of Tables

2.1	PyAudioAnalysis features, taken from [18]	17
5.1	Radiosands project priorities (sorted by priority)	24
5.2	Type classes with distribution	24
5.3	Gender classes with distribution	25
5.4	Emotion classes with distribution	25
5.5	Content classes with distribution	25
5.6	Genre classes with distribution	26
5.7	SVM segmentation scores per class	29
5.8	InaSpeechSegmenter segmentation scores per class	30
5.9	GMM gender recognition scores per class	32
5.10	InaSpeechSegmenter original scores [26]	32
5.11	InaSpeechSegmenter gender recognition scores per class	33

Glossary

BPM	beats per minute. 26
CNN	convolutional neural network. 11, 12, 18, 29, 34, 35, 49
CSV	comma-separated values. 21
FN	false negative. 14
FP	false positive. 14
GMM	Gaussian mixture model. 11, 31, 32, 49, 50
MFCC	mel frequency cepstral coefficient. 17, 31
RMS	root mean square. 16, 36
STFT	short-time Fourier transform. 12, 14, 16, 35
SVM	support vector machine. 11, 28–30, 38, 49, 50
TN	true negative. 14
TP	true positive. 14

A Appendix

A.1 Installation Instructions

If applicable, the installation instructions can be found in the complementary data in the form of a README file.

A.2 Complementary Data on USB Drive

document The L^AT_EX sources of this document and the PDF

emotion-evaluation All the files and the results of the evaluation of the emotion module

minutes-etc The meeting minutes, other documents



rabio The Rabio toolkit – the final product of this thesis

radiosands-development The evaluation pipeline and all its source files, the annotated ground truth data and all the WAV files

A Appendix

A Appendix

A.3 Task Definition

 <small>Zürcher Hochschule für Angewandte Wissenschaften</small> School of Engineering	[ONLINE ADMINISTRATION] PRAKTISCHE ARBEITEN	[DEPT. T ADMIN] TOOLS
zurück		Logout
Projektarbeit 2018 - HS: PA18_stdm_5		
Allgemeines:		
Titel:	Machine Learning to Support Artists	
Anzahl Studierende:	2	
Durchführung in Englisch möglich:	Ja, die Arbeit kann vollständig in Englisch durchgeführt werden und ist auch für Incomings geeignet.	
Betreuer:		
HauptbetreuerIn:	Thilo Stadelmann, stdm 	Zugeteilte Studenten:
		Diese Arbeit ist zugeteilt an: - Tobias Schlatter, schlato1 (IT) - Daniel Wassmer, wassmdan (IT)
Fachgebiet:		
DA Datenanalyse	Studiengänge:	
SOW Software	IT Informatik	
Zuordnung der Arbeit :		
InIT Institut für angewandte Informationstechnologie	Infrastruktur:	
	benötigt keinen zugeteilten Arbeitsplatz an der ZHAW	
Interne Partner :		
DeptN Departement N	Industriepartner:	
	Es wurden keine Industriepartner definiert!	
Beschreibung:		
<p>This thesis project is about supporting the exhibition piece "Radiosands" to observe dozens of live radio streams of radios placed in one room: detect keywords, analyze any (societally relevant) patterns in them (like multiple streams e.g. focusing on a specific tragic topic at the same time). Then, as a feedback, control the radios in a way to reinforce certain important aspects (loudness and channel of each radio can be controlled). For example, if one station reports on a tragic accident, all other radios could suddenly turn silent or stop playing happy pop music.</p>		
Informations-Link:		
Unter folgendem Link finden sie weitere Informationen zum Thema: https://github.com/sven2hirsch/radiosands		
Voraussetzungen:		
<p>You can attempt this project in either German or English. No prior knowledge of generative models or sound analysis is required. First contacts with AI and machine learning help, but far more important is your fascination for the topic, your skills in quickly engineering a complex processing pipeline (we will likely use Python, TensorFlow, and Docker to run experiments on a GPU cluster), and your eagerness to experiment systematically.</p> <p>This thesis lends itself very well for a continuation in a subsequent Bachelor thesis. It offers a first glance into research and scientific as well as artistic work, and is also a great predecessor for potentially pursuing Master studies with our team.</p>		
zurück		Logout