1 2 9 0

FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE Ð
COIMBRA

UNIVERSITY OF COIMBRA

Master's Thesis

Automatic classification of ultrasonic vocalizations in rodents: A Neurodevelopmental Study

Author:
Diogo Rafael Mendes Pessoa

Supervisors:
Professor César Teixeira
Professor Miguel Castelo-Branco

*Dissertation presented to the University of Coimbra
in order to complete the necessary requirements to obtain
the Master's degree in Biomedical Engineering.*

July, 2019

This work was developed in collaboration with:

CISUC
ADAPTIVE COMPUTATION

CISUC - Center for Informatics and Systems of the University of Coimbra

1 2 9 0
INSTITUTO DE
CIÊNCIAS NUCLEARES
APLICADAS À SAÚDE
UNIVERSIDADE Ð
COIMBRA

ICNAS - Institute of Nuclear Sciences Applied to Health (University of Coimbra)

Resumo

A evolução da tecnologia levou a um aumento exponencial da quantidade de dados recolhidos e armazenados nas mais diversas áreas. No que diz respeito a estudos bioacústicos, a tendência observada é semelhante, nomeadamente nos estudos desenvolvidos com roedores. Para além disso, neste tipo de estudos tem-se feito sentir um crescente interesse, visto que estes se apresentam como uma ferramenta simples e flexível que permite a avaliação de modelos animais.

Ao longo dos anos, diversos estudos têm salientado a importância do estudo dos sons produzidos por roedores sob o ponto de vista de serem usados como um biomarcador que permita a avaliação da interação entre os animais, nomeadamente em estudos de índole social. Assim, o estudo das vocalizações ultrassónicas tem sido utilizado por diversas ocasiões como meio para caracterizar interação social. Contudo, o verdadeiro significado dos sons produzidos pelos animais assim como que tipo de circunstâncias os levam a produzir certos tipos de sons em detrimento de outros, permanecem desconhecidos.

Tipicamente, o número de vocalizações adquiridas em experiências de índole social atinge facilmente a ordem das dezenas ou centenas por minuto, o que torna a tarefa de as processar manualmente extremamente difícil. Desta forma, a fim de processar tamanhas quantidades de dados em tempo útil, torna-se imperativo o desenvolvimento de software computacional capaz de fazer esse processamento, sendo que deve ser capaz de reconhecer e classificar as vocalizações dos sinais a processar. Desta forma, a principal motivação desta tese está relacionada com necessidade de desenvolvimento de software com as características anteriormente mencionadas.

Nesta tese uma nova metodologia, baseada na entropia espectral, capaz de segmentar vocalizações ultrassónicas foi desenvolvida, sendo capaz de obter valores de *recall* de 97%. Para além disso, foram também construídos diversos modelos de classificação capazes de discriminar entre dez classes diferentes de vocalizações, com 81.1% de taxa de acerto. Tal facto constitui um avanço significativo que diz respeito ao estado da arte, em que são apenas distinguidas cinco classes. Do trabalho desenvolvido neste projeto, resultou como produto final uma aplicação desenvolvida em MATLAB que integra, quer os algoritmos de segmentação desenvolvidos quer os de classificação. Assim, o software desenvolvido permite o completo processamento de sinais áudio não processados, desde a extração de informação relevante, passando pela classificação e reconhecimento de vocalizações. Todo o processo funciona de forma automática.

Palavras-Chave: Segmentação de Vocalizações Ultrassónicas; Classificação de Vocalizações Ultrassónicas; Ultrassons; Processamento de Bio-Sinais; Modelos Animais de Autismo.

Abstract

The evolution of technology led to an exponential increase in the amount of data being collected and stored in the most diverse areas. This is also true regarding bioacoustic studies, namely in rodents. These bioacoustic studies have recently drawn increasing attention since they present a simple and flexible tool to access animal models.

Over the years, several studies have emphasized the importance of studying the sounds produced by rodents from the point of view of being used as a biomarker to evaluate the interaction between animals, mainly in social related studies. Thus, the study of Ultrasonic Vocalizations has been used on multiple occasions in order to characterize reduced social interaction. However, the true meaning of the ultrasonic vocalizations produced by mouse and which circumstances led the animals to vocalize certain sounds to the detriment of others, still remains largely unknown.

During social experiments, the number of vocalizations acquired is typically in the order of the tens or hundreds per minute, which makes it harder to process them manually. Thus, in order to process such a big throughput of data, computer software is required. For this reason, the need for computational software capable of process Ultrasonic Vocalizations, namely recognize and categorize them, is the main motivation behind this work.

This thesis project presents a novel algorithm, based on spectral entropy, capable of segmenting ultrasonic vocalizations, with a recall value up to 97%, as well as several classification methods capable of discriminate among ten different vocalizations with 81.1% accuracy. This presents a significant step forward in relation to the state of the art classification methodologies, which only can classify five different classes. As the final product, a software application was developed in MATLAB, integrating the proposed segmentation and classification solutions in the same fully automated pipeline, capable of processing raw audio files and extract meaningful information.

Keywords: USV Segmentation; USV Classification; Ultrasounds; Bio-Signals Processing; Autism Rodent Models.

Agradecimentos

Em primeiro lugar, gostaria de expressar o meu sincero agradecimento ao Professor Doutor César Teixeira, não só por toda a orientação dada ao longo deste projeto, como também pela enorme disponibilidade que sempre demonstrou no decorrer do mesmo. Este apoio contínuo possibilitou o esclarecimento atempado das minhas questões e a permuta de ideias que, gradualmente, contribuíram para a edificação e desenvolvimento deste projeto.

O mesmo se aplica ao Professor Doutor Miguel Castelo-Branco, a quem expresso, de igual forma, a minha gratidão por todo o apoio prestado.

Gratifico também a Doutora Lorena Petrella, pela cedência de todos os dados subjacentes a este trabalho, bem como por toda a orientação, paciência e apoio concedidos no decorrer do mesmo.

Gostava, ainda, de agradecer ao Professor Doutor Pedro Martins, que apesar de não ser elemento integrante do projeto, imediatamente se prontificou a ajudar-me e aconselhar-me no seu desenvolvimento.

Quero agradecer a todos os meus amigos, com especial enfoque nos meus colegas do laboratório G. 5.2, que desde o início constituíram uma importante fonte de ajuda, apoio e companheirismo.

Deixo, igualmente, um especial agradecimento à minha família, particularmente aos meus pais e à minha irmã, por todo o apoio fornecido e por todos os sacrifícios que fizeram para que pudesse chegar aqui.

Por último, deixo o meu mais sincero agradecimento à minha namorada, Ana Luísa, bem como à sua família. Um muito obrigado por teres estado sempre ao meu lado durante esta caminhada e por todas a experiências vividas. Certamente que muito a ti te devo por ter chegado ao fim desta etapa!

A todos, muito obrigado!

“Unless you try to do something beyond what you have already mastered, you will never grow.”

Ralph Waldo Emerson

Contents

List of Figures

List of Tables

List of Acronyms

ASD Autism Spectrum Disorder

BOW Bag of Words

BOVW Bag of Visual Words

CWT Continuous Wavelet Transform

FSST Fourier Synchrosqueezed Transform

LBP Local Binary Patterns

MSER Maximally Stable Extremal Regions

ML Machine Learning

NLP Natural Language Processing

PCA Principal Component Analysis

PSD Power Spectral Density

STAZCR Short Time Average Zero Crossing Rate

STE Short Time Energy

SIFT Scale-Invariant Feature Transform

SNR Signal to Noise Ratio

STFT Short-Time Fourier Transform

SURF Speeded-Up Robust Features

SVM Support Vector Machine

TFRE Time Frequency Ridge Extraction

USV Ultrasonic Vocalization

WSST Wavelet Synchrosqueezed Transform

Chapter 1

Introduction

This chapter briefly presents the motivation behind the thesis project project as well as the main objectives. Furthermore, a summary of the main contributions is also presented.

1.1 Motivation

For a long time being rodents have been used in biomedical research and play, undoubtedly, an essential role in the progression of scientific knowledge. According to the *National Association for Biomedical Research*, about 95% of all animals used in laboratories are mice or rats [1]. Therefore, is essential to understand how these animals interact and behave in different circumstances in order to create good animal models of several conditions and improve the scientific knowledge.

Despite the fact that is known for a long time that mice produce ultrasonic sounds denominated as Ultrasonic Vocalizations (USVs) or syllables [1], it is still unclear what is the relation between the types of sounds produced and the emotional state of the animals, as well as the relation to the social or biological context in which the animals are inserted. Thus, in this project, several experiments were conducted, under different circumstances (described in chapter 4), in an attempt to submit two different animal models to several emotional states, so that the types of sounds produced by the animals can be related to each emotion or emotions. Furthermore, it also intended to correlate the USVs production with the neurodevelopmental state of each model, in order to access how can USVs be used as a biomarker to detect and describe autism in rodents.

The fundamental knowledge about rodents bio-acoustics has gained progressive interest and can be a very useful and powerful tool in order to develop and formulate new animal models in different behavioral studies, namely autism animals models, which were used in this project. Furthermore, these type of studies also have as an advantage the fact that neither they are invasive or harmful to the animals, as the only thing required to extract the information is the audio recording of the sounds produced by the animals.

[1] https://www.nabr.org/wp-content/uploads/2015/05/Mice-Rats-In-Biomedical-Research-NABR.pdf

1.2 Objectives

The main goal of this project is the development of a classification system that is able to properly detect and categorize rodents USVs. In order to develop this system, several steps must be considered, which lead to the partition of the project in multiple smaller objectives. It is also intended to characterize the vocal repertoire of the two mouse species used in the studies, C57BL/6 (wild type), $Nf1^{+/-}$ (autism model) and a control group, as well as the relations of the produced sounds with the emotional states of the animals and the assessment of the differences between the $Nf1^{+/-}$ and control groups.

The objectives can be listed as follows:

- Development of an algorithm that is able to recognize and segment vocalizations in raw audio signals;

- Creation of an USVs database. This is a two steps process, in which the syllables are isolated first, with the segmentation algorithm, and then, manually classified, so that they can be used in supervised Machine Learning (ML) algorithms;

- Development of classification algorithms that are able to discriminate between the ten different USVs classes (see section 2.4);

- Study the vocal repertoire of each species of rodents involved in this study, mainly to uncover the relations between the types os sounds produced and the social paradigms, among other factors.

- Development of a software capable of processing raw audio files and categorize the syllables that they contain, using the trained machine learning models, as well as the extraction of some relevant characteristics of the vocalizations, like maximum and minimum frequencies, frequency bandwidth, spectral power and energy, among others. The developed software is then intended to be made available online, in Open-Source format.

1.3 Thesis Outline

The document describes all stages of the work developed during the project, from the understanding of the problem to the developed algorithms and according results. The document is organized in Chapters, whose content is briefly described bellow.

1. **Introduction:** Describes the problem that is trying to be solved and what is the motivation and objectives of the project;

2. **Background:** Explains some basic concepts which are key to understand the problem;

3. **State of the Art:** The State of the Art Chapter presents a review of the literature, as well as the listing of some of the available software used in rodents bioacoustic studies;
4. **Dataset:** Describes all the signals acquisition process as well as a the dataset construction process and some statistical considerations regarding itself;
5. **Methods:** Describes all the developed algorithms in all project phases;
6. **Results and Discussion:** Presents the results obtained and reflects on the developed methods and what their potentialities and limitations are;
7. **Conclusion:** Reflects on all the work developed and and what contributions were achieved regarding the main objectives of the work.

1.4 Contributions

This main contributions related to this work are the following ones:

- Creation of an audio signal database with rodents (wild-type and Nf1 groups) USVs with signals acquired form several social paradigms;
- Creation of an USVs database;
- Development of a new method to segment USVs;
- Development of several methods to extract the time-frequency ridge from USVs spectral representations;
- Development of multiple classification systems that are able to differentiate multiple USVs classes;
- Development of a software application, which incorporates an automatic pipeline capable of processing raw audio files and extract relevant characteristics of their USVs.

1.5 Publications

D. Pessoa, L. Petrella, M. Castelo-Branco, C. Teixeira; "Automatic segmentation of ultrasonic vocalizations in rodents"; IFMBE Proceedings, MEDICON 2019 - 15th Mediterranean Conference on Medical and Biological Engineering and Computing (*in printing*).

Chapter 2

Background Concepts

In the Background Concepts Chapter several relevant concepts, related to this thesis, are presented. These concepts allow to better understand the motivations behind the work developed and also help the understatement of some key topics.

It starts with the presentation of the characteristics of the neurodevelopmental disease that is being studied and continues with the impact that rodents have in biomedical research and how the study of the sounds they produce can impact improve the understanding of the animal models.

2.1 Autism

Autism Spectrum Disorder (ASD) is a complex neurodevelopmental disorder characterized by repetitive or stereotyped behavior and impairments in communication and social interaction [2,3]. Emotional processing is highly affected on ASD, explained in part, at least, by abnormalities within the limbic system. Typically, people with this condition show signs of anxiety and mental retardation, delayed learning of language, difficulty in establish conversations and eye contact, bad planning capacity, narrow and intense interests and impaired sensory sensitivities.

The ASD diagnosis is usually based on behavioral analysis. In 2018, the USA *Centers for Disease Control and Prevention* estimated that 1 in 59 children suffer from ASD[1]. Also, in 2014, researchers estimated the lifetime cost of caring for an individual with autism is as great as $2.4 million, which results in almost $90 billion spent annually in costs for autism by the US government, says the *Autism Society*[1].

2.2 Rodents in Biomedical Research

Rodents are mammals of the order *Rodentia* and compose one of the largest mammalian taxa, representing more than 40% of all mammal species [4].

[1] https://www.autism-society.org/what-is/

Rodents were the first animals to be used for scientific purposes and make up to 95% of all the animal models used in laboratory [5]. This species is the most commonly used because their physiology and genetic closely resembles the human one and it does not present significant constrains to its use in laboratory environment. Despite the substantial differences between human and rodents, the similarities are strong enough to give researchers a powerful and versatile mammalian system in which to investigate human diseases [5]. Furthermore, the sequencing of rodent genomes has allowed researchers to reproduce human diseases in rodents through genetic engineering and manipulation. Moreover, the analysis of the interaction between genes, environmental factors and human phenotype are not feasible. This fact stresses the need of using cellular and animal models [3].

In this project, vocalizations from two rodents different strains have been analyzed: the wild-type strain (C57BL/6) and a genetic mice model of autism or $Nf1^{+/-}$ mice (and the non-mutant sibling used as control group). The $Nf1^{+/-}$ mice, developed as a model of neurofibromatosis type I (through mutation in the neurofibromatosis gene), also mimics diverse characteristics of ASD.

2.3 Rodent Bioacoustics

Even thought several studies have been conducted on rodents throughout the years, there is still a lot to understand regarding mouse vocal communication and how it affects interaction between them. Their social behavior can vary a lot from species to species, and the study of Ultrasonic Vocalizations (USVs) would provide important insight about how animals interact regarding their territory, sexual receptivity, food, predators, among other aspects [4].

Mice produce a wide and complex repertoire of sounds in the most different social contexts, ranging from the audible to the ultrasonic domain, i.e., up to 200 kHz or more. Although these sounds are often used in biomedical and scientific research as an index of sociability between animals, their biological significance is not fully understood [6–9].

As it happens with most terrestrial mammals, rodents produce sound with their larynx, a organ located into the throat (see Figure 2.1), and through an external air stream. When the air stream passes through the larynx, changes in the air particles are made and sounds are produced. Besides that, the modulation of sounds produced can be controlled with the combination of expiration, partial or full closure of the laryngeal glottis, and the use of a supralaryngeal component of the vocal tract [4]. All of the mechanisms described above are controlled by a total of forty muscles and their according nerves [4].

[2] https://doi.org/10.7554/eLife.19153.003

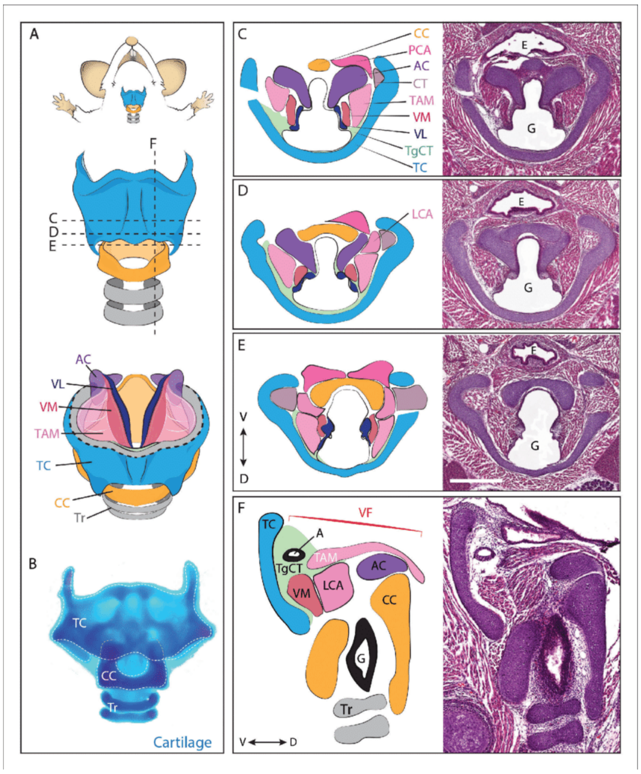


Figure 2.1: Mouse larynx anatomy[2]. Abbreviations: (AC) Arytenoid Cartilage, (CC) Cricoid Cartilage, (CT) Cricothyroid muscle, (E) Esophagus, (G) Glottis, (L) Larynx, (LCA) Lateral Cricoarytenoid muscle, (PCA) Posterior Cricoarytenoid muscle, (T) Tongue, (TAM) Thyroarytenoid Muscle, (TC) Thryoid Cartilage, (TgCT) Thyroglottal connective tissue, (Tr) Trachea, (VL) Vocal Ligament, (VM) Vocalis Muscle, (VF) Vocal fold. (V) and (D) indicate dorso-ventral axes..

2.4 USVs Classes Definition

One of the biggest difficulties when classifying USVs is that syllable types differ between rodent strains. Furthermore, it has also been shown that vocalizations differ with the environmental/social situation in which the animals are inserted, and also with the age and sex of the animals.

Based on our acoustic records, we found out that the syllable types defined by Scattoni [7] and Grimsley [8], offer a good coverage of the observed USVs repertoire. Thus, the definitions given by these authors have been used in this study.

The classes considered in this work are listed and defined below.

- **Complex**: monosyllabic syllable with two or more directional changes in frequency >6 kHz (Grimsley), Figure 2.1a;

- **1 Frequency Step**: syllables with two elements, in which the second element was $\geq$10 kHz different from the preceding element and there was no separation in time between steps (Grimsley), Figure 2.1b;

- **Multiple Frequency Steps**: syllables were two or more instantaneous frequency changes appear as a vertically discontinuous “step” on the spectrogram, but with no interruption in time (Adapted from Grimsley and Scattoni), Figure 2.1c;

- **Upward**: syllables with frequency upwardly modulated with a frequency change ≥6 kHz (Grimsley), Figure 2.1d;

- **Downward**: syllables with frequency downwardly modulated with a frequency change ≥6 kHz (Grimsley), Figure 2.1e;

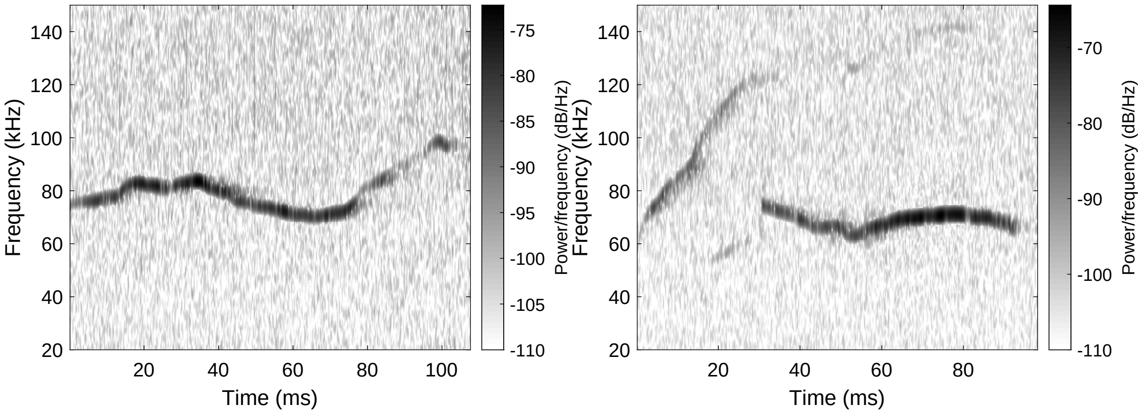
- **Flat**: syllable with constant frequency, with frequency modulation <6 kHz (Grimsley), Figure 2.1f;

- **Short**: syllables that lasts <5 ms (Grimsley), Figure 2.1g;

- **Chevron**: syllables shaped like an inverted U, where the highest frequency was at least 6 kHz greater than the starting and ending frequencies (Grimsley), Figure 2.1g;

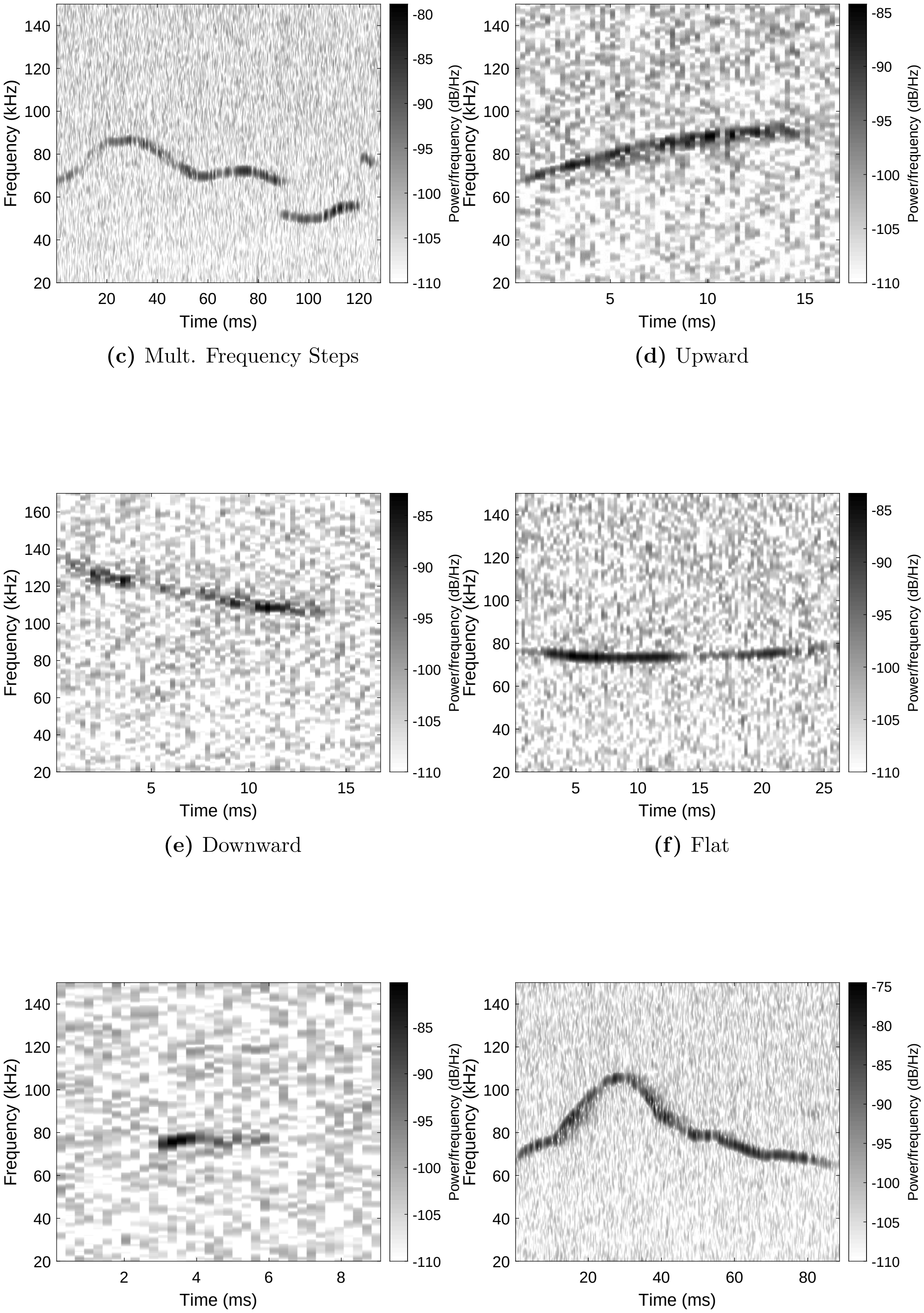
- **Reverse Chevron**: syllables shaped like a U, where the lowest frequency is at least 6 kHz lower than the starting and ending frequencies (Grimsley), Figure 2.1i;

- **Composite**: were formed by two harmonically independent components, emitted simultaneously (Scattoni), Figure 2.1j.



(a) Complex

(b) 1 Frequency Step



(c) Mult. Frequency Steps

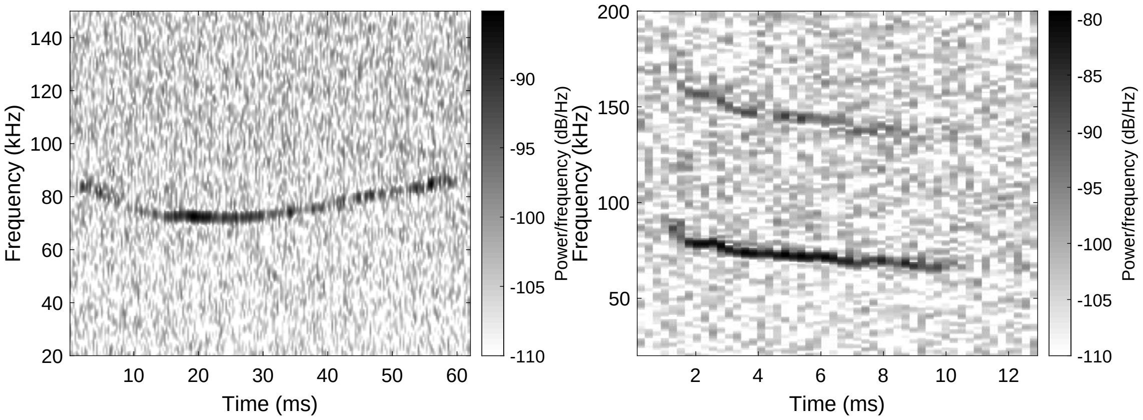
(d) Upward

(e) Downward

(f) Flat

(g) Short

(h) Chevron



(i) Reverse Chevron

(j) Composite

Figure 2.1: Time-frequency representation of the 11 types of USVs considered in this study.

Chapter 3

State of the Art

With the increase of computational power, which increased the capability of processing large set of files, bioacoustic studies have drawn more and more attention. These types of studies have been conducted in multiple animals species such as birds, frogs, whales, dolphins, elephants, and many others [10–15], and have focused not only in distinguish the different types of sounds produced by the animals and the according context in which the they are produced, but also in discrimination between animal types within the species themselves.

Throughout the course of the years, rodents bioacoustics studies have also experienced an increasing demand, since mouse are the most used animals in laboratories. Several researchers have unveiled the potential of mouse Ultrasonic Vocalizations (USVs) analysis in further improvement of the knowledge about rodents and how they interact in different circumstances, being a very interesting and powerful tool, namely, in social interaction studies.

When talking about bioacoustic studies, there are usually two main steps involved, being the signals segmentation and the study, processing and analysis of the sounds themselves. Even tough there are several tools available related to mouse USVs analysis, most of them only allow to group vocalization based on clustering processes, and the ones which allow to classify the vocalizations, can only distinguish a small number of sounds, which is far from ideal since mouse can produce a wide and diverse spectrum of vocalizations in the most different situations.

3.1 USVs analysis and Behavioral Patterns

Several studies have shown that rodents produce different types of sounds accordingly to the social or environmental context, which led to the hypothesis that the repertoire produced by the animals can be used as a biomarker in neuropsychiatric mice models of neuropsychiatric disorders [7, 9].

Some of the more common experiments in USVs social and biological paradigms studies are related to sexual behavior and courtship, and demonstrated that USVs are used either to attract and/or maintain close proximity to females and facilitate mating [16,17]. USVs are also preponderant in social cohesion by triggering and maintaining the interaction between two individuals of the same sex [18]. USVs production alterations have also been associated to genetic abnormalities in several neurotransmitter systems [19].

3.2 USVs Segmentation

Typically, in studies related to bioacoustics analysis, the first step consists in the audio signals segmentation, in order to isolate the syllables or calls produced by the animals. To this end, several algorithms have been proposed in the literature, as listed in Table 3.1.

The segmentation process consists essentially in the partition of the audio signal in smaller pieces containing the USVs, in order to isolate these specific events. This is a critical step to undertake, since it will allow the individual processing and information extraction of each vocalization on its own.

One of the major hurdles in automatic detecting of USVs is the very low Signal to Noise Ratio (SNR), as USVs signals often have broadband interference or ambient noise, which can partially mask the signals. Thus, powerful and general algorithms are required to handle the task.

Table 3.1: Summary table of the different audio segmentation methods.

Method	Authors
Faster R-CNN	Coffey et al. 2019 [20]
Voice Activity Detector	Ramirez et al. 2007 [21]
Harma's Method	Harma et al. 2004 [22]
STE & STAZCR	Sakim et al. 2014 [23]
Signal Bandwidth	Zala et al. 2017 [24]
Time varying parameters	Holly et al. 2005 [25]
Hilbert Follower	Potamitis et al. 2014 [26]

The Faster R-CNN, used in DeepSqueak [20] toolbox, uses a two stage methodology to isolate the individual calls of the animals in the audio files. First, short segments of audio files, with variable length previously defined, are converted into sonograms which are then passed by an Faster-RCNN image object detector. This step detects all the possible interest areas, which may or may not be vocalizations. Then, in the second stage, the results produced by the Faster-RCNN are passed to a second network which will classify the proposed regions into real vocalizations or background noise. Thus, both classification networks used in each of the two stages need to be trained with annotated data.

The Voice Activity Detector, used in MUPET [27] software, is an audio activity detector proposed by Ramírez et al. [21]. The syllable detector divides the audio segment in multiple frames and then measures the SNR and other parameters of each one. Then, with the use of a threshold it is determined whether a frame is voiced or unvoiced.

The Harma's method was first proposed by Harma in 2004, and has been used in multiple studies of bioacoustic in different kinds of animals [28–31]. The method is a syllable segmentation algorithm based on the iterative amplitude-frequency information. Essentially, this method consists in the computation the spectrogram of an audio segment using Short Time Fast Fourier Transform ($S(f,t)$) and then finding f_n and t_n, such that $|S(f_n,t_n)|$ is the maximum value in the spectrogram. Those values are stored as $w_n(0) = fn$ and $a_n(0) = 20log(|S(f_n,t_n)|)$. Then, starting from $|S(f_n,t_n)|$ for $t > t_0$ and for $t < t_0$, the maximum peak of $S(f,t)$ is traced back and forward until a stopping criteria is met. Once this criteria is met, the initial and final instants of the calls are stored. After that, the process is repeated iteratively.

Short Time Energy (STE) and Short Time Average Zero Crossing Rate (STAZCR) [32,33] is also an algorithm developed to segment bioacoustic signals. The STE is computed in order to estimate, with a threshold, the pieces of the signal that contain sounds (voiced segments) and then, STAZCR is employed as a part of the front-end processing. This algorithm was also used in multiple bioacoustic studies [23].

The Signal Bandwidth algorithm is part of A-MUD interface [24] (*Automatic Mouse Ultrasound Detector*), which is available for free. The first step of this segmentation algorithm aims to reduce the noise in the signal with pre-whitening techniques. Then, two specific thresholds, that constitute the narrowness of the signal's bandwidth, are estimated. The signal's bandwidth is defined as the number of frequency bins for which a certain ratio of the total spectrum energy was achieved. Lastly, using the bandwidth previously determined, the vocalizations are identified, being that in the end of the process several temporal corrections are applied.

In the method used by Holy and Guo [25], Ivanenko [34] and Arriaga [35], syllables are segmented by computing three time-varying parameters from the audio spectrograms: the mean frequency, the "spectral purity" (fraction of total power concentrated into a single frequency bin), and the "spectral discontinuity". After that, syllables are identified if those three measured parameters met certain conditions.

The Hilbert follower [26] is essentially an audio activity detector which segments the signals by following the characteristic shape of syllables envelop. This algorithm computes the Hilbert envelope of the signal $x(n)$, $x_h(n) = Hilbert(x(n))$, and then, if the envelope is above a threshold, it is considered to be audio activity, and, therefore, a vocalization.

Besides all the algorithms mentioned and briefly described above, there is also available commercial software capable of segmenting USVs, such as *Ultravox*[1] and *SASLab Pro*[2]. However, the algorithms used in these products are not disclosed.

The performance values presented in Table 3.2 for DeepSquek, MUPPET and UltraVox were obtained from DeepSqueak article [20].

[1] https://www.noldus.com/animal-behavior-research/products/ultravox-xt

[2] https://www.avisoft.com/soundanalysis.html

Table 3.2: Summary table of the different audio segmentation methods performance (**N.a.-not available**).

Tool	**Performance**
DeepSqueak [20]	Precision≈0.97 Recall≈0.9
MUPPET [27]	Precision≈0.95 Recall≈0.93
UltraVox[3]	Precision≈0.95 Recall≈0.82
Time varying parameters [25]	Precision≈N.a. Recall≈0.95

3.3 USVs Classification

Even tough there are several tools available to study and analyze mouse USVs, most of them take a non unsupervised approach and use, mostly, clustering methodologies. These means that these tools do not attribute a specific label to each USVs. Instead, with clustering, it is possible to group the USVs into different clusters, or groups, based the similarity of a certain set of descriptors, trying to maximize the inter-clustering separability and minimizing the intra-clustering variability.

One of the major hurdles in USVs classification is the fact that it does not exist a standard set of well defined vocalizations. Several authors in the literature define the types of vocalizations that they observe in their own experiments with their own species of rodents, which might or might not correspond to the ones observed in other species of rodents by other authors. Furthermore, there are even some types of vocalizations with different definitions depending on the author. Taking into account the previous considerations, it is easy to understand why it is so difficult to establish and recognize all types of vocalizations that exists, since they vary a lot, mostly between species.

In spite of being mainly a tool with clustering capabilities, the DeepSqueak software [20] also provides supervised classifiers, namely neural networks. These networks are able to distinguish between five types of different vocalizations (Split, Inverted U, Short Rise, Wave, and Step), which are by no means comprehensive enough to describe the full spectrum of sounds produced by mice, as the authors refer, since they were able to recognize up to twenty different types of syllables with their clustering methods. Note that the classes identified in [20] do not have a direct correspondence to the ones used in this work. Holy and Guo [25] were also able to categorize the same 5 different syllables identified with DeepSqueak, however, in neither of the cases is given any classification metric or result that would allow to access the algorithms performance.

As it happens with the segmentation algorithms, there is also available commercial software capable of classify USVs such as *SASLab Pro*[4] and *SONOTRACK*[5]. However,

[4] https://www.avisoft.com/soundanalysis.html

[5] https://www.metris.nl/en/products/sonotrack

the algorithms used in these products are not disclosed, neither are the performance results regarding the classification available.

3.4 USVs Software and Toolboxes

As mentioned before, there are some tools available used to support studies and research on mouse USVs and bioacoustic studies in general. In this section, a brief description of these tools and the characterization of their features is made, as well as the description of the main strengths and limitations of each one.

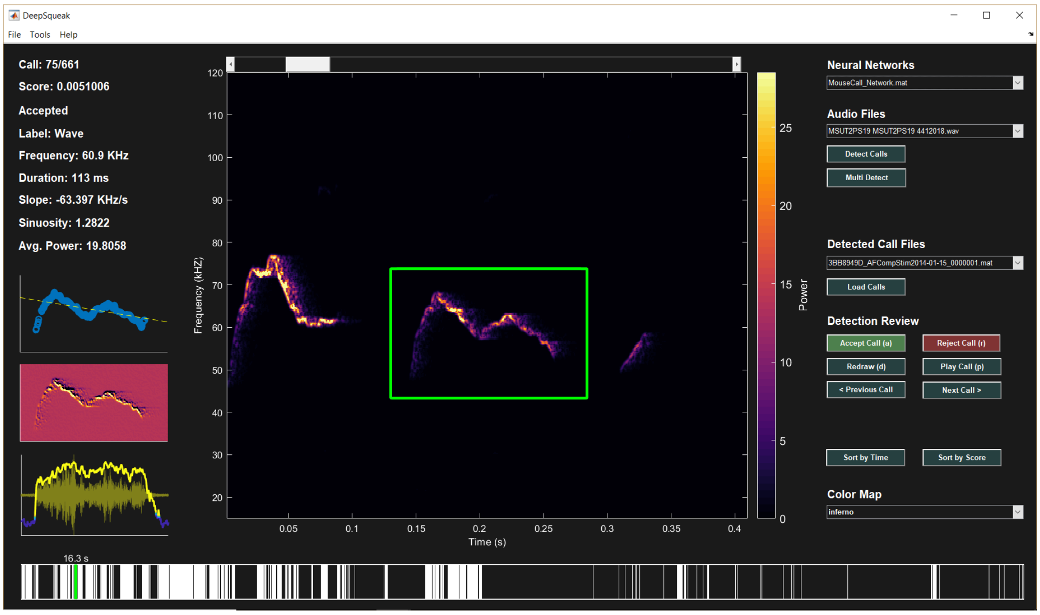
3.4.1 DeepSqueak

DeepSqueak is a fully graphical MATLAB package (see Figure 3.1) which allows for the detection and classification of rodents USVs. The interface's aims to allow non-experts easy entry into USVs detection and analysis.

The interface allows the segmentation of audio signals (in .WAV or .FLAC format) using Faster R-CNNN object detection networks and output syllable statistics in spreadsheets for further analysis. It is possible to output to a spreadsheet characteristics of the vocalizations such as the minimum, maximum and mean frequency of the vocalizations, duration, slope, sinuosity and power. Besides that, to each vocalization is also given a confidence score, that is, a score that translates the confidence of the system that it is a real vocalization. DeepSqueak also offers the possibility to classify the vocalizations, as was mentioned before. However, this classification is quiet limited, since it only allows to discriminate between five different types (Split, Inverted U, Short Rise, Wave, and Step), and is far from comprehensive, as the authors refer.

Notwithstanding, DeepSqueak excels due to its capability of unsupervised syllable clustering. The interface offers the possibility of grouping, with two different clustering techniques (k-means and adaptive resonance theory neural network), the vocalizations which where detected in the segmentation phase. In order to be grouped, vocalizations are characterized by three different features, shape, frequency and duration. These descriptors are then passed to the clustering algorithm and the vocalizations are grouped according to them. The weight for each of the features used can be adjusted by the user. With k-means algorithm, the user has the possibility of choosing the number or clusters of it can be chosen automatically estimated using the elbow method. On the other hand, the algorithm of adaptive resonance theory neural network does not require any input.

[6] https://github.com/DrCoffey/DeepSqueak



Call: 75/661

Score: 0.0051006

Accepted

Label: Wave

Frequency: 60.9 KHz

Duration: 113 ms

Slope: -63.397 KHz/s

Sinuosity: 1.2822

Avg. Power: 19.8058

Neural Networks: MouseCall_Network.mat

Audio Files: MSUT2PS19 MSUT2PS19 4412018.wav

Detected Call Files: 3BB8949D_AFCompStim2014-01-15_0000001.mat

Color Map: inferno

Figure 3.1: DeepSqueak Interface[6].

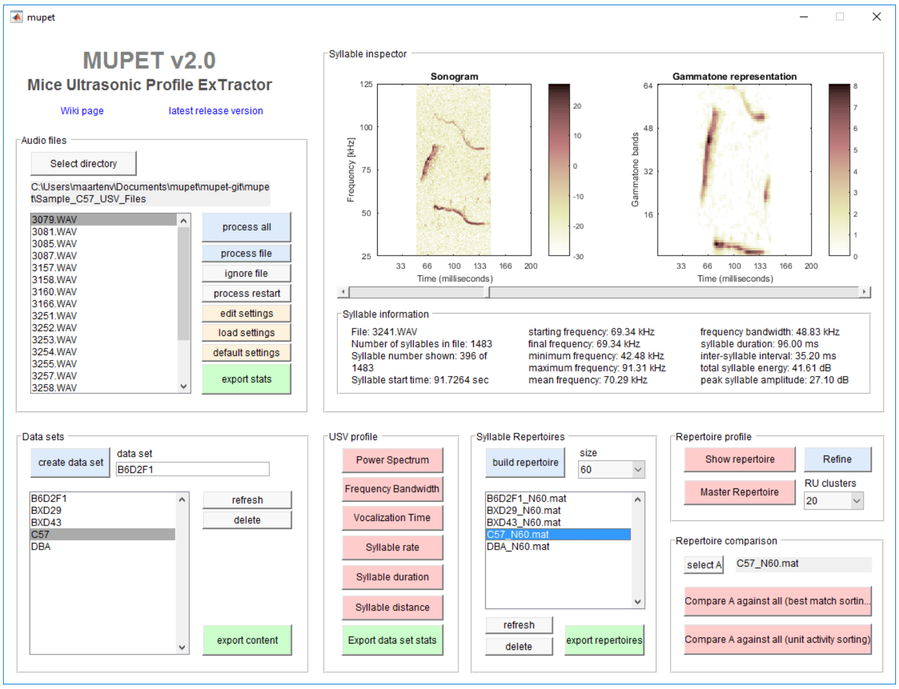
3.4.2 MUPET

MUPET (see Figure 3.2), which stands for Mouse Ultrasonic Profile ExTraction, is an open-access software that automatically generates mouse vocalization repertoires. The software uses signal processing techniques to perform rapid, unsupervised analysis of mouse ultrasonic vocalization repertoires, including unbiased syllable discovery [27]. The MUPET software presents five fundamental capabilities that allow it to process and study the sounds produced by rodents, being the syllable detection capability, acoustical dataset analysis, syllable repertoire building, measure of the similarity between different syllables repertoires and, at last, cluster analysis of the repertoires.

In order to segment the audio signals, the software uses a voice activity detector, based on the work of Ramírez et al. [21], which will split the audio file in smaller pieces and classify them into voiced or unvoiced segments. After the segmentation of the signals, it is possible to build vocalizations datasets in which USVs can be grouped with clustering techniques according to their frequency, duration, shape and amplitude. Furthermore, other of the greatest features of MUPET is the possible of comparing the similarity between different repertoires, which can be a very useful tool when comparing two different strains of mouse, for instance.

Besides all of the functionalities mentioned above, the interface is also able to generate *.csv* spreadsheets with several characteristics of the detected vocalizations, such as the initial and final frequencies, the maximum and minimum frequencies, among others.

[7] https://sail.usc.edu/mupet/



C:\Users\maartenv\Documents\mupet\mupet-git\mupet\Sample_C57_USV_Files

3079.WAV
3081.WAV
3085.WAV
3087.WAV
3157.WAV
3158.WAV
3160.WAV
3166.WAV
3251.WAV
3252.WAV
3253.WAV
3254.WAV
3255.WAV
3257.WAV
3258.WAV

File: 3241.WAV
Number of syllables in file: 1483
Syllable number shown: 396 of 1483
Syllable start time: 91.7264 sec

starting frequency: 69.34 kHz
final frequency: 69.34 kHz
minimum frequency: 42.48 kHz
maximum frequency: 91.31 kHz
mean frequency: 70.29 kHz

frequency bandwidth: 48.83 kHz
syllable duration: 96.00 ms
inter-syllable interval: 35.20 ms
total syllable energy: 41.61 dB
peak syllable amplitude: 27.10 dB

data set: B6D2F1

B6D2F1
BXD29
BXD43
C57
DBA

size: 60

B6D2F1_N60.mat
BXD29_N60.mat
BXD43_N60.mat
C57_N60.mat
DBA_N60.mat

RU clusters: 20

select A: C57_N60.mat

Figure 3.2: MUPET v2.0 Interface[7].

3.4.3 A-MUD

A-MUD (Automatic Mouse Ultrasound Detector or A-MUD 1.0) in a segmentation algorithm implemented in STx (S_TOOLS-STx version 4.2.2), a software from the Acoustic Research Institute (Austria), used to process large quantities of audio data [24]. The A-MUD algorithm, as described above, is composed by a multi step process, which will determine the start and ending point of each vocalization. Furthermore, the software is also able to measure a series of parameters of each segmented vocalization, such as the mean frequency, frequency bandwidth, mean amplitude, the frequency and amplitude, accordingly, at the starting and ending points as well as the maximum and minimum frequencies and amplitudes. The A-MUD algorithm performance values can be seen in Figure 3.3

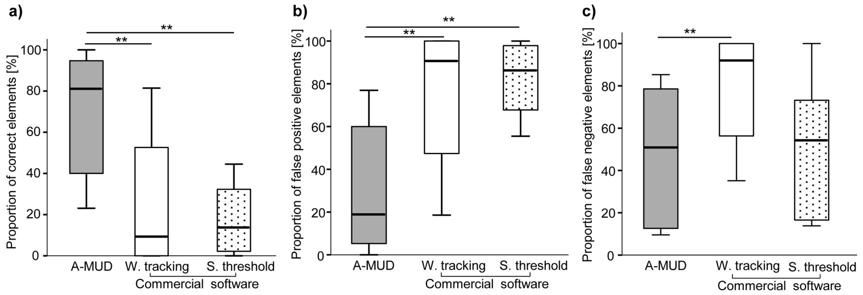
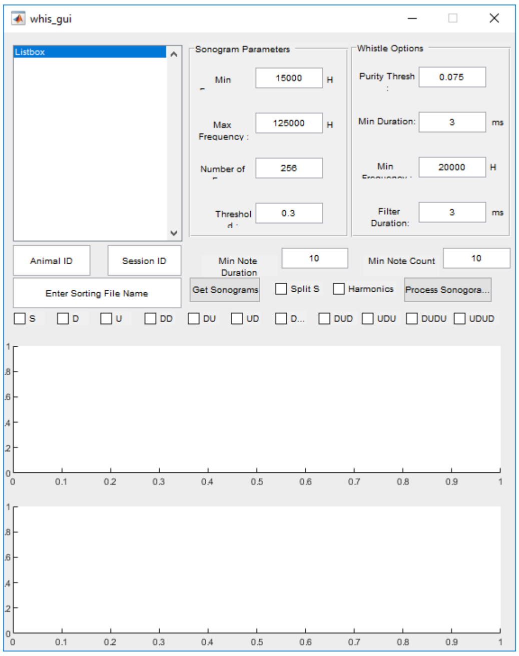


Figure 3.3: A-MUD performance results (Comparison between A-MUD and commercial software)[8].

[8] https://doi.org/10.1371/journal.pone.0181200.g001

3.4.4 Mouse Song Analyzer 1.3

Mouse Song Analyzer v1.3 is MATLAB program developed by Arriaga et al. (2012) [35], modified from code originally written by Timothy E. Holy [25]. The software allows to represent the sonograms of the mouse ultrasonic vocalizations as well as the extraction of several characteristics and the classification according to Holy and Guo [25], Arriaga et al [35] and Scattoni [7] classifications.



Min: 15000 H
Max Frequency : 125000 H
Number of: 256
Threshold : 0.3
Purity Thresh : 0.075
Min Duration: 3 ms
Min Frequency : 20000 H
Filter Duration: 3 ms
Min Note Duration: 10
Min Note Count: 10

Figure 3.4: Mouse Song Analyzer 1.3 Interface.

3.4.5 VoICE

VoICE is a semi-autonomous pipeline, developed in MATLAB and R, which allows grouping vocal elements through the creation of a high dimensionality datasets, using spectral scoring similarity between all vocalizations within a recording session [36].

The method performs automated, hierarchical clustering of similar shapes. However, the assignment of the classes to each vocalization is made manually, having the advantage of the most similar syllables being already grouped. The identified vocalizations are based on the ones described by Scattoni [7, 37].

One of the main weaknesses of the software, is the fact that the pipeline does not have the capability of segment raw audios signals, whereby, all the vocalizations must be previously selected and passed as individual separated files to the system.

Table 3.3: Summary table of the different Toolboxes and software on USVs.

	DeepSqueak	MUPET	Mouse Song Analyer v1.3	VoICE	A-MUD
References	[20]	[27]	[25]	[36]	[24]
Platform	MATLAB	MATLAB	MATLAB	MATLAB and R	S_TOOLS-STx version 4.2.2
Input	.WAV or .FLAC Audio files	.WAV Audio files	.WAV Audio files	Separated clips of vocalizations (.WAV files)	.WAV Audio files
Syllable Detection	Faster R-CNN	Voice activity detector	Time Varying Parameters	Not Supported	Signal Bandwidth
Species	Mouse	Mouse	Mouse	Mouse and Bird	Mouse
Highlights	Quality USV detection and classification at high speeds. Grouping of new vocalizations with clustering, with vocalizations information extraction.	Output regarding syllables information as well as syllables datasets construction and comparison. Automated, unbiased discovery of recurring syllable shapes using machine learning.	Automated, rule-based categorization of syllable shapes, and syllables information extraction.	Automated, hierarchical clustering of similar shapes which facilitates manual, rule-base categorization of a smaller number of cluster eigencalls	Syllable segmentation algorithm toolbox.

Chapter 4

Dataset Description and Acquisition

In this chapter the system used to acquire the audio signals in this work is described, as well as its equipment. The social paradigms considered in the project are also defined. Furthermore, the dataset annotation process is also explained. Lastly, some statistical tests were conducted, which are also presented.

4.1 Signals Acquisition System

The recording system used to acquire the mouse audio calls consists on a multi-channel recording system (*Avisoft-UltraSoundGate 416H, Avisoft Bioacoustics*) with four channels working with two microphones (*microphone CM16 / CMPA, Avisoft Bioacoustics*), one for each test chamber (see Figure 4.1). The signals can be digitized at a maximum sampling frequency of 700 kHz. Associated software allows the definition of the recording parameters, the real time signal/spectrum visualization and the recording of the signals in the PC memory. Moreover, a playback system, consisting of a speaker (*Ultrasonic Speaker Vifa, Avisoft Bioacoustics*) and a D/A converter(*UltraSoundGate Player 116 (single channel), Avisoft Bioacoustics*), was acquired from the same manufacturer. This playback system is used to reproduce acoustic stimuli. Tests are video-recorded with a webcam positioned at the top of the test cabin, and using a red light that enables video recording while minimizing undesired effects. The sound-attenuating cabin was projected with 55 cm x 50 cm x 70 cm (H x D x W), to provide enough space for the experimental montage. It was constructed of acrylic plates with a wall thickness of 1.5 cm, to block the external sound. Moreover, the internal side of the cabin was fully covered by sound absorbing foam. The cabin was also designed with a frontal door and holes for ventilation and wiring. An additional acrylic cage was constructed, where mice are placed during tests. It is half separated by an acrylic plate with holds in the inferior part. During tests two animals are placed one in each compartment (with one microphone) for the identification of each mice Ultrasonic Vocalizations (USVs), while the holes allow them to keep communicating. Preliminary tests confirmed the good performance and versatility of the assembled system.

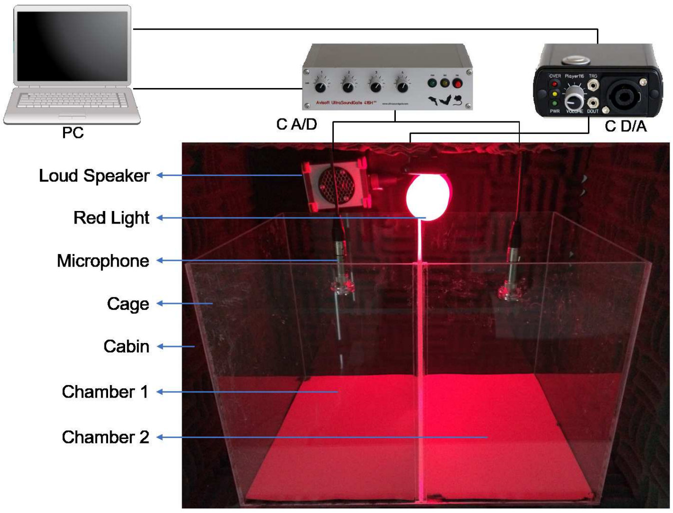


Figure 4.1: USV recorder system.

4.2 Paradigms in Signal Acquisition

Having in consideration that multiple paradigms have been used in the literature to record rodent's social interaction, a set of experiments, was designed to induce and study diverse emotional states. Some of the paradigms are common among the ones used in the literature, while others present themselves as new approaches.

The basic emotions can be defined by Robert Plutchik's wheel of emotions as; joy, trust, fear, surprise, sadness, disgust, anger and anticipation. One paradigm was implemented for seven of these basic emotions except for trust. Moreover, another paradigm for the homeostatic emotion hunger was also conducted.

All the tests in which the signals were acquired were conducted in pairs were the animals where separated by an acrylic plate and communicating through holes (except for Joy in juvenile mice and for Anger test in adult mice). In all tests two audio channels were recorded, having each mouse a microphone compartment (see Figure 4.1). For all tests (except for Joy and Anger tests) mice are left 10 minutes for adaptation before start recording.

Tests were implemented on both, males and females, and at two time-points: juvenile mice (22-38 days of age) and adult mice (60-82 days of age), except for the anger test that was only implemented in adult males, since neither juveniles nor females presented aggressive behavior.

The tests for anticipation, surprise and sadness (or disappointment) are related with each other. It is based in the well-known capacity of mice to anticipate (learn) situations when trained with a repeated stimulus. Here, animals were trained with high-frequency acoustic beeps (20 kHz beeps during 1 minute), followed by food introduction.

All the paradigms are defined bellow.

Paradigms

- **Joy**: The test differs for juvenile and adult mice. Juvenile mice: mice are highly social animals, and juvenile mice interact for play. To evoke vocalizations during the test, animals are isolated on individual cages the day before. In the testing moment the animals (of the same gender) are put together for free social interaction. Adult mice: it is well known from the literature that mice vocalize to attract the couple. On this sense for the test, male and female are simultaneously introduced in each compartment of the cage. This test is repeated in two different days, to give the animals the chance to familiarize with each other.

- **Disgust**: On this paradigm was used an olfactory stimuli to induce disgusting emotion and related vocalizations. A previous study showed that benzaldehyde evokes an adverse reaction in mice [38]. In this test, two familiar mice (of the same sex) are put on each compartment of the cage, with suspended empty falcon tubes. After 120s of recording the tubes are substituted by others containing cotton with benzaldehyde (bitter almond essence).

- **Fear**: Mice shown emotional responses in the presence of predators odor [39]. Thus, in this test was used cat's urine to induce fear stimuli. Two familiar mice (same sex) are put on each compartment of the cage, with suspended empty falcon tubes. After 120s of recording the tubes are substituted by others containing cotton soaked with cat urine.

- **Anger**: Mice are highly territorial animals. Usually, after some minutes in a territory they recognize and adopt it.When an unknown mouse tries to occupy the same space, it is common for the resident to reacts violently in order to protect their space. This adverse reaction to intruders has also been documented in several studies, being a strong vocalization stimuli [40]. On this test, one male is putted in the recording cage and left for habituation for 30 minutes. The record starts, and 120 s latter an unfamiliar mouse (intruder) is introduced inside a cylindrical wire cage.

- **Anticipation**: In this test, the food is withdrawn from the cage ± 3h before the test. At the test moment a pair of familiar mice is put in the cage. After 120s of recording the acoustic stimulus is reproduced and after 15s interval a piece feed is introduced on each compartment. Mice are trained along 3 days and the anticipation is assessed at the 4th day, following the same procedure.

- **Surprise**: It is implemented in the consecutive day of anticipation test. The procedure is similar, with the exception that, after the acoustic stimuli, sunflower seeds are introduced (instead of mice feed). Mice were previously familiarized with sunflower seeds, and it was choose since the animals have shown high interest for it.

- **Sadness/Disappointment**: It is implemented in the consecutive day of the surprise test. The procedure is similar, with the exception that, after the acoustic stimuli, no food was introduced.

- **Hunger**: The food is withdrawn from the cage approximately 6h before the test. Two familiar mice (same sex) are put on each compartment of the cage, with suspended empty falcon tubes. After 120s of recording the tubes are substituted by open tubes containing mice feed, but kept inaccessible, so that the mice just smell it.

- **Background**: This test serves as background for disgusting, fear and hunger tests, and is implemented to elucidate if the procedure of opening the frontal door and changing the tubes induces vocalizations. Two familiar mice (same sex) are put on each compartment of the cage, with suspended empty falcon tubes. After 120s of recording the tubes are removed and place again.

4.3 Dataset Annotation

The labeling process of any dataset it is always a laborious and exhaustive task. However, in order to develop any kind of supervised classification algorithms or methodologies, it is mandatory to have labeled data. So, it was necessary to manually annotate the vocalizations detected by the segmentation algorithm (see section 5.1). All the vocalizations used in the classification algorithms were manually annotated.

Since two microphones are capturing the audio simultaneously, when the USV are being annotated, the two channels must also be analyzed together. This is a crucial point since most of the time the sound produced by a mouse is captured by the two microphones, which may cause interpretation problems. This may happen when one of mouse vocalizes to close too the communication hole or even when one of animals simply emit a very powerful sound. In order to handle this problem, the signals from both channels were analyzed at the same time and whenever a vocalization was present in both recordings it was manually chosen which channel to keep, according to the quality of the vocalization in each of the channels.

To annotate all the vocalizations, an application was developed using MATLAB 2018b (see Figure 4.2). The interface allows to classify the vocalizations of only one signal at a time, however, it also shows the complementary channel in the same temporal instants. Even tough only a channel can be classified at a time, whenever it happens a vocalization to be present in both channels, the user has the possibility to choose which vocalization to keep, if the vocalization of the main channel being classified or the complementary channel. When this happens, the start and end points of the vocalization on the complementary channel are also shown, as it is possible to observe in Figure 4.2.

Furthermore, in order to facilitate the annotation process, there are two sub-windows on the right side to give auxiliary information to the user. On the upper sub-window, it is displayed a cleaner version of the time-frequency representation from the initial to the final instants of the vocalizations, while on lower sub-window it is used an algorithm to extract the time-frequency ridge (contour) of the vocalization. With that contour vector some useful parameters are determined, such as the initial, final, maximum and minimum frequencies as well as the duration of each syllable, in order to facilitate the process of manual association to a certain class.

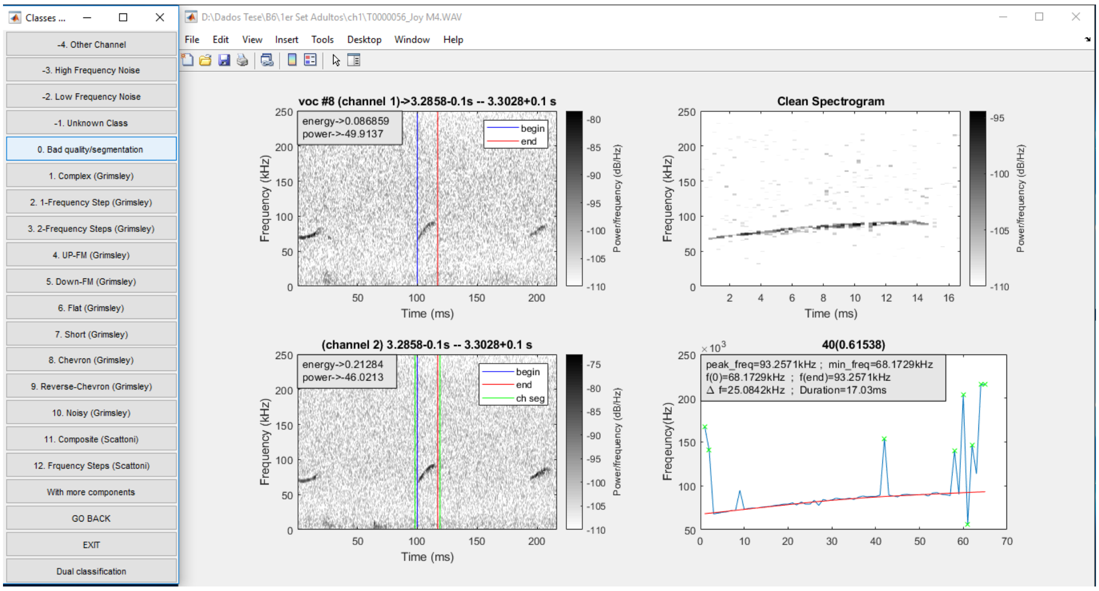


Figure 4.2: Vocalizations Labeling Interface.

4.4 Dataset Analysis and Statistics

Using the described system in section 4.1, a total of 666 ($.WAV$) audio signals were acquired in different social conditions (see section 4.2) and the three different group of animals with different ages, C57BL/6 and $Nf1^{+/-}$ and the according control group (autism model group), young and adults accordingly. In total, almost 81 hours of social interaction between mice were acquired, as listed in Table 4.1.

Table 4.1: Summary table of the recording hours by paradigm (hours).

Experiment	**Total Time (h)**	**C57BL/6 (h)**	$Nf1^{+/-}$ **(h)**	**Adults (h)**	**Youngsters (h)**
Joy	11.3	5.72	5.58	6.94	4.35
Anticipation	28.52	12.15	16.37	15.76	12.76
Surprise	6.81	3.05	3.76	3.96	2.85
Fear	5.16	2.89	2.27	2.94	2.22
Hunger	6.62	2.57	4.05	3.69	2.93
Disgusting	6.25	2.71	3.54	3.12	3.13
Disappointment	7.27	3.04	4.23	3.96	3.30
Anger	2.57	1.50	1.07	2.57	0.00
Background	6.34	1.79	4.55	2.92	3.42
Total	80.83	35.42	45.42	45.87	34.97

From the total 666 signals, 357 were manually annotated, according to the process described in section 4.3. In Table 4.2, is possible to observe how many signals were classified by social paradigm. With the manual annotation process, a dataset composed by 4633 USVs was created. During the manual annotation process, dubious vocalizations or with very low quality were excluded. The USVs database distribution by class and paradigm is presented in Table 4.3.

Table 4.2: Number of acquired and classified signals by paradigm.

Paradigm	Total	Classified
Joy	72	65
Anticipation	236	30
Surprise	56	31
Fear	44	38
Hunger	60	43
Disgusting	60	51
Disappointment	60	41
Anger	22	16
Background	56	42
Total	666	357

Table 4.3: Vocalizations distribution by class and paradigm.

Classes	Joy	Anticipation	Surprise	Fear	Hunger	Disgusting	Disappointment	Anger	Background	Sum
Complex	40	0	0	2	0	3	0	0	1	**46**
1 Freq. Step	605	1	3	9	13	9	10	1	11	**662**
M. Freq. Steps	111	0	2	1	6	9	11	3	11	**154**
Up	1632	4	9	28	29	25	14	23	16	**1780**
Down	127	14	24	77	40	68	6	1	11	**368**
Flat	739	9	27	40	28	23	21	54	6	**947**
Short	223	4	13	46	54	23	4	4	19	**390**
Chevron	113	0	2	8	3	4	3	0	2	**135**
Rev. Chevron	33	0	5	9	0	5	1	0	1	**54**
Composite	22	11	6	23	15	4	11	0	4	**96**
Sum	**3645**	**43**	**91**	**243**	**188**	**173**	**81**	**86**	**82**	**4632**

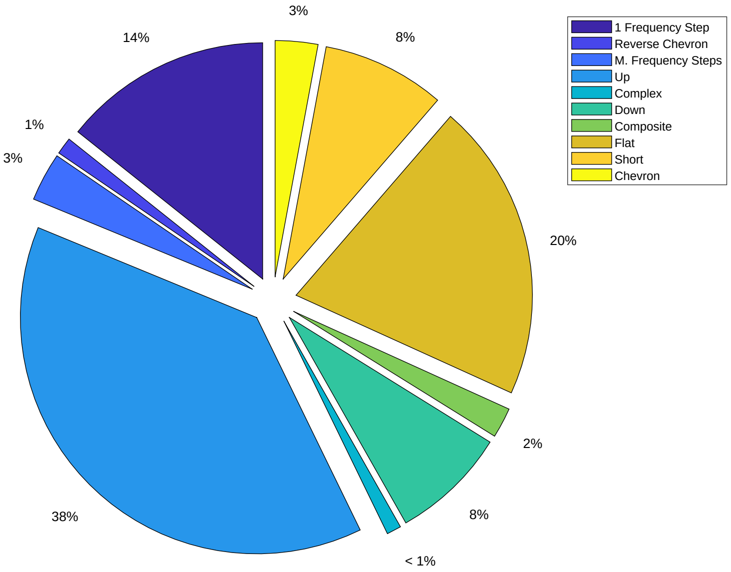


Figure 4.3: Dataset classes distribution.

Through the analysis of Figure 4.3, is evident that some classes have a small predominance in the whole dataset, resulting in an very unbalanced group of USVs. Furthermore, it is clear that the paradigm from which more vocalizations where annotated, was the Joy one, as presented Table 4.3. Note that this is also the paradigm with the higher number of annotated signals.

In order to understand how the types of vocalizations produced are related to the social paradigms and what paradigms lead the animals to vocalize more, some statistical tests were carried.

The first analysis performed is related to the rates of vocalization per minute of recording for each paradigm. This allows to understand which tests stimulate the most the animals and lead them to vocalize.

Table 4.4: Vocalizations rate by groups of animals and paradigm (**v/min**-vocalizations per minute).

Paradigm	**C57BL/6**			$Nf1^{+/-}$			$Nf1^{+/-}$ **Control**		
	Max (v/min)	**Min (v/min)**	**Mean (v/min)**	**Max (v/min)**	**Min (v/min)**	**Mean (v/min)**	**Max (v/min)**	**Min (v/min)**	**Mean (v/min)**
Anger	4.70	0.00	1.26	1.17	0.17	0.67	0.67	0.00	0.25
Anticipation	5.67	0.00	0.56	-	-	-	-	-	-
Background	2.57	0.86	1.32	1.29	0.00	0.31	1.29	0.00	0.27
Disappointment	2.00	0.00	0.52	4.07	0.00	0.80	0.86	0.00	0.18
Disgusting	5.37	0.00	1.27	4.07	0.00	0.44	4.00	0.00	0.53
Fear	6.55	0.00	2.42	1.71	0.00	0.69	2.36	0.00	0.56
Hunger	4.71	0.00	2.39	2.14	0.00	0.64	1.71	0.00	0.55
Joy	80.71	0.00	7.48	20.31	0.00	2.72	16.65	0.00	3.18
Surprise	4.67	0.00	1.49	10.93	0.00	1.62	-	-	-

Through the analysis of Table 4.4, some conclusions can be drawn. In general, C57BL/6 mouse tend to vocalize more, presenting higher vocalization rates for all paradigms in comparison to the remaining groups. Furthermore, the paradigm where all the animals groups vocalized the most was the Joy one. Other observation to be made is related to the fact that almost to every paradigm in each group of animals it existed signals where no vocalizations were produced. Note that in this analysis only the signals that were manually annotated were considered. In Table 4.4 the missing values means that no signals were annotated for that particular experiments.

After analyzing the paradigms which produce more vocalizations, statistical tests were conducted in order to understand if where there any relations between the used paradigms and the according classes considered, that is, investigate if some types of USVs are associated with some specific emotions. In order to study this relation, we used an N-Way ANOVA in which the production rate of each vocalization class from each of the signals classified is evaluated. This analysis was performed for each paradigm separately, and, inside each paradigm, individually for each of the animals genotypes considered in the study. Note that this analysis only contemplates the manually annotated signals.

The N-Way ANOVA test results are presented in Table 4.5. Besides the p-values obtained with the ANOVA test, in Table 4.5 the more common class by paradigm for each genotype is also presented. In these table, there are two columns with information about statistical differences, however they are not referent to the same thing. The first one is referring to the statistical difference between all the classes distributions whereas the second one is referring to the statistical difference between the most common class and the reaming ones, in order to understand if the more common class is really associated to a certain paradigm. The determination of the statistical differences between the most common classes is made through the analysis of figures 4.4, 4.5, 4.6, 4.7, 4.8, 4.9, 4.10, 4.11 and 4.12. In those figures the statistical differences obtained with the N-War ANOVA are presented and the vocalizations with higher production rate are highlighted in blue while other classes that have significantly different distributions, in the case that there are any, are marked in red.

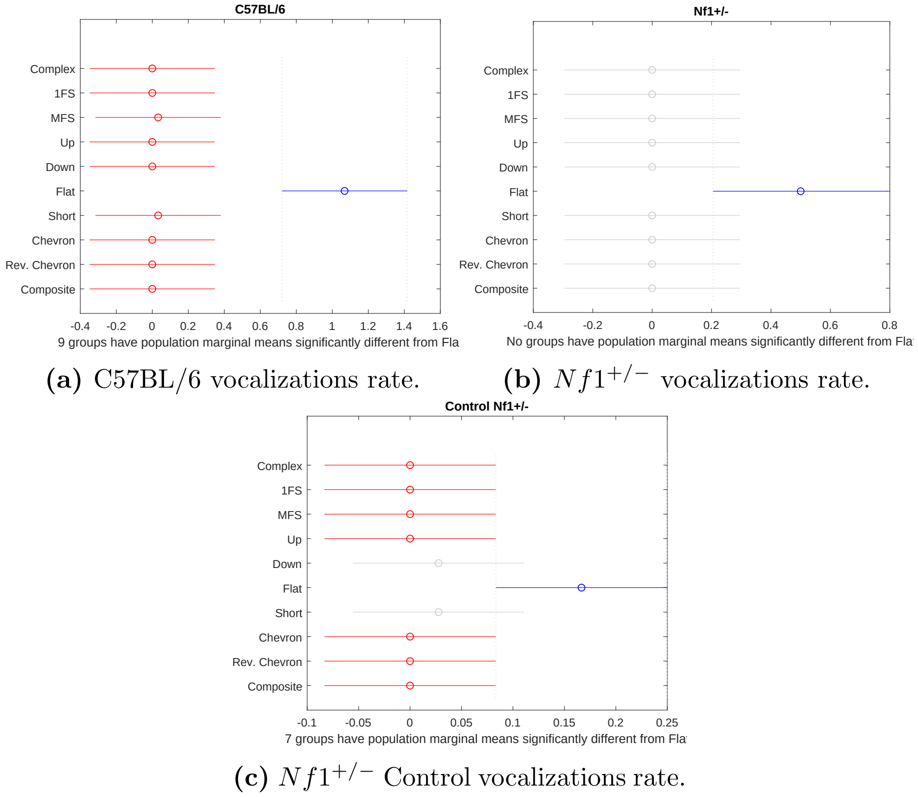
Through the analysis of the Table 4.5 and all of the figures presented bellow, some interesting conclusions emerge. One of the most prominent relations between the emotional state and the types of vocalizations is visible in the Fear paradigm, where the class Flat is clearly demarcated from all the others classes in all animal groups.

Aside from the Fear paradigm, no other presents the same class of vocalizations as being more common across all the genotypes. However, other interesting observation is related to the difference between the wild-type group and the $Nf1^{+/-}$ and according control group, being that the $Nf1^{+/-}$ and according control groups typically presents as more common vocalizations classes with more complex structures, such as the Multiple Frequency Steps.

As it happens with the previous analysis, some of the experiments with the different animal groups do not have annotated data, thus, in this cases, all the vocalizations distributions present the value zero.

Table 4.5: N-Way ANOVA results.

	C57BL/6				$Nf1^{+/-}$				$Nf1^{+/-}$ **Control**			
	p-Value	Statistical Difference All classes	Most common class	Statistical Diff. from other classes	p-Value	Statistical Difference All classes	Most common class	Statistical Diff. from other classes	p-Value	Statistical Difference All classes	Most common class	Statistical Diff. from other classes
Anger	3.58E-05	Yes	Flat	Yes	1.11E-01	No	Flat	Yes	4.30E-02	Yes	Flat	Yes
Anticipation	4.10E-02	Yes	Flat	No	-	-	-	-	-	-	-	-
Background	3.74E-02	Yes	Short	No	7.67E-02	No	MFS	No	7.85E-01	No	MFS	No
Disappointment	3.97E-01	No	Flat	No	1.44E-01	No	Flat	No	5.64E-01	No	MFS	No
Disgusting	1.09E-04	Yes	Down	Yes	5.61E-01	No	MFS	No	3.07E-01	No	Down	No
Fear	4.46E-06	Yes	Down	Yes	4.88E-01	No	1FS	No	2.59E-01	No	Up and Flat	No
Hunger	1.63E-07	Yes	Short	Yes	2.03E-01	No	Up	No	4.10E-02	Yes	Short	Yes
Joy	9.56E-02	No	1FS	No	3.58E-03	Yes	Up	Yes	6.38E-03	Yes	Up	Yes
Surprise	6.00E-02	No	Down	No	6.74E-01	No	Flat	No	-	-	-	-

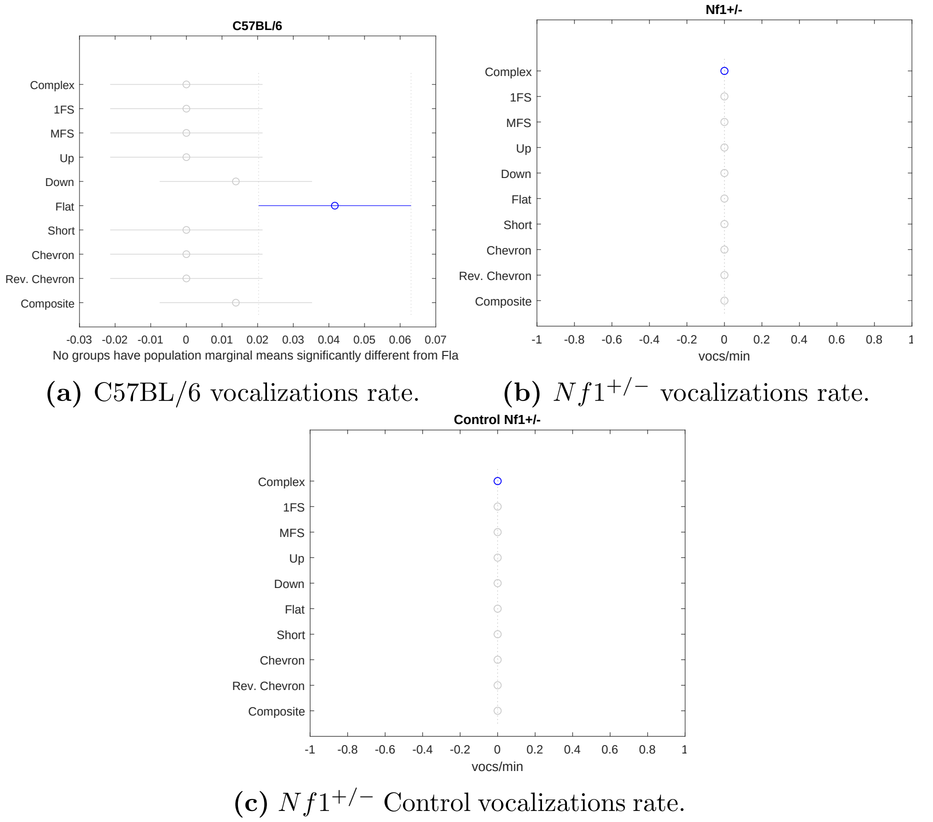


(a) C57BL/6 vocalizations rate.

(b) $Nf1^{+/-}$ vocalizations rate.

(c) $Nf1^{+/-}$ Control vocalizations rate.

Figure 4.4: Anger paradigm vocalizations rates distributions for each genotype.

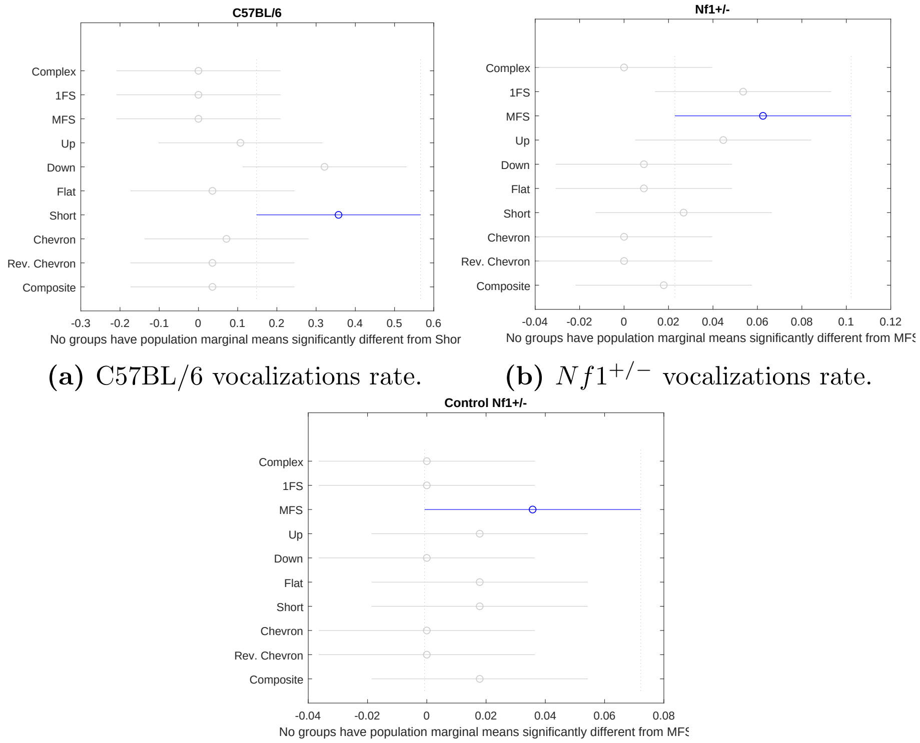


(a) C57BL/6 vocalizations rate.

(b) $Nf1^{+/-}$ vocalizations rate.

(c) $Nf1^{+/-}$ Control vocalizations rate.

Figure 4.5: Anticipation paradigm vocalizations rates distributions for each genotype.

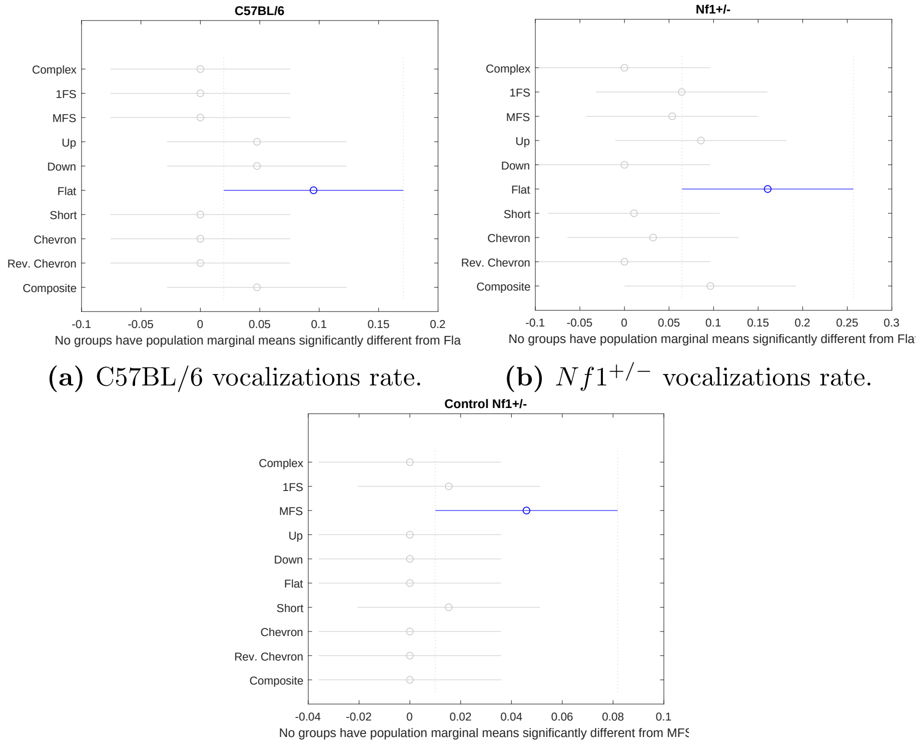


(a) C57BL/6 vocalizations rate.

(b) $Nf1^{+/-}$ vocalizations rate.

(c) $Nf1^{+/-}$ Control vocalizations rate.

Figure 4.6: Background paradigm vocalizations rates distributions for each genotype.

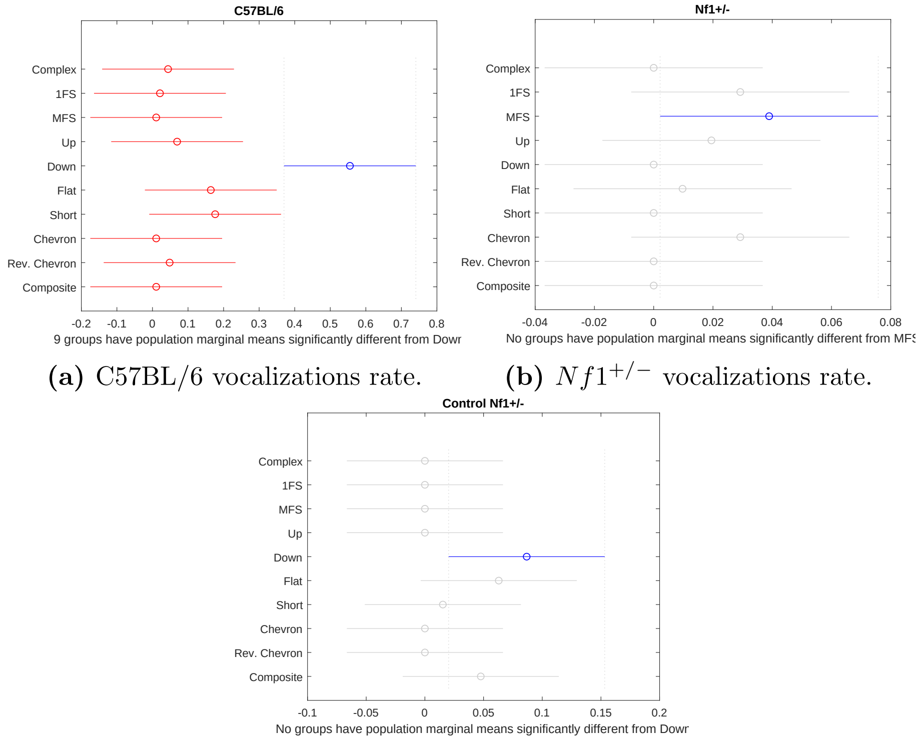


(a) C57BL/6 vocalizations rate.

(b) $Nf1^{+/-}$ vocalizations rate.

(c) $Nf1^{+/-}$ Control vocalizations rate.

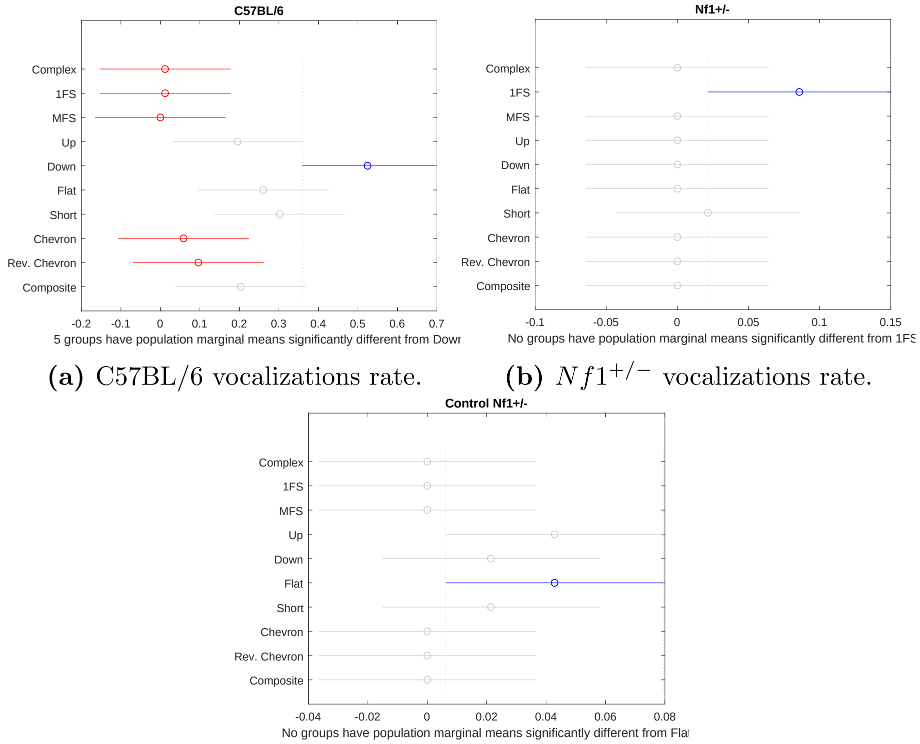
Figure 4.7: Disappointment paradigm vocalizations rates distributions for each genotype.



(a) C57BL/6 vocalizations rate. **(b)** $Nf1^{+/-}$ vocalizations rate.

(c) $Nf1^{+/-}$ Control vocalizations rate.

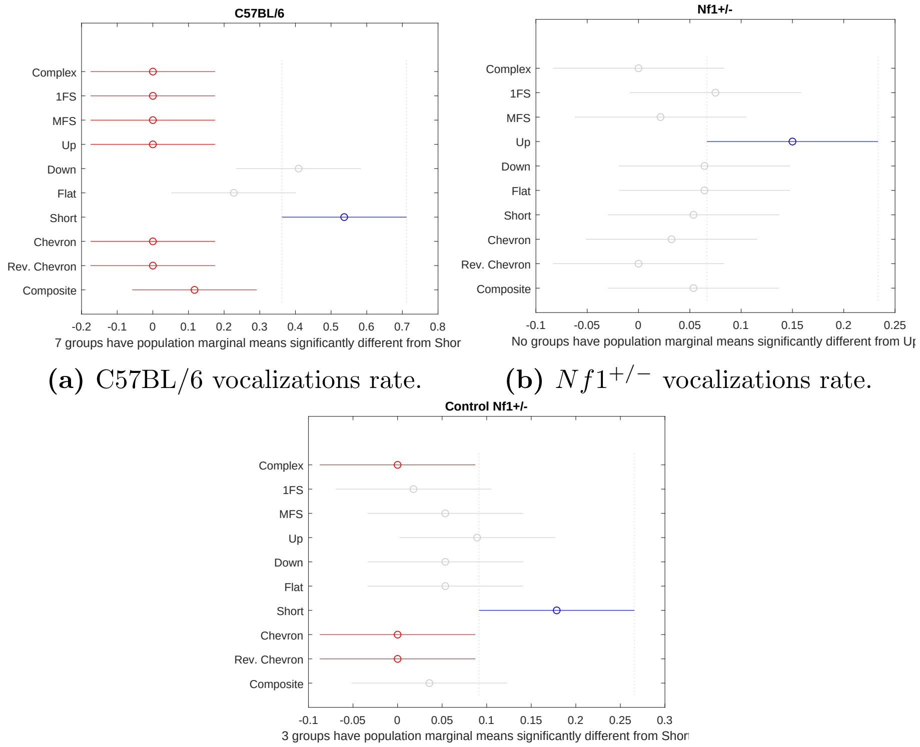
Figure 4.8: Disgusting paradigm vocalizations rates distributions for each genotype.



(a) C57BL/6 vocalizations rate. **(b)** $Nf1^{+/-}$ vocalizations rate.

(c) $Nf1^{+/-}$ Control vocalizations rate.

Figure 4.9: Fear paradigm vocalizations rates distributions for each genotype.

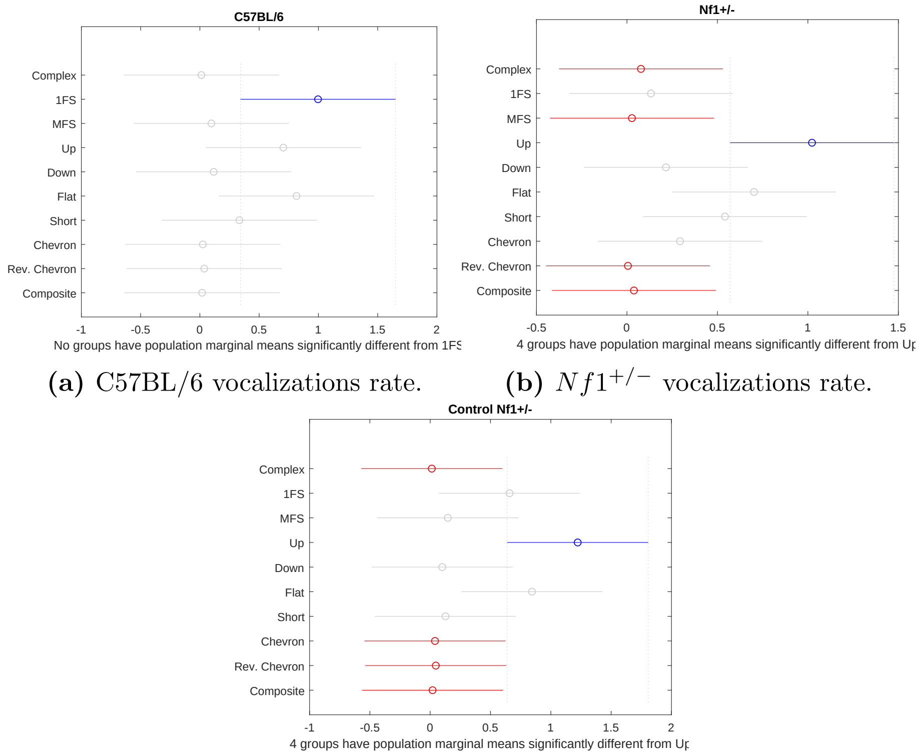


(a) C57BL/6 vocalizations rate.

(b) $Nf1^{+/-}$ vocalizations rate.

(c) $Nf1^{+/-}$ Control vocalizations rate.

Figure 4.10: Hunger paradigm vocalizations rates distributions for each genotype.

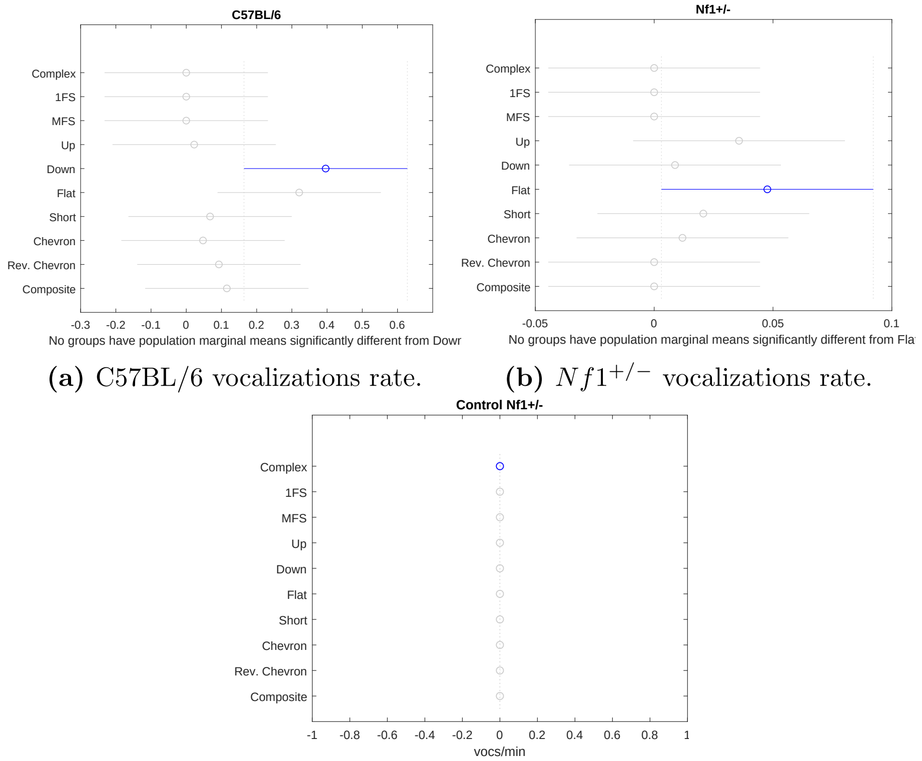


(a) C57BL/6 vocalizations rate.

(b) $Nf1^{+/-}$ vocalizations rate.

(c) $Nf1^{+/-}$ Control vocalizations rate.

Figure 4.11: Joy paradigm vocalizations rates distributions for each genotype.



(a) C57BL/6 vocalizations rate.

(b) $Nf1^{+/-}$ vocalizations rate.

(c) $Nf1^{+/-}$ Control vocalizations rate.

Figure 4.12: Surprise paradigm vocalizations rates distributions for each genotype.

Chapter 5

Methods

The bioacoustic studies which involve classification, usually, tend to follow a similar workflow to the one represented in Figure 5.1.

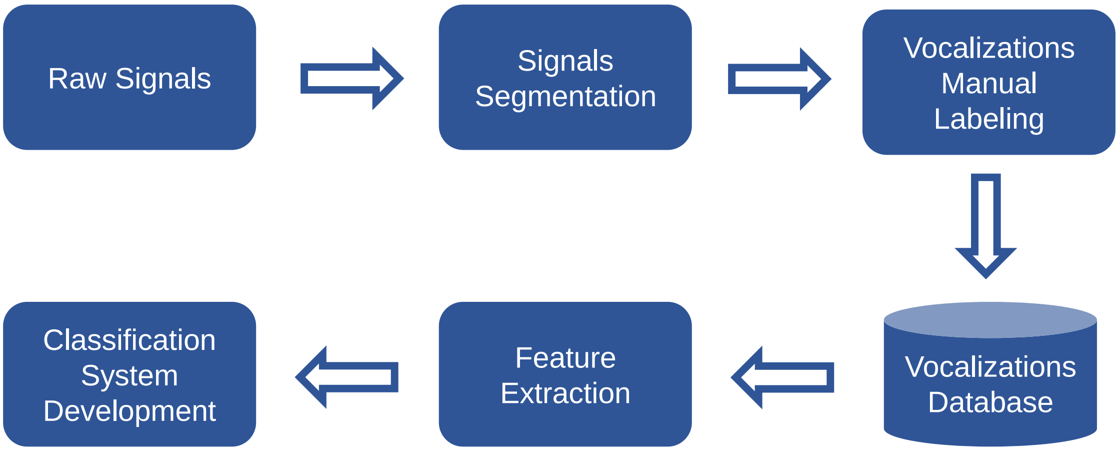


Figure 5.1: General workflow overview.

In this section, the methods used in the segmentation and classification phases are described.

5.1 Segmentation

This section describes the segmentation algorithm based on the spectral entropy. Furthermore, it also describes an image classification system, Bag of Visual Words (BOVW), which was used as a post-processing “filter” technique to categorize the segmentation’s algorithm output as real vocalizations or noise.

5.1.1 Spectral Entropy Segmentation Algorithm

Spectral entropy is a generalization of information entropy, which was a concept first proposed by Shannon in an attempt to measure information’s uncertainty [41]. This concept can be formulated by

$$H = -\sum_{m=1}^{N} P(m) \log_2 P(m) \tag{5.1}$$

The spectral entropy of a signal is a measure of its spectral power distribution and deals with the signal’s normalized power distribution in the frequency domain as a probability distribution $P(m)$[1]. Thus, the Spectral Entropy corresponds to the Shannon Entropy of that distribution and has been proposed to measure the distribution of frequencies of a signal $x(n)$ at a given instant t, defined as:

$$H(t) = -\sum_{m=1}^{N} P(t,m) \log_2 P(t,m) \tag{5.2}$$

$$P(t,m) = \frac{S(t,m)}{\sum_f S(t,f)} \tag{5.3}$$

where N is the number of signal samples, $S(t,m)$ is the power spectrum and $S(t,f)$ is the time-frequency spectrogram.

The proposed segmentation algorithm was developed with MATLAB R2018b using an Intel®i7-4700HQ processor and 8GB of RAM. The segmentation algorithm is essentially composed by two parts: pre-processing and segmentation.

First the signal is filtered with a high-pass Butterworth filter of order 20 with a cut-off frequency of 30 kHz. The high-pass filter aims essentially to reduce the interference of low frequency noise, which can be generated by the mouse walking or touching the box, among other external factors.

After the pre-processing phase, the signal is divided into multiple non overlapping windows of 1 s and, for each window, the spectral entropy is computed on intervals of 1.9 ms (temporal resolution). Then, the values where spectral entropy vector are below a certain threshold (threshold 1 Figure 5.2) are stored as the initial and final points of a vocalization. In addition, the value of the spectral entropy must also be smaller than a second threshold (threshold 2 Figure 5.2) at some point between the stored instants,

[1] https://www.mathworks.com/help/signal/ref/pentropy.html

in order reduce the false positive rate of the algorithm. The way these thresholds were estimated is described in subsection 6.1.1. Besides that, vocalizations that followed each other at intervals <15 milliseconds were subsequently merged and considered as a single vocalization [34]. This last step was considered, since in some vocalizations there was a noticeable interruption of the vocalization or loss of power which make the algorithm segment two or more different vocalizations instead of just one.

The output produced by the algorithm is represented in Figure 5.2, where the first threshold corresponds to the green line and the second to the red line.

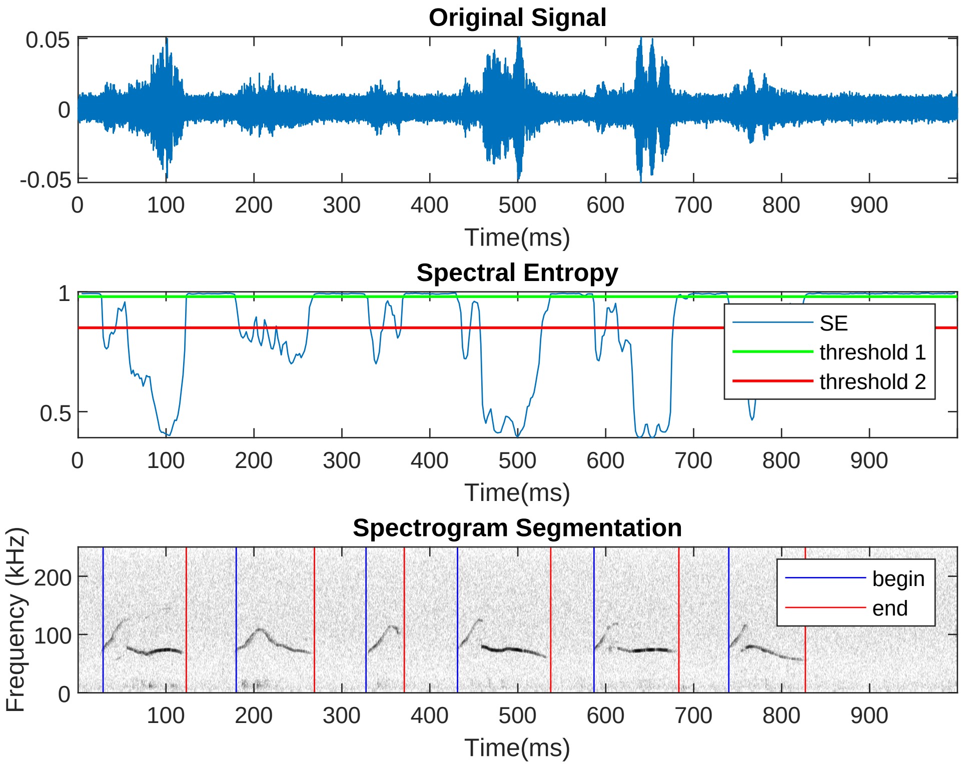


Figure 5.2: Example of a Signal Segmentation.

Through the analysis of the segmentation results obtained with the multiple thresholds (see Figure 6.1), it is possible to conclude that the higher the second threshold is, the higher the recall value becomes, which means that algorithm can effectively recognize a higher number of vocalizations. The second threshold is used to minimize the rate of false positive detections of the algorithm, however, it has a negative impact on the true positive rate, which is far from ideal, since in these kind of studies the main focus is trying to recognize each and every sound produced by the animals.

5.1.2 Bag of Visual Words

To tackle the false positives detection problem, another approach was taken, aside from the use of the second threshold. After the segmentation process, using only threshold 1 (green line in Figure 5.2), all the syllables detected were then passed through an image classification system in order to be classified as vocalizations or noise. The used classifier was a BOVW.

The concept of image classification with BOVWs is an idea adapted from a methodology originally developed in text classification and Natural Language Processing (NLP) [42]. In text classification applications, the Bag of Words (BOW) works by counting the number of times each word appears in a document, and then, using the frequency of each word in the document, a frequency histogram of all words is created to represent the document. The concept is similar in images classification, differing only in the information used to represent the objects to classify. While in text classification the objects are represented by frequency histograms of the words, in image classification the objects are defined by frequency histograms of visual words, which are descriptors used to characterize the images [43].

The first step in BOVW classification is the extraction of the features or key-points from the images in order to build the features histograms (see Figure 5.3). Thus, this process main propose is to recognize the key points of each image and their frequency. In BOVW these points are represented by a set of descriptors. Several sets of features can be used to create the bag of features, such as Speeded-Up Robust Features (SURF), Scale-Invariant Feature Transform (SIFT), among others. However, KAZE features (see subsubsection 5.2.3.5) were chosen, because there are the most adequate to detect the key points in the Ultrasonic Vocalizations (USVs) time-frequency maps.

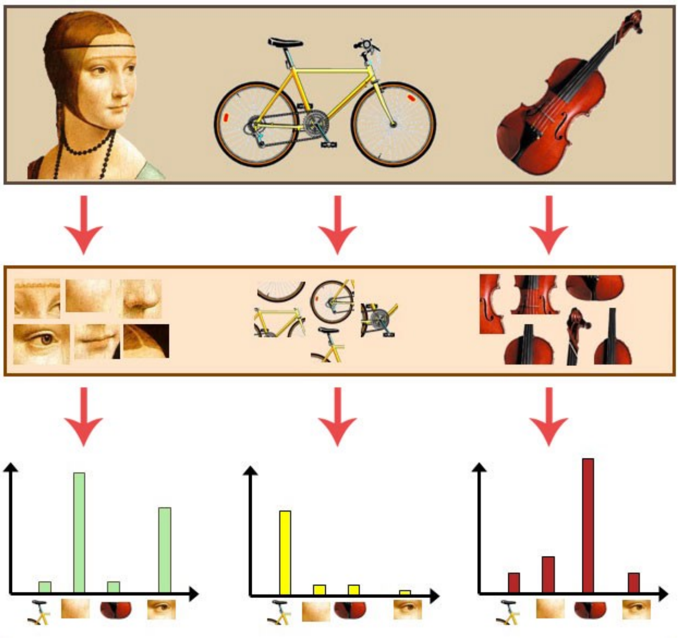


Figure 5.3: Example of features histogram creation[2].

After the descriptors have been extracted from the multiple images, the key points from all the training images are grouped with clustering techniques, such as k-Means, being that the center of each cluster corresponds to a different visual word [43] (see Figure 5.4). Thus, the center of each cluster will be used as the visual vocabulary dictionary.

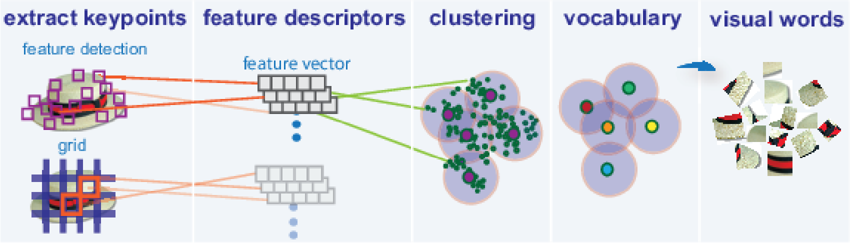


Figure 5.4: Visual Words Creation[3].

The BOVWs are then trained using simple classifiers, such as Support Vector Machines (SVMs). To train the classifiers, this method uses the bag of visual words object to encode images in the image set into the histogram of visual words. After, the histograms of visual words are used as the positive and negative samples to train the classifier. Given a new image, the system extracts the key points, and using the trained bag of features encodes the new images, converting it to a features histogram, and, lastly, to a feature vector (see Figure 5.5).

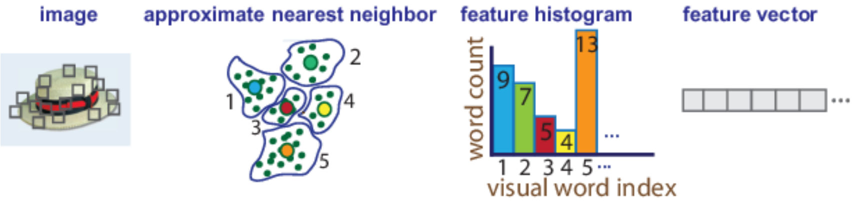


Figure 5.5: New image feature vector creation[4].

5.2 Feature Extraction

The feature extraction process translates, in some way, the passage from raw data to machine algorithms interpretative data, which means the representation of raw data by a set of descriptors or features. This process can be divided in multiple steps, starting by the brainstorming about what features may or may not be useful to the problem in question. During this initial phase, some concerns must be addressed. First, the adequation of the features and their discriminate power must be considered, being that the higher is their discriminate power, the easier is to unequivocally represent the data. After a certain set of features has been selected, the features must then be extracted and tested with the machine learning algorithms, in order to access their quality. If the performances obtained

[2] https://towardsdatascience.com/bag-of-visual-words-in-a-nutshell-9ceea97ce0fb
[3] https://www.mathworks.com/help/vision/ug/image-classification-with-bag-of-visual-words.html
[4] https://www.mathworks.com/help/vision/ug/image-classification-with-bag-of-visual-words.html

are not good enough, the process must be repeated iteratively until satisfactory results are achieved.

Nowadays, the process of feature engineering can somehow be avoided in newer classification methodologies, such as deep learning and artificial neural networks. In these methodologies, raw data are usually directly fed to the classifiers. Then, the networks try to extract patterns from data. However, using this approach may took away the interpretability of how the classification works and how are the data being represented.

Summing up, the feature engineering process is one of the core and most, if not the most, important elements regarding machine learning algorithms. If the data is well described and represented, it will be possible to obtain good performance even with the simplest Machine Learning (ML) algorithm.

Usually, in bioacoustic studies, one of the most frequent ways to extract information from a certain sound produced by an animal is through the extraction of characteristics from the time-frequency ridge, or contour, obtained from the time-frequency representations of the sounds. Furthermore, spectral features are also common.

In order to describe the USVs in a most complete way, several types of features were extracted. Three main types of features were considered from: the time, frequency and time-frequency domains.

5.2.1 Time Domain Features

5.2.1.1 Time Amplitude Related Features

The amplitude features are related to the raw signal in the time domain. Using the signal in this domain, six descriptors were extracted from the signal amplitude, namely:

1. The mean;
2. The median;
3. The standard deviation;
4. The skewness;
5. The kurtosis;
6. The root mean square.

5.2.1.2 Zero Crossing Rate

The zero-crossing rate translates the rate at which the signal changes from positive to zero to negative or from negative to zero to positive in a given window, thus, the higher the crossing rate is the the more high-frequency content can be assumed to be in the signal [44].

$$ZCR(i) = \frac{1}{2N} \sum_{n=0}^{N-1} |\text{sign}[x_i(n)] - \text{sign}[x_i(n-1)]| \tag{5.4}$$

where

$$sign[x_i(n)] = \begin{cases} 1, x_i(n) \geq 0 \\ -1, x_i(n) < 0 \end{cases} \tag{5.5}$$

with N corresponding to the total number of frames used.

5.2.1.3 Short Time Energy

The Short Time Energy (STE) is the energy of short audio segment and is typically used to detect voice or unvoiced segments [45]. The STE is given by the following expression:

$$E_n = \sum_{m=n-N+1}^{n} [x(m)w(n-m)]^2 \tag{5.6}$$

where $x(m)$ is the signal in the temporal domain and w is the window used to determine the energy.

5.2.1.4 Tonal Power Ratio

The tonal power expresses the relation between Tonal Energy (E_T) and the total Power of the signal [44].

$$T_{pr} = \frac{E_T(n)}{\sum_{i=0}^{k/2-1} |X(k,n)|^2} \tag{5.7}$$

The tonal power ratio is a value between 0 and 1, and low results hint toward a flat (noisy) spectrum while high results indicate a tonal spectrum.

5.2.2 Frequency Domain Features

5.2.2.1 Tonality

The tonality of each vocalization corresponds to the subtraction from 1 of the geometric mean of the power spectrum divided by its arithmetic mean [20].

$$Tonality = 1 - \frac{e^{\frac{1}{N}\sum_{n=0}^{N-1} ln(X(n))}}{\frac{1}{N}\sum_{n=0}^{N-1} X(n)} \tag{5.8}$$

where $X(n)$ is the power spectrum.

5.2.2.2 Signal Power

The Signal Power corresponds to the total power in decibels (dB) and is given by the sum of Fourier Transform Spectrum.

5.2.2.3 Spectral Entropy

The Spectral Entropy concept is described in subsection 5.1.1. This feature gives information regarding the continuity of the time frequency spectrum and its distribution. After estimating the spectral entropy distribution for each USV, the mean, median, maximum, minimum and standard deviation of the distribution as well as non-instantaneous spectral entropy values were determined.

5.2.2.4 Spectral Centroid

The Spectral Centroid is a measure used in digital signal processing to characterize the spectrum. The centroid indicates where the center of gravity (COG) or "center of mass" of the spectrum is located, and corresponds to the frequency-weighted sum of the power spectrum normalized by its unweighted sum [44].

$$SC = \frac{\sum_{k=b_1}^{b_2} f_k s_k}{\sum_{k=b_1}^{b_2} s_k} \tag{5.9}$$

where f_k is the frequency in Hz corresponding to bin k; s_k is the spectral value at bin k; b_1 and b_2 are the band edges.

5.2.2.5 Spectral Spread

The Spectral Spread, sometimes also referred to as instantaneous bandwidth, describes the concentration of the power spectrum around the spectral centroid which can bee seen as a measure of the average spread of the spectrum in relation to its centroid. It can be interpreted as the standard deviation of the power spectrum around the spectral centroid [44] .

$$SS = \sqrt{\frac{\sum_{k=b_1}^{b_2} (f_k - SC)^2 s_k}{\sum_{k=b_1}^{b_2} s_k}} \tag{5.10}$$

where f_k is the frequency in Hz corresponding to bin k; s_k is the spectral value at bin k; b_1 and b_2 are the band edges; SC is the spectral centroid.

5.2.2.6 Spectral Slope

The Spectral Slope is a measure of the slope of the spectral shape and it is calculated using a linear regression of the magnitude spectrum [44].

$$SSl = \frac{\sum_{k=b_1}^{b_2} (f_k - \mu_f)(s_k - \mu_s)}{\sum_{k=b_1}^{b_2} (f_k - \mu_f)^2} \tag{5.11}$$

where f_k is the frequency in Hz corresponding to bin k; μ_f is the mean frequency; s_k is the spectral value at bin k; b_1 and b_2 are the band edges; μ_s is the mean spectral value.

5.2.2.7 Spectral Rolloff

The Spectral Rolloff is defined as the frequency bin below which the accumulated magnitudes of the $STFT$, $X(k, n)$, reach a certain percentage of the overall sum of magnitudes [44]. The $RolloffPoint = i$ is defined in such a way that

$$\sum_{k=b_1}^{i} |s_k| = k \sum_{k=b_1}^{b_2} s_k \tag{5.12}$$

where s_k is the spectral value at bin k; b_1 and b_2 are the band edges; k is a specified energy threshold.

The result of the Spectral Rolloff is a bin index and it can be converted either to Hz or to a parameter ranging between zero and one. Low results indicate insignificant magnitude components at high frequencies and a low audio bandwidth [44], accordingly.

5.2.2.8 Spectral Flux

The Spectral Flux is a measure of how quickly the power spectrum of a signal is changing, therefore it is defined as the average difference between consecutive $STFT$ frames [44].

$$SF(t) = \left(\sum_{k=b_1}^{b2} |s_k(t) - s_k(t-1)|^p \right)^{1/p} \tag{5.13}$$

where s_k is the spectral value at bin k; b_1 and b_2 are the band edges; p is the norm type.

The Spectral Flux can be interpreted as an approximation to the roughness and varies between 0 and A, representing A the maximum possible spectral magnitude [44].

5.2.2.9 Spectral Decrease

The spectral decrease estimates the steepness of the decrease of the spectral envelope over frequency, which essentially represents the amount of decrease of the spectrum [44].

$$SD = \frac{\sum_{k=b_1+1}^{b_2} \frac{s_k - s_{b_1}}{k-1}}{\sum_{k=b_1+1}^{b_2} s_k} \tag{5.14}$$

where s_k is the spectral value at bin k; b_1 and b_2 are the band edges.

5.2.2.10 Spectral Crest Factor

The Spectral Crest is a simple measure of tonalness which relates the ratio of the maximum of the spectrum to the arithmetic mean of the spectrum [44].

$$SCr = \frac{\max(s_{k \epsilon [b_1, b_2]})}{\frac{1}{b_2 - b_1} \sum_{k=b_1}^{b_2} s_k} \tag{5.15}$$

where s_k is the spectral value at bin k; b_1 and b_2 are the band edges.

5.2.2.11 Spectral Flatness

The spectral flatness is the ratio of the geometric mean and arithmetic mean of the magnitude spectrum. The metric is an indication of the peakiness of the spectrum [44]. A higher spectral flatness indicates noise, while a lower spectral flatness indicates tonality.

$$SF = \frac{\left(\prod_{k=b_1}^{b_2} s_k\right)^{\frac{1}{b_2-b_1}}}{\frac{1}{b_2-b_1}\sum_{k=b_1}^{b_2} s_k} \tag{5.16}$$

where s_k is the spectral value at bin k; b_1 and b_2 are the band edges.

5.2.2.12 Spectral Skewness

The spectral skewness measures symmetry around the spectral centroid, thus it is a measure of the symmetry of the distribution of the spectral magnitude values around their arithmetic mean [44]. In phonetics, spectral skewness is often referred to as spectral tilt and is used with other spectral moments to distinguish the place of articulation[5].

$$SSk = \frac{\sum_{k=b_1}^{b_2}(f_k - SC)^3 s_k}{SS^3 \sum_{k=b_1}^{b_2} s_k} \tag{5.17}$$

where f_k is the freuqency in Hz corresponding to bin k; s_k is the spectral value at bin k; b_1 and b_2 are the band edges; SC is the spectral centroid; SS is the spectral spread.

5.2.2.13 Spectral Kurtosis

The spectral kurtosis measures whether the distribution of the spectral magnitude values is shaped like a Gaussian distribution or not [44], and measures the flatness, or non-Gaussianity, of the spectrum around its centroid. Conventionally, it is used to indicate the peakiness of a spectrum.

$$SK = \frac{\sum_{k=b_1}^{b_2}(f_k - SC)^4 s_k}{SS^4 \sum_{k=b_1}^{b_2} s_k} \tag{5.18}$$

where f_k is the frequency in Hz corresponding to bin k; s_k is the spectral value at bin k; b_1 and b_2 are the band edges; SC is the spectral centroid; SS is the spectral spread.

5.2.2.14 Spectral Pitch Chroma

The Pitch Chroma is a histogram-like 12-dimensional vector with each dimension representing one pitch class. To extract the pitch Chroma the frequency representation of the audio signal block is grouped into semi-tone bands, then a measure of salience is computed in each band, and lastly, the sum of all bands (over all octaves) corresponding to a specific pitch class is calculated [4].

[5] https://www.mathworks.com/help/audio/ug/spectral-descriptors.html

5.2.2.15 Spectral Edge Frequency

The Spectral Edge Frequency corresponds to the frequency bellow which a certain percentage of the area under the power spectrum density is concentrated. Therefore, it allows to estimate where the stronger frequencies of a vocalization are concentrated. Typical values used for the percentage value are 70%, 90%, 95%, having the used value been 70%.

5.2.2.16 Harmonic components

The harmonic components feature intends to translate the presence, or absence, of harmonic components in the vocalizations. An harmonic component is a component whose frequency is multiple of the main component of the vocalization. Thus, this is a binary descriptor, in which the value 1 translates the presence of an harmonic component, and the value 0, the absence.

This estimation is made through the Power Spectral Density (PSD) distribution, where the frequency with maximum power is estimated. Afterwards, other frequencies with spectral power above a certain percentage of the maximum spectral power are also estimated. If one or more of these frequencies are multiple of the frequency with the highest power, they are considered to be harmonic components.

5.2.2.17 Power Spectrum Related Features

Through the USVs PSD distribution, several features were also extracted. The extracted features from the PSD distribution were the following ones:

1. **Maximum Value:** Which corresponds to the maximum power value observed in the PSD;
2. **Most Powerful Frequency:** Which corresponds to the frequency with the highest power value of the PSD;
3. **Minimum Value:** Which corresponds to the minimum power value observed in the PSD;
4. **Least Powerful Frequency:** Which corresponds to the frequency with the lowest power value of the PSD;
5. **Mean Power:** Which corresponds to the mean power value of the PSD;
6. **Median Power:** Which corresponds to the median power value of the PSD;
7. **Power Standard Deviation:** Which corresponds to the standard deviation of the power values of the PSD;
8. **Mean Frequency:** Which corresponds to the mean normalized frequency of the PSD of a time-domain signal.

5.2.3 Time-Frequency Domain Features

USVs are non-stationary signals, and thus methods based on time-frequency maps are adequate to describe them.

In order to extract almost all the features presented in this section the time-frequency representations of the vocalizations were determined using the Short-Time Fourier Transform (STFT).

The STFT is a Fourier-related transform used to determine the amplitude of the oscillatory components as well as their phase at local sections of a signal as it changes over time, and is defined as:

$$STFT(\tau,\omega) \equiv X(\tau,\omega) = \int_{-\infty}^{+\infty} x(t)w(t-\tau)e^{-jwt}dt \tag{5.19}$$

where $x(t)$ is the signal in the time domain, w is the windowed function and τ is the window time location and determines the epoch under analysis.

In all STFT based methods presented, the spectrograms are generated using a Hamming window with 256 points, 50% of overlap and 1024 sampling points to calculate the discrete Fourier transform.

5.2.3.1 Contour Extraction

In bioacoustic studies and signals processing applications, it is usual to extract oscillatory components and their properties from time–frequency representations of the sounds, where multiple techniques to obtain the time–frequency contour can be used [46–50]. This procedure is based on the search of the appropriate ridge curve, which consist in a sequence of amplitude peak positions (ridge points), corresponding to the component of interest and providing a measure of its instantaneous frequency alongside the time axis [51]. This extraction presents a serious challenge, not only due to the amount of noise that the audio signals contain, but also due to the fact that some vocalizations present some irregularities in their time-frequency spectrum and lots of discontinuity points.

In this project, several techniques were developed and tested in order to extract the contour of the vocalizations. These techniques can be divided in three main groups (see Table 5.1), according to the approach that was used to extract the time-frequency ridge. Furthermore, two different techniques were also used to determine the time-frequency representations, namely the STFT and synchrosqueezing methods.

The synchrosqueezed time-frequency representation methods are reassignment methods. The reassignment process can be seen, conceptually, as a two-step process, being the first step a smoothing step, whose main purpose is to eliminate oscillatory interference, and the second stage a squeezing step, whose intention is to refocus the contributions that survived the smoothing process [52]. Thus, synchrosqueezing is essentially a post-processing method whose main objective is to sharpen the timescale representation [53], and it can be used with multiple time-frequency estimation techniques, such as the STFT or wavelets, which were the two methods used.

The Fourier Synchrosqueezed Transform (FSST) is based on the STFT, typically generating sharper time-frequency estimations than the STFT. The FSST determines the STFT of a signal f using a spectral window g, being defined as

$$V_g(t,\eta) = \int_{-\infty}^{+\infty} f(x)g(x-t)e^{-j2\pi\eta(x-t)}dt \tag{5.20}$$

The Wavelet Synchrosqueezed Transform (WSST) is also a reallocation method which aims to “sharpen” the time-frequency representation. The WSST works by firstly determining the Continuous Wavelet Transform (CWT) of the input time domain signal. Then, the instantaneous frequencies from the CWT are extracted using a phase transform. Lastly, the CWT is “squeezed” over regions where the phase transform is constant. The resulting instantaneous frequency value is reassigned to a single value at the centroid of the CWT time-frequency region. This reallocation process results in a sharper time-frequency estimation.

In total, five different methods were developed to extract the USVs contour, which are represented in Table 5.1, grouped accordingly to the approach taken.

Table 5.1: Contour extraction methods summary table.

Method		**TF estimation method**
Maximum Time-Frequency Ridge Extraction	Default	STFT, FSST, WSST
	Interpolation	STFT
	Tonality	STFT
Best Path		STFT
Image Based		STFT

Maximum Time-Frequency Ridge Extraction

The Maximum Time-Frequency Ridge Extraction is a group of three different methods developed to extract the USVs contour. All methods are based on the determination of the maximum time-frequency ridge, as the name suggest. However, in two of them, some of the points obtained from the maximum time-frequency ridge are excluded based on specific criteria.

The firs step in all methods based on the Maximum Time-Frequency Ridge Extraction is the determination of the time-frequency map of the vocalization, using the STFT or the syncrosqueezing methods (FSST and WSST). After that, all algorithms iterate along each temporal instant of the spectrogram and find the point of maximum power, which are then stored in a vector containing the value of the maximum frequency per epoch. In the example represented in Figure 5.6, the contour vector contains the frequency values corresponding to every yellow square. This initial process is similar to all the five methods.

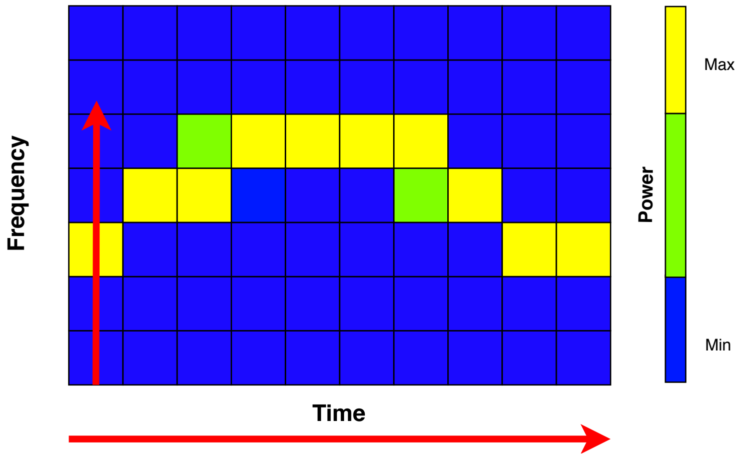


Figure 5.6: Example of time-frequency ridge extraction.

In the second stage of the methods is where they differ. In the default Maximum Time-Frequency Ridge Extraction Default methods all the values obtained from the time-frequency ridge are kept, regardless of the time-frequency estimation method used. On the other hand, for the two other methods (Maximum Interpolation Time-Frequency Ridge Extraction and Maximum Interpolation Time-Frequency Ridge Extraction), some points are removed.

In the Maximum Interpolation Time-Frequency Ridge Extraction method, using an outlier identification method based on the median value of the frequencies vector, the outlier points are removed (see Figure 5.7). After the removal process, the “gaps” in the frequencies vector are replaced with points estimated with a linear interpolation method.

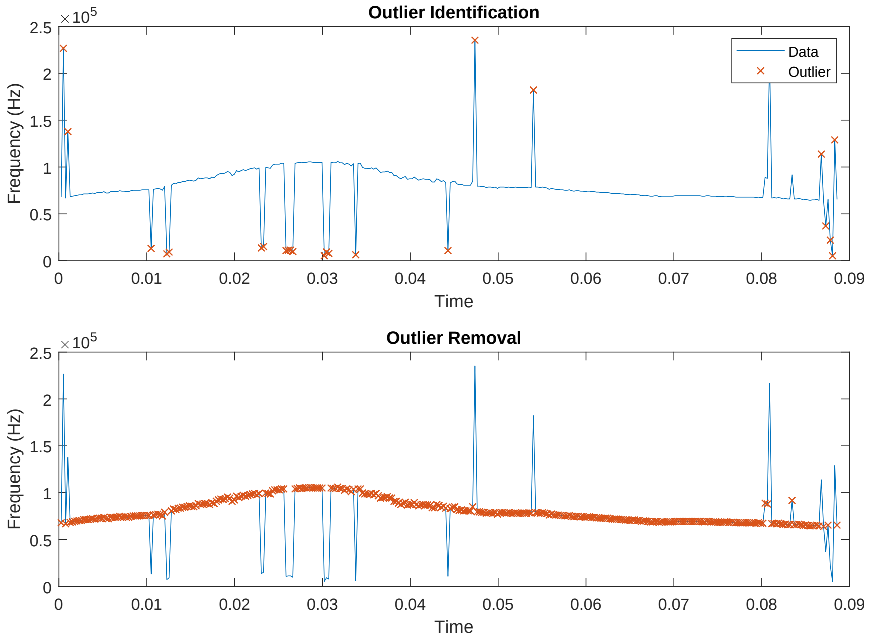


Figure 5.7: Maximum Interpolation Time-Frequency Ridge Extraction outlier removal.

In the Maximum Tonality Time-Frequency Ridge Extraction method, the tonality vector for each vocalization is determined. Then, the points in which the tonality value is bellow a certain threshold are excluded, and, in a similar away, the "gaps" produced by this process (instants corresponding to the red dots in Figure 5.8) are replaced with points estimated with a linear interpolation method.

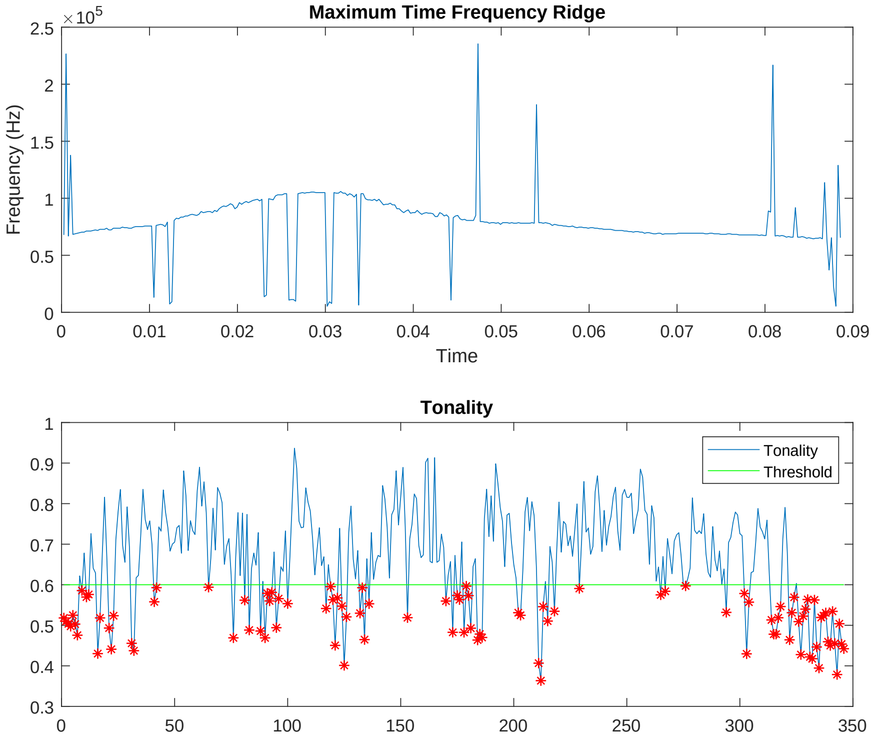
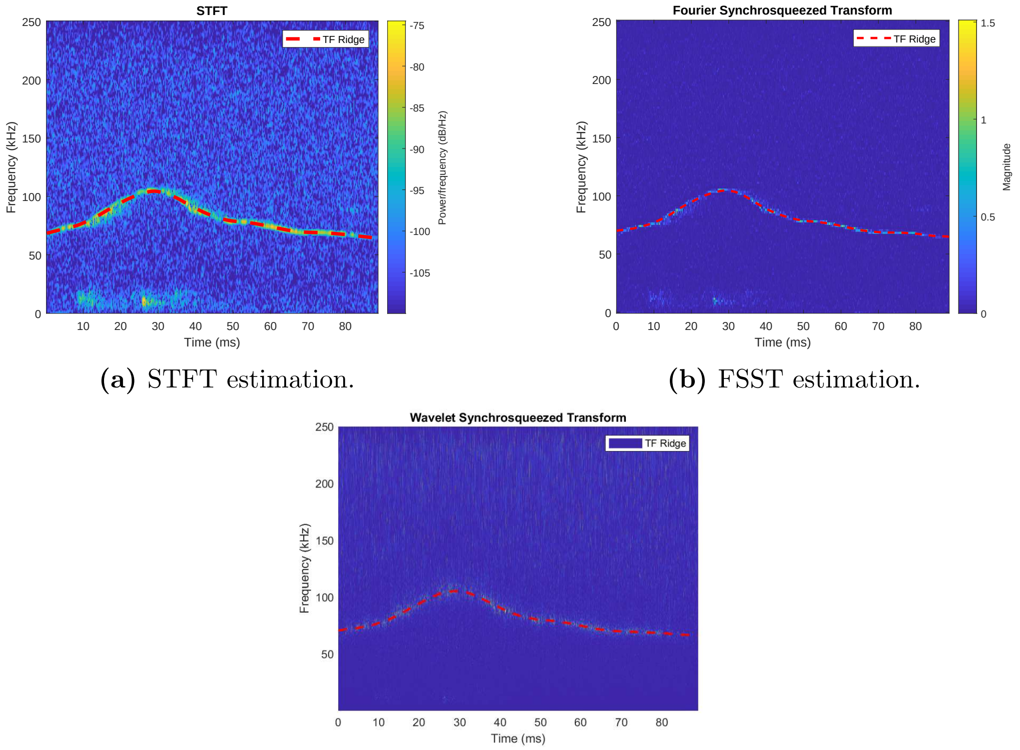


Figure 5.8: Maximum Tonality Time-Frequency Ridge Extraction outlier removal.

Lastly, after the determination of the vectors containing the frequency values determined with all five methods, a smoothing operation using a linear regression method ('rlowess') was implemented on the methods which used the STFT, and a moving average filter with an order equal to 10% of the size of all the frequency vector size on the methods that used the FSST and WSST. Note that a different method for smoothing the final contour vector was used depending on the method used to estimate the time-frequency representation due to computational constraints. Since the time-frequency maps generated with syncrosqueezed method have a much bigger number of data points than the ones obtained with STFT, the regression method ('rlowess') would be to expensive in computational terms.

The output produced by the Maximum Time-Frequency Ridge Extraction methods is presented in Figure 5.9.



(a) STFT estimation.

(b) FSST estimation.

(c) WSST estimation.

Figure 5.9: Maximum Time-Frequency Ridge Extraction outputs.

Best Path Time-Frequency Ridge Extraction

The Best Path Time-Frequency Ridge Extraction takes a different approach from the previous described methods. First, the STFT is used to estimate the time-frequency representation of the vocalizations. Then, the point of maximum spectral power is determined (red square in Figure 5.10). Starting form this reference point, the spectrogram is traced back and forward (see Figure 5.10). For each temporal instant, the N points of maximum spectral power are determined, being that the frequency value that is chosen corresponds to the one which is closer to the previous frequency value, his last neighbor. Lastly, the forward and backward pieces of the contour are joined backed together, creating the complete vector of frequencies, which is then smoothed with a linear regression method ('rlowess').

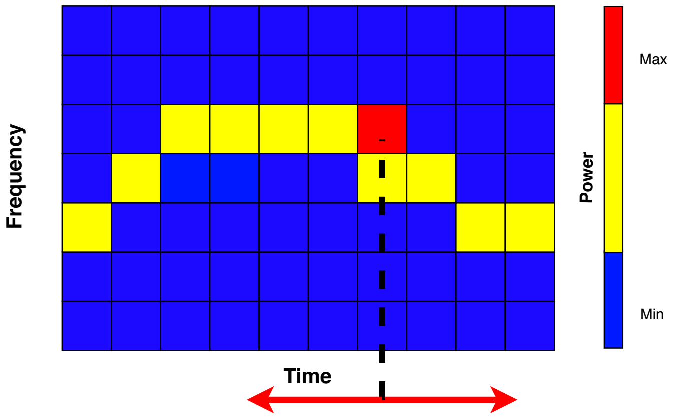


Figure 5.10: Best Path Time-Frequency Ridge extraction.

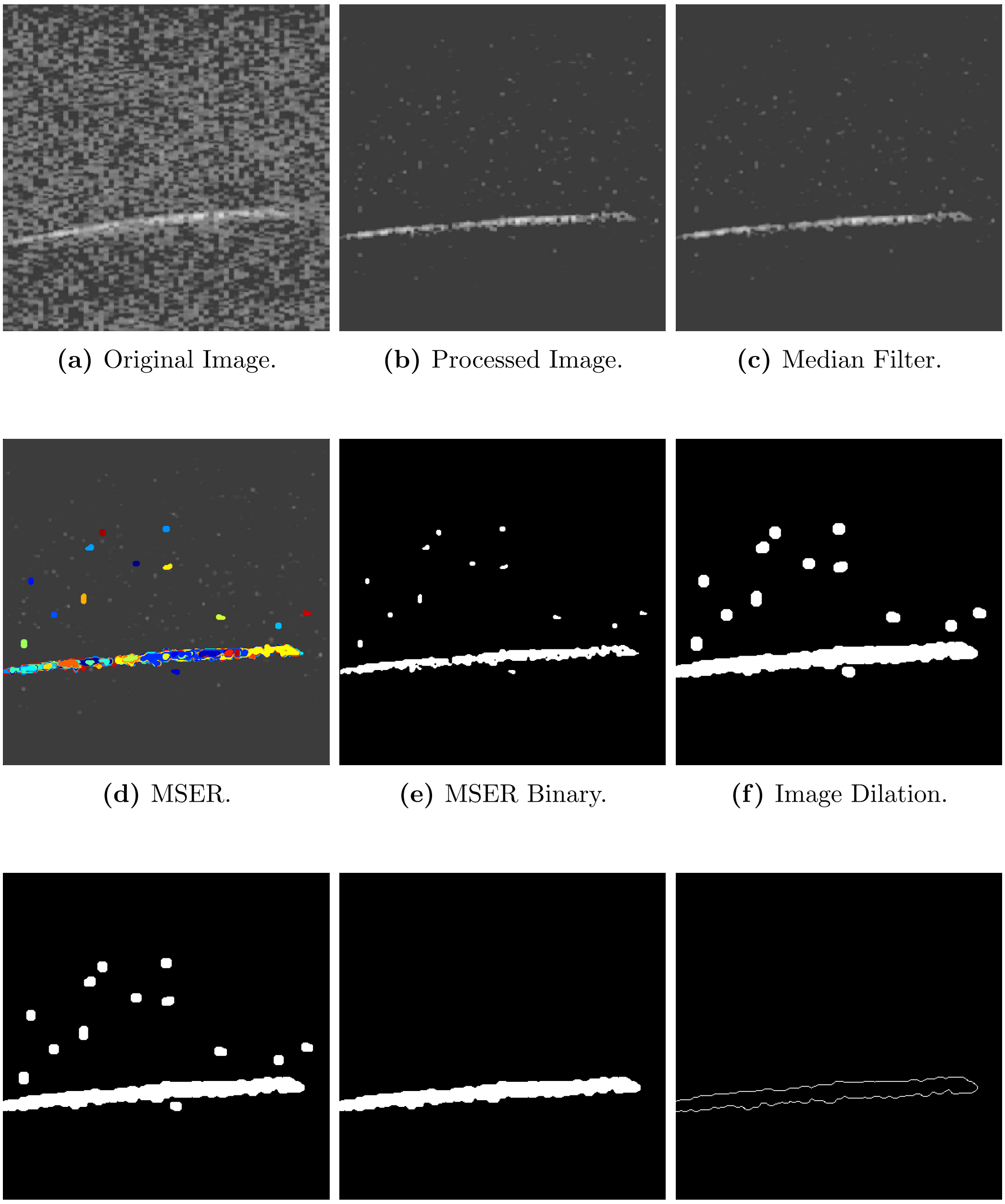
Image Based Time-Frequency Ridge Extraction

This method is based on image techniques to extract the contour from USVs STFT images. First, the STFT of each vocalization is converted to a gray image with 250-by-250 pixels (see Figure 5.10a). The image downsizing will not only decrease the computational cost, but will also made the conversion from pixels to frequency values easier. Since the signals sampled at 500 kHz, and, consequently, the spectrograms are represented from 0 to 250 kHz, using a 250-by-250 image, will allow direct conversion from pixels to frequency values, being each pixel in the vertical axis equivalent to 1 kHz. Then, a median filter is applied to the image in order to reduce the grain level (see Figure 5.10c).

In order to identify the regions, or pixels, of the image that constitutes the vocalization, the Maximally Stable Extremal Regions (MSER) (see subsubsection 5.2.3.6) were determined, resulting in the identification of regions with similar intensity. After these regions have been computed, a binary image is created using all the pixels that are part of the MSER previously identified (see Figure 5.10e). The binarization process can, sometimes, produce some “gaps” in the main structures of the vocalizations, which are regions not identified by the MSER detection algorithm (see Figure 5.10e). In order to fill these non-identified regions, a dilation process was used. Dilation is a basic morphology mathematical operator. This morphological operator works by gradually enlarge the boundaries of regions of foreground pixels (white pixels in the binary images), which leads to the areas of foreground pixels grow in size while holes within those regions become smaller or completely filled [54]. After the image have been dilated, it is then eroded to regain some of its original shape, and to reduce the noise and irrelevant details. Similarly to dilation, erosion is also a morphology mathematical operator. However, as the name indicates, the erosion process has the reverse effect of the dilation one. The erosion operator works by eroding away the boundaries of regions of foreground pixels (white pixels in the binary image) [54]. This process results in the size shrinking of the foreground pixel areas and in the enlargement of holes within those areas.

The next step is the determination of the properties of all the binary regions in the images. This step aims to remove the noise from the binary images. After determining the regions properties, the ones which present an eccentricity inferior to 0.9 or an area inferior to 10% of the area of the bigger region, are excluded (see Figure 5.10h).

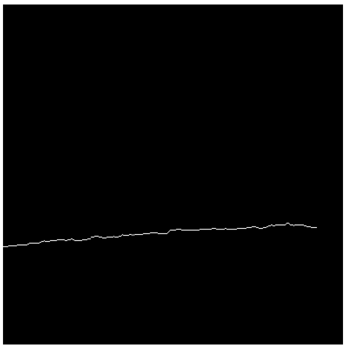
Finally, the edge shapes of the remaining regions are determined and, consequently, the vocalizations contours, which corresponds to the middle inner portion of the edge contours of each element (see Figure 5.10i and Figure 5.10j).



(a) Original Image. **(b)** Processed Image. **(c)** Median Filter.

(d) MSER. **(e)** MSER Binary. **(f)** Image Dilation.

(g) Image Erosion. **(h)** Regions Removal. **(i)** Edge Contour.



(j) Final Contour.

Figure 5.10: Image Based Time-Frequency Ridge Extraction method steps.

5.2.3.2 Contour Based Features

After the contour of the vocalizations has been extracted with the algorithms described in subsubsection 5.2.3.1, several descriptors can be obtained from it (see Figure 5.11). The contour based descriptors play an essential role in the description of the vocalizations, since most USV classes (see section 2.4) are defined based on frequency variations on the time-frequency ridges. Therefore, these group of features are intrinsic to the classes themselves, demonstrating their discriminative power and importance on USVs analysis.

The features extracted form the vocalization contour were the following ones:

1. **Peak frequency:** Which corresponds to the highest frequency value;
2. **Minimum frequency:** Which corresponds to the lowest frequency value;
3. **Initial frequency:** Which corresponds to the frequency at the initial temporal instant;
4. **Final frequency:** Which corresponds to the frequency at the final temporal instant;
5. **Frequency bandwidth:** Which corresponds to the difference between the highest and the lowest frequency values;
6. **Difference maximum and initial frequencies:** Which corresponds to the difference between the peak frequency and the frequency at the initial instant;
7. **Difference maximum and final frequencies:** Which corresponds to the difference between the peak frequency and the frequency at the final instant;
8. **Difference minimum and initial frequencies:** Which corresponds to the difference between the lowest frequency and the frequency at the initial instant;
9. **Difference minimum and final frequencies:** Which corresponds to the difference between the lowest frequency and the frequency at the final instant;

10. **Difference final and initial frequencies:** Which corresponds to the difference between the frequency the final and initial instants;

11. **Number of directional changes:** Which corresponds to the number of times the trend of contour vector has changed direction;

12. **Duration:** Which correspond to the time between the initial and final instants;

13. **Trend:** Binary feature that translates the global behavior of th USV. Has the value 1 if the final frequency is bigger than the initial, -1 in the opposite situation, and 0 if both frequencies are equal.

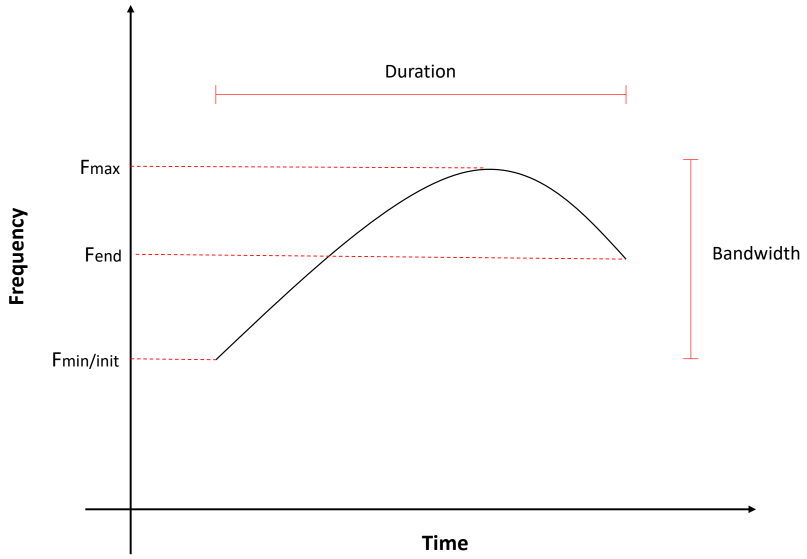


Figure 5.11: Example of contour features extraction.

5.2.3.3 Number of Frequency Steps

The number of frequency steps is a feature that translates the number of discontinuities or "steps" in the time-frequency representation of the vocalizations. This is a very useful descriptor since there are classes that are defined precisely according to the presence of these "steps". Yet, this is not trivial feature to extract, mostly due to the quality of the vocalizations.

In order to estimate the number of "steps", first, the probable instants (white lines in Figure 5.12) in which might exist a discontinuity are determined with a change point detection algorithm[6]. After that, the probable change points are iterated and their neighborhood is analyzed, being determined the most powerful frequencies in a certain number

[6] https://www.mathworks.com/help/signal/ref/findchangepts.html

of samples for each side, left and right. Then, an average value of the frequencies to the left and right is computed and if they differ more than a certain established threshold, that instant is considered to be a frequency step.

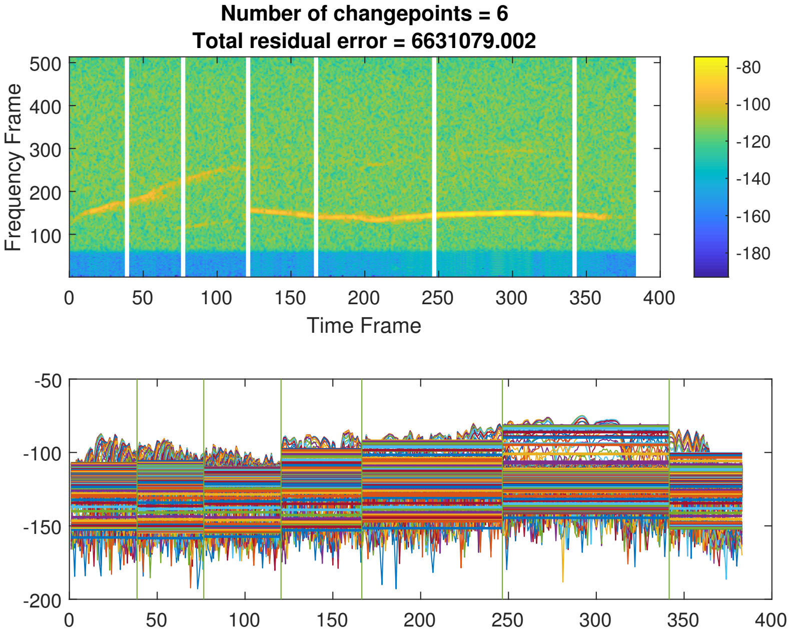


Figure 5.12: Number of frequency breaks estimation (Example).

5.2.3.4 Local Binary Patterns

Local Binary Patterns (LBP) are a set of descriptors that encode local texture information typically used in computer vision problems. In order to compute this set of descriptors, a gray-scale image with 250-by-250 pixels of the time-frequency map of the USVs is determined using the STFT. Then, LBP features are computed through the analysis of neighborhood of each pixel in the image individually. Then, to each pixel is assigned a binary code, with the same number of elements as neighbors, where each element has the value 1 if its intensity value is bigger that the central pixel, or 0 if it is smaller. After that, each pixel binary code is converted to a integer number, being the output result a matrix where each element is the result of this conversion.

Therefore, these descriptors attempt to capture the texture of the image and are insensitive to orientation and scaling.

5.2.3.5 KAZE Features

KAZE features are a multiscale 2D feature detection that exploit non-linear scale space through non-linear diffusion filtering [55].

The KAZE features detector is based on the scale normalized determinant of Hessian Matrix which is computed at multiple scale levels. Then, the points of maximum response to the detector are selected up as feature-points using a moving window [56]. Feature description introduces the property of rotation invariance by finding dominant orientation in a circular neighborhood around each detected feature. Then, the main orientation of each key-point is determined as well as a rotation invariant using first order image derivatives [55]. Therefore, KAZE features are invariant to rotation and scale.

In order to extract KAZE features, the USVs time-frequency maps were determined with the STFT and converted to 250-by-250 pixels gray-scale images.

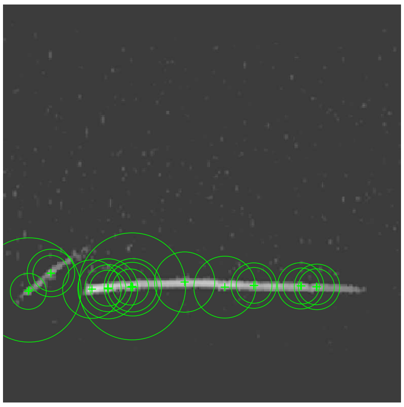


Figure 5.13: KAZE features example (Example).

5.2.3.6 MSER

In order to extract MSER, the USVs time-frequency maps were determined with the STFT and converted to 250-by-250 pixels gray-scale images.

The MSER are used as a method of blob detection in images. MSER detection implicitly takes advantage of the robustness of boundary-related structures without detecting them and is, in its essence, an intensity-based region detection dealing with connected components and extracting extreme regions that are stable to intensity perturbations [57].

Plus, using the boundaries information in order to detect the MSER confers robustness to the algorithm, and it also allows to capture relevant information of the useful image parts [57].

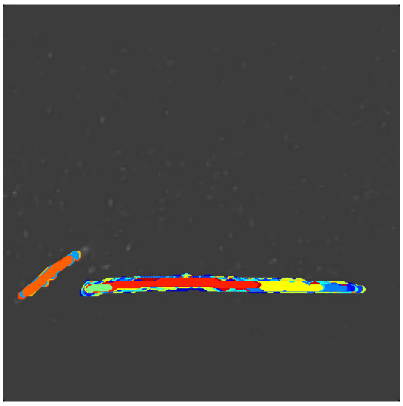


Figure 5.14: MSER example (Example).

Features Summary

All the features extracted which were used in these work, both in the segmentation phase and in the classification phase, are represented in Figure 5.15.

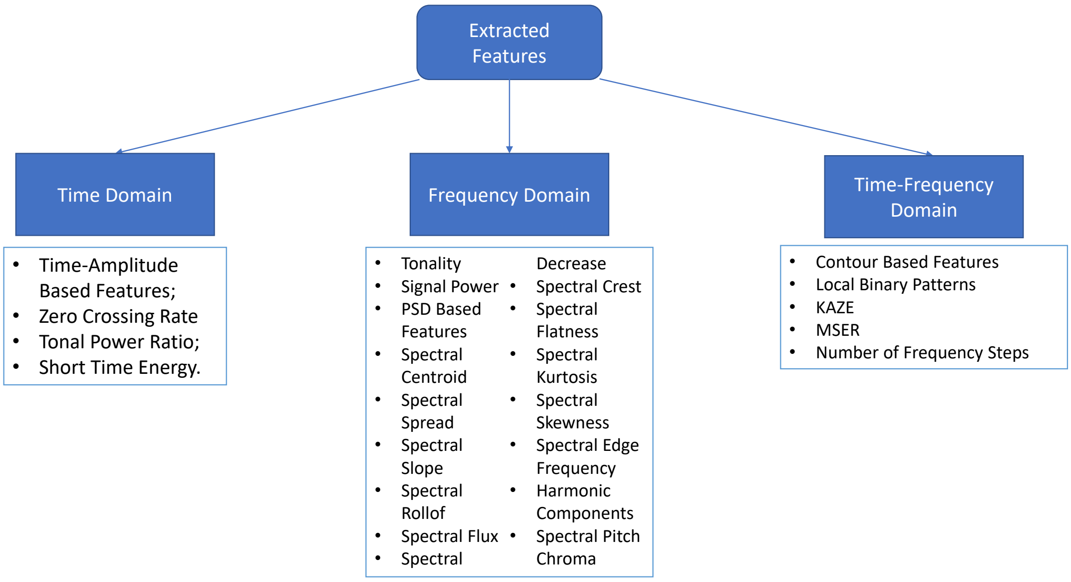


Figure 5.15: Features summary.

5.3 Classification

In this section the models used to build the classifiers are described, as well as some pre-classification techniques typically use to maximize the performance of the classifiers.

5.3.1 Data Normalization

Usually in ML problems, a common practice is the normalization of the data. The data normalization allows to represent the different characteristics in the same range of values which might help to avoid the emergence of a negative trends in the classifiers.

In this work, the *z-score* normalization was used. The *z-score* measures the distance of a data point from the mean, describing that distance in numbers of standard deviations. Thus, the *z-score* normalization is given by:

$$z = \frac{(x - \mu)}{\sigma} \tag{5.21}$$

where μ is the mean value of the data and σ the standard deviation.

5.3.2 Feature Selection

In ML, feature selection is the process where a subset of features/descriptors are identified from the original dataset in order to develop classification models. Feature selection techniques are extremely useful in ML and are used, mainly, due to the following reasons: 1) Diminish the problem dimensionality, and, therefore, simplifying the models, making them faster to train and preventing from overfitting; 2) Exclude some features which might not contribute to the classification; 3) Exclude redundant features that provide the same information; 4) And, mainly, find the best combination of features in order to achieve the highest performance as possible in classification.

Feature selection methods can be classified in three main types: filters, wrappers or embedded methods. Filter methods are independent of the used classifier and are fast and generic methods. These types of methods present as main disadvantage the fact that they are only based on the data itself, therefore, no classification performance assessment is made. On the other hand, the wrapper and embedded methods may present better results since the feature selection is performed together with a classifier. This solutions tend to be heavier in computational terms, and provide a less general solution.

In this project, only filter methods were used to perform feature selection.

5.3.2.1 Kruskal-Wallis

Kruskal-Wallis is a non-parametric statistical test typically used to determine features discriminative power. The test works by sorting the feature values and assigns ordinal ranks in order to find their discriminative power. The sums of these ranks for the classes are then used to compute the value of H statistics, which reflects the difference of the ranks' sums. The H values are given by:

$$H = \frac{12}{n(n+1)} \sum_{i=1}^{c} n_i (R_i - \bar{R})^2 \tag{5.22}$$

where R_i is the average of ranks for the samples belonging to class i, $\bar{R}$ is the average of all ranks for all classes, n_i is the number of patterns belonging to class i and n is te total number of patterns.

5.3.2.2 RelliefF

ReliefF is an attribute estimator algorithm which finds the weights of predictors in single or multi-class categorical variable. This is a filter method typically robust and noise tolerant [58]. This method determines a score for each feature which is then applied to rank and select top scoring features.

ReliefF works by searching for k the nearest neighbors from the same class, called nearest hits Hj, and also k nearest neighbors from the other classes, called nearest misses $Mj(C)$. After that, the weight value for each feature j is updated based on the following expression:

$$W_j = \frac{W_{dy \wedge dj}}{W_{dy}} - \frac{W_{dj} - W_{dy \wedge dj}}{m - W_{dy}} \tag{5.23}$$

where W_{dy} is the weight of having different values for the response y; W_{dj} is the weight of having different values for the predictor Fj and $W_{dy \wedge dj}$ is the weight of having different response values and different values for the predictor Fj.

5.3.3 Dimensionality Reduction

5.3.3.1 Principal Component analysis

Principal Component Analysis (PCA), is a feature transformation method that focus on reducing data dimensionality while preserving its "variability". Therefore, the main goal of PCA is to represent a large set of descriptors into a smaller one while preserving most of its relevant information quantified by data variance.

5.3.4 Classifiers and Models Description

In order to develop the classification models several classifiers were tested, which are described bellow. All the classifications models are based on a supervised learning methodology, thus, all data provided to the classifiers were previously annotated.

5.3.4.1 Support Vector Machine

SVMs have been used in multiple bioacoustic studies, having shown successful results [31]. SVMs are binary classifiers, which means that they can be used do discriminate between 2 classes. However, they can also be used in multi-class problems using different techniques to combine multiple two-class SVMs in order to build a multi-class classifier [59, 60]. This classifier works by determining the decision hyperplane, which maximizes the separation in data from different classes (see Figure 5.16), so it is a maximum margin classifier [59].

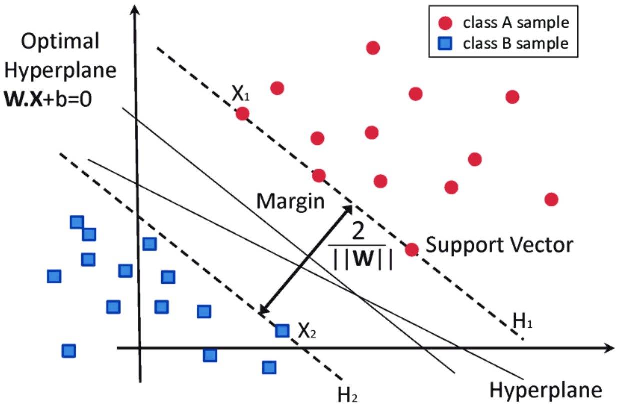


Figure 5.16: SVM[7]

SVMs determine the support vectors for each class, which are at a distance $1/||\mathbf{w}||$ from the separation hyperplane. Therefore, these vectors will determine the margin of separation. The principle of SVMs is to maximize of the separation margin $(2/||\mathbf{w}||)$, which is translated by the minimization of the criterion $\psi(\mathbf{w})$, given by:

$$\psi(\mathbf{w}) = \frac{1}{2}||\mathbf{w}||^2; \quad \text{subjected to} \quad y_i(\mathbf{w}'x_i + b) \geqslant 1, with : i = 1, .., n \tag{5.24}$$

Even tough the fact that the typical SVM is a linear classifier, it can also be a non-linear classifier with the use of a kernel function. The idea behind a kernel is the mapping of non-linear separable datasets into a higher dimensional space where it is possible to find a hyperplane that can linearly separate the samples. There are several kernels available, being some of the more common the linear, polynomial, Gaussian, RBF and Sigmoid ones.

In their original formulation, SVMs are hard margin classifiers, which means that the support vectors try to completely separate the data from the classes involved, not allowing any misclassifications. The hard-margin approach, often leads to misfitted models which lack generalization capability, since the decision borders are too strict and have low tolerance to outliers or mislabeled data. There is, however, a parameter C (Cost), which can be used to handle this problem. The parameter C, uses a soft-margin approach, giving to model the possibility of allowing some missclassfied data when the data is being fitted. The higher the C value is, the smaller is the classifier marging, and, therefore, the less permissive to missclassifcation the model is, and vice-versa. This effect can be seen in Figure 5.17.

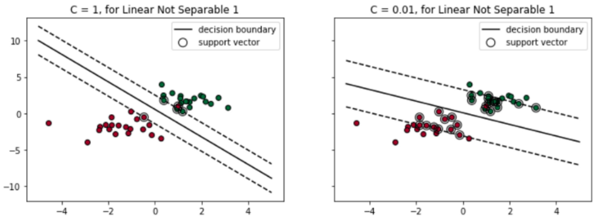


Figure 5.17: Effect of C value in SVM model[8]

Moreover, multi-class SVMs can also use several coding matrices to create multiple binary problems. A coding matrix is a matrix whose elements determine which classes are trained by each binary learner, that is, how the multi-class problem is reduced to a series of binary problems. Some of the more common coding designs are the One-vs-All and One-vs-One. All SVM models used in the classification phase were designed with a One-vs-One coding matrix. This means that for each binary learner, one class is positive, another is negative, and the remaining ones are ignored. This design exhausts all combinations of class pair assignments. In this process, a class is assigned to a pattern based on a majority voting process.

[7] https://www.researchgate.net/figure/Classification-of-data-by-support-vector-machine-SVM_fig8_304611323

[8] https://towardsdatascience.com/support-vector-machine-simply-explained-fee28eba5496

5.3.4.2 k-NN

k-NN models, which stand for k Nearest Neighbors, is a classification algorithm also commonly used in bioacoustic studies [28,31,61]. k-NN is a non-parametric method, since no assumptions regarding the data distribution are made. The general expression for non-parametric probability density estimation is given by

$$p(x) \approx \frac{k}{nV} \tag{5.25}$$

where V is the volume surrounding x, n is the total number of samples and k is the number of patterns inside V. The k-NN method results of the fixation from the value k and the determination of the minimum value for V.

The k-NN classifier assigns the class to a new pattern according to the class labels that is found in majority among the k nearest neighbors. Thus, the value o k will affect the algorithms' performance. For example, in Figure 5.18, if k is equal to 3, the green circular pattern will be classified as red, however if k is equal to 5, the pattern will be assigned to the blue class, given the fact that there are more blue patterns in the neighborhood. There are several metrics that can be used to determine the distance between data points, such as Euclidean, Mahalanobis, Cityblock, among others, being that the metric must be chosen according to the problem and its restrictions.

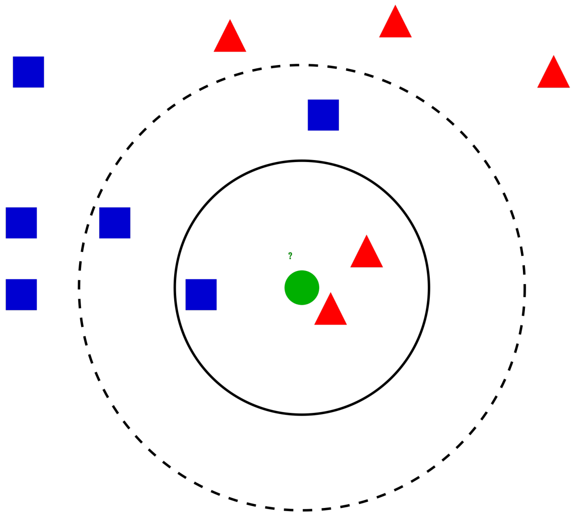


Figure 5.18: k-NN Example[9]

Furthermore, the values of k will also impact the shape and complexity of decision boundaries. If the value of k is too small, the decision boundaries tend to be more complex, which may lead to phenomenons of overffiting [59](see Figure 5.19). The inverse situation is also verified, as higher values of k tend to form flatter decision boundaries that may lead to underfitting.

[9] https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/

[10] https://i.imgur.com/dVYBfHo.png

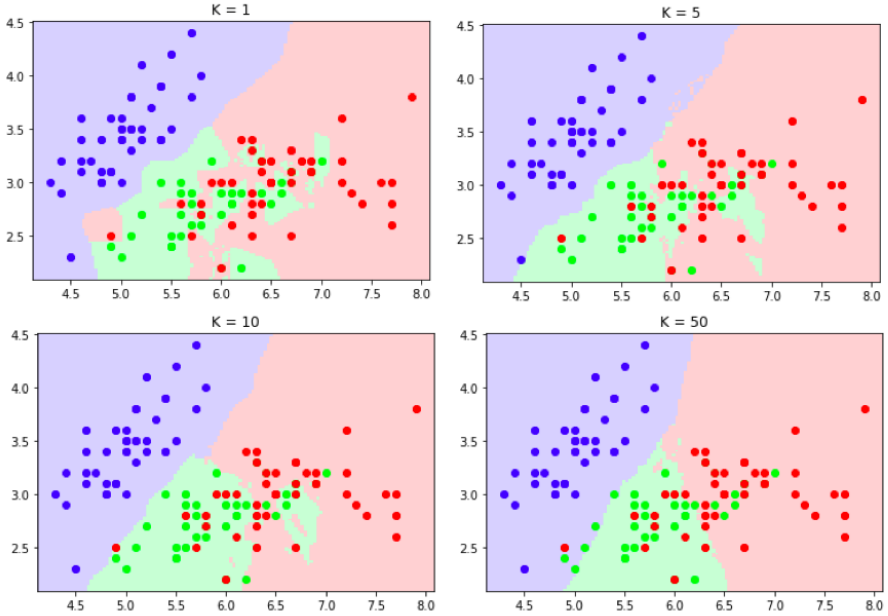


Figure 5.19: k-NN number of neighbors effect[10].

5.3.4.3 Decision Tree

Decision trees are a non-parametric method used for both classification and regression problems. The goal of these classification algorithms is to create a model that predicts the value of a target variable based on several input variables, and, to do that, tree like structures are built (see Figure 5.20), as the name indicates, with a "if ... then ... else ..." construction. In these "trees", each interior node represents a predictors, each branch represents a decision rule and each leaf, being the terminal node, represents an output (a categorical or continuous value). Thus, the classification process is done following the chained decision rules until the terminal node is reached.

Decision Trees typically grow in a top-down way, and growing a decision tree involves deciding at each step on which features to choose and what conditions to use for splitting, along with knowing when to stop growing the tree itself. The decision rules, are based of a single feature and threshold, which tries to look for the combination that splits the data into the purest subsets. This keeps happening recursively, until the maximum depth of the tree is reached, or until the data impurity can no longer be reduced. There are several criteria used to measure impurity, such as the Gini impurity or the entropy value [62].

This type of models has as main advantages its interpretability, since they use simple chained decision rules. However, they may also be more susceptible to overfitting phenomena.

[11] http://www.learnbymarketing.com/methods/classification-and-regression-decision-trees-explained/

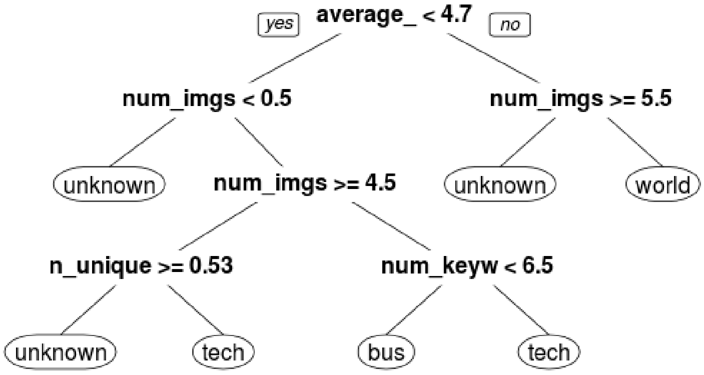


Figure 5.20: Decision Tree Example[11].

5.3.4.4 Tree Ensemble

The Tree Ensemble classifier is an ensemble method which combines several decision trees to produce better predictive performance than utilizing a single decision tree. The ensemble models are based on the premise that when a group of weak learners is joined together a strong learner is formed, which in this concrete case can be translated to the performance of several decision trees joined together being better than each tree by itself.

There are two main techniques that are used to develop Tree Ensemble classifiers and ensemble methods in general , which are *Bagging* and *Boosting*. The general idea behind the *Bagging* process is to training a set of classification models in a parallel way, where each model is trained by a random subset of the data, while in the *Boosting* process is to train a set of classification models in a sequential way, where each individual model learns from mistakes made by the previous model (see Figure 5.21). While *Bagging* ensembles are more adequate to high dimensional data problems, *Boosting* methods are better in low dimensional context and support multiple aggregation methods *AdaBoost*,*GentleBoost*,*LSBoost*, among others.

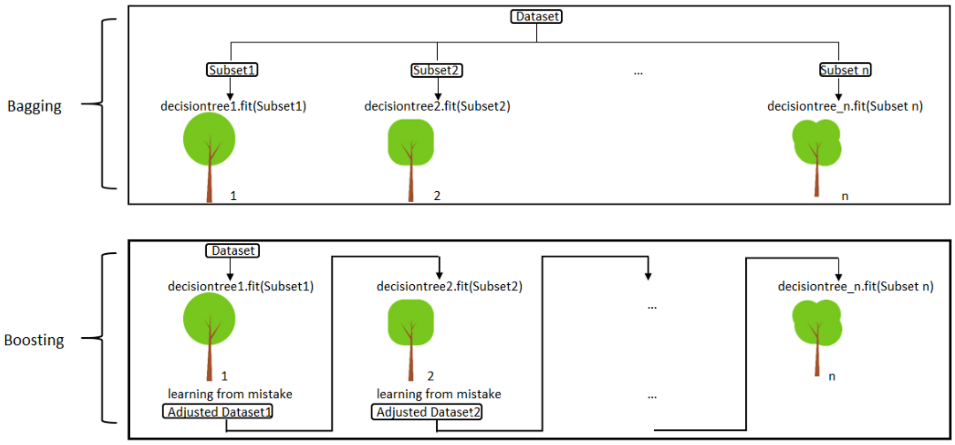

Bagging
Dataset
Subset1
Subset2
...
Subset n
decisiontree1.fit(Subset1)
decisiontree2.fit(Subset2)
decisiontree_n.fit(Subset n)
1
2
n
Boosting
Dataset
decisiontree1.fit(Subset1)
decisiontree2.fit(Subset2)
...
decisiontree_n.fit(Subset n)
1
2
n
learning from mistake
Adjusted Dataset1
learning from mistake
Adjusted Dataset2
...

Figure 5.21: Bagging vs Boosting[12].

5.3.4.5 Fisher Discriminant Analysis

Fisher Discriminant Analysis or Fisher's linear discriminant is part of of a broader type of classification techniques entitled Discriminant Analysis. This classification method works by projecting the data into a smaller dimension where the separation between classes and compactness within each class of the data are maximized (Fisher Criteria), as demonstrated in Figure 5.22. Thus, a way to view a linear classification model like this, is in terms of dimensionality reduction. After the data has been projected, according to the Fisher criteria, a linear decision hyperplane is determined to separate the data from the multiple classes. Note that this method is by default a binary classifier, however it can also be used in multi-class problems using multiple binary classifiers in an One-Against-All classification schemes [59].

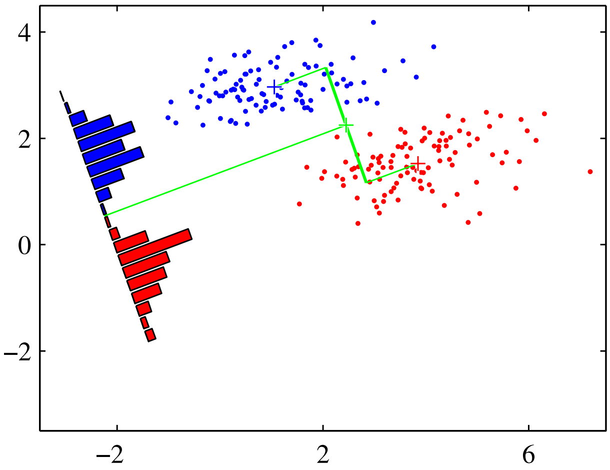


Figure 5.22: LDA data projection (Adapted from [59]).

5.3.5 Evaluation Metrics

There are several metrics that can be used to access classification algorithms performance, however, not all of them can give a real perception of how good or bad the classifiers really are. For instance, some metrics can be deceiving in cases of extremely unbalanced datasets.

Five of the most common metrics used in machine learning, which are the ones also used in this work, are Accuracy, Precision, Recall and Specificity and F_1 Score, whose equations are as follows:

$$Accuracy = \frac{TN + TP}{TN + FP + FN + TP} \tag{5.26}$$

$$Specificity = \frac{TN}{TN + FP} \tag{5.27}$$

[12] https://towardsdatascience.com/basic-ensemble-learning-random-forest-adaboost-gradient-boosting-step-by-step-explained-95d49d1e2725

$$Precision = \frac{TP}{TP + FP} \tag{5.28}$$

$$Recall = \frac{TP}{TP + FN} \tag{5.29}$$

$$F_1 Score = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{5.30}$$

where,

1. **True Positives (TP):** Corresponds to the number of samples correctly classified as positive;

2. **True Negatives (TN):** Corresponds to the number of samples correctly classified as negative;

3. **False Positives (FP):** Corresponds to the number of samples wrongly classified as positive;

4. **False Negatives (FN):** Corresponds to the number of samples wrongly classified as negative;

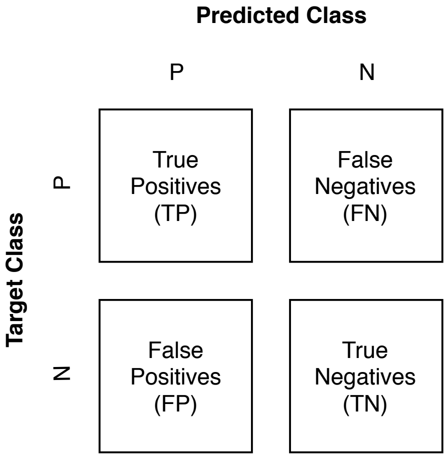
Apart from Accuracy, all metrics mentioned are used in binary problems, and, since the problem to evaluate is a multi-class one, some adjustments had to be made. Therefore, in order to calculate the value of Precision, Recall and Specificity and F_1 Score, were considered several binary problems, where each one of the existing classes is considered against all of the remaining ones. Consequently, for each class, the TP and FP values correspond to the number of samples from the class correctly and wrongly identified, accordingly, while the TN and FN values correspond to the number of samples from all the remaining classes combined that were correctly and wrongly identified as belonging, or not, to the class.

Another tool used to access how the classifiers behave and what are the classes that are most mismatched with each other, is the confusion matrix, in which the true labels of the samples are represented against the output produced by the classifiers (see Figure 5.23).

5.3.6 Experimental Setup

In this subsection the whole process used to elaborate the classification models is described as well as which measures were taken to avoid typical classification problems or bottlenecks.

In order to avoid overfitting phenomena, all the annotated data were divided into a training and test dataset. The training dataset will then be used to perform feature selection and reduction as well as classifiers hyperparameters optimization, with a k-fold cross-validation methodology. The k-fold cross-validation method is based on the the partition of the dataset in k equally sized folders. After the partition of the data, k-1 of the total partitions are used to train the models and the remaining one is used for validation



		Predicted Class	
		P	N
Target Class	P	True Positives (TP)	False Negatives (FN)
	N	False Positives (FP)	True Negatives (TN)

Figure 5.23: Confusion Matrix Example.

purposes. The process is the repeated k times, being each of the partitions, used at a time, to access the performance of the classifications models (see Figure 5.24). After all iterations have been completed, the mean value as well as the standard deviation are extracted, giving a more realistic and accurate value regarding the classification performance. The k-fold cross-validation method also has a stratified version, in each the data partition keeps the the same percentage distribution of all classes as in the original dataset.

Thus, several classifications models were developed and optimized based on the training data. These models, with the best parameters found based on the training dataset, will then be tested on the test dataset. Therefore, the test dataset will serve only to test the performance the developed algorithms with the training data, never being used to perform any kind of optimization on the models. This way, the complete separation in test and training sets ensures an external validation element, since none of data belonging to the test set has been used in the models development.

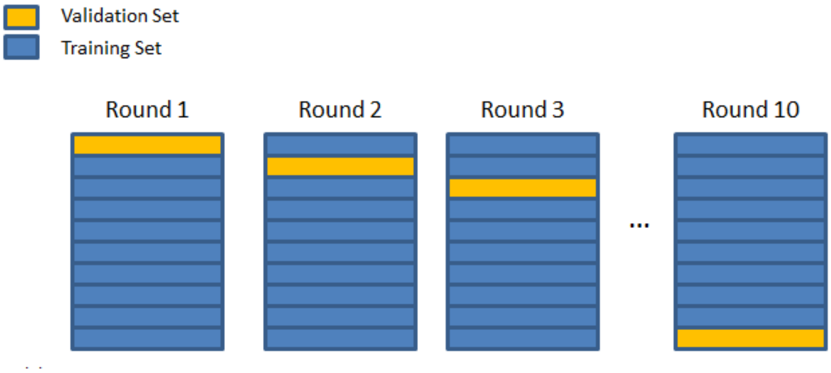


Figure 5.24: 10-fold cross Validation Example[13].

[13] https://codesachin.files.wordpress.com/2015/08/cv.png

The whole training, validation and test process is illustrated in Figure 5.25.

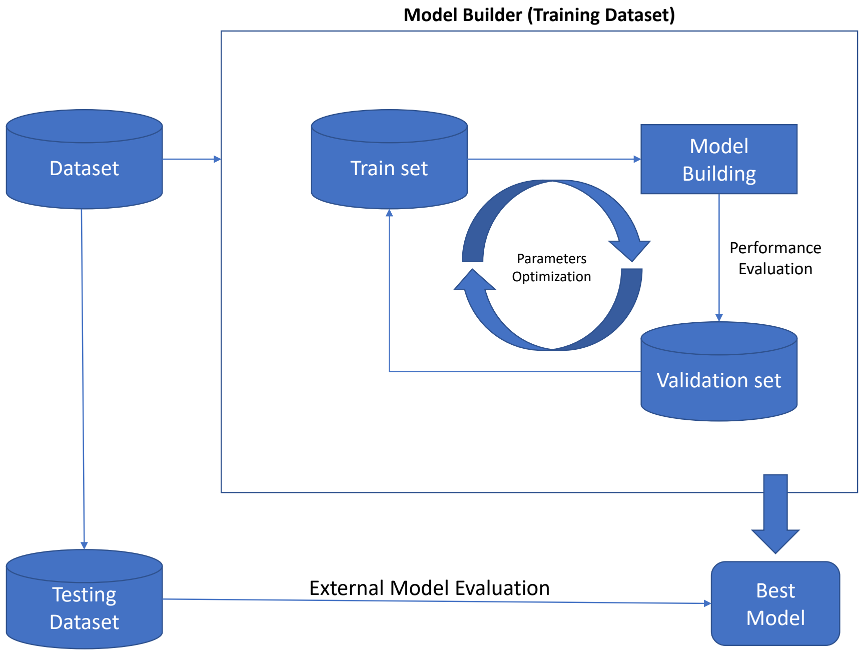


Figure 5.25: Models development diagram.

Chapter 6

Results and Discussion

Since in these work several algorithms regarding segmentation and classification of Ultrasonic Vocalizations (USVs) have been proposed, in this section the according results, obtained in each of the USVs processing phases, are presented and discussed. This chapter is divided in two main sections, namely the segmentation and classification ones.

6.1 Segmentation

In this section the methodology used to evaluate the proposed segmentation algorithm is show as well as the results obtained and the according discussion.

6.1.1 Segmentation Results

To evaluate and develop the proposed algorithm, 16 audio files (.WAV), sampled at 500 kHz, with 45 s each, were considered and manually segmented. The files contain spontaneous vocalizations originated from the different social contexts in which the signals were acquired (see section 4.2). In total, the signals used contained 460 vocalizations.

In order to evaluate the algorithm's performance, the results obtained by the automatic segmentation algorithm were compared to the results obtained with the manual annotation and precision and recall were considered as performance values (see equations 6.1 and 6.2).

$$Precision = \frac{tp}{tp + fp} \tag{6.1}$$

$$Recall = \frac{tp}{tp + fn} \tag{6.2}$$

In Equation 6.1 and Equation 6.2, the true positives value (tp) represents the number of vocalizations properly segmented, the false positives value (fp) represents the number of instances detected as vocalizations by the automatic algorithm but do not correspond to real vocalizations and the false negatives value (fn) represents the number of real calls that were not detected by the algorithm.

Furthermore, to test the robustness of the algorithm, different levels of Gaussian white noise were added to the signal before being segmented. This way, it was possible to check if there were any significant drops in performance and at which noise levels.

As mentioned in subsection 5.1.1, the segmentation algorithm has two different thresholds with different purposes. The first threshold was estimated by observation in order to be as higher as possible without reaching the top baseline of the spectral entropy of the signals (about 0.98) and it was settled to 0.97, whereas the second one was tested with multiple values. The search process for the value of the second threshold is illustrated in Figure 6.1, where the values of precision and recall obtained with the segmentation algorithm for the different values of the second threshold are represented.

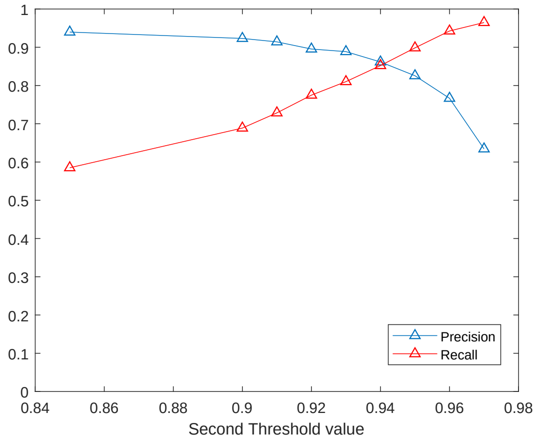
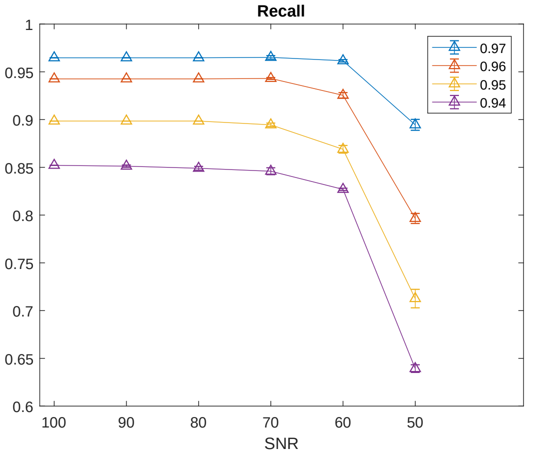
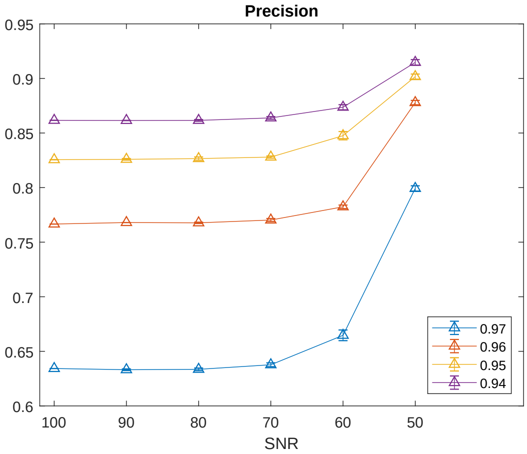


Figure 6.1: Precision and recall across the different values tested for the second threshold.

After finding the range of values for the second threshold corresponding to the best performance, {0.94;0.95;0.96;0.97}, the algorithm was tested with Gaussian white noise addition, and several levels of signal to noise ratios (SNRs) were considered. For each value of SNR, the segmentation algorithm was tested five times, and then the mean and the standard deviation of the precision and recall values were calculated (see Figure 6.1).



(a) Recall values for different SNRs.

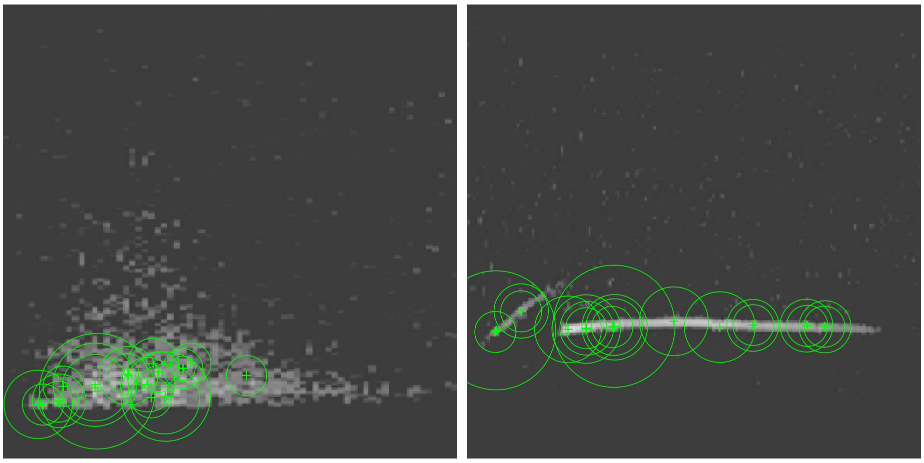


(b) Precision values for different SNRs.

Figure 6.1: Algorithm's performance with different levels of noise addition.

6.1.1.1 BOVW Training and Results

To train the Bag of Visual Words (BOVW) classifier, 1600 images were used, 800 images of real vocalizations and the remaining 800 of noise images. The images were obtained from the sounds time-frequency representation, which were determined using the Short-Time Fourier Transform (STFT) with a Hamming window with 256 points, 50% of overlap and 1024 sampling points to compute the discrete Fourier transform. Afterwards, all images were resized to a 250-by-250 pixels gray image, in order to reduce computational load while processing the images and extracting features. As mentioned before, KAZE features were used to extract the key interest points of each image (see Figure 6.2), using a dense representation and extraction with multiple scales (x1.6 and x3.2).



(a) Noise **(b)** Vocalization.

Figure 6.2: Example of KAZE feature extraction.

To assess the performance of the classification algorithm, the group of 1600 images was divided, with 30% of the data used to train the BOVW and 70% used for testing. The according confusion matrix is represented in Table 6.1.

Table 6.1: Confusion matrix for Bag of Visual Words method (Training).

Known \ Predicted	Noise	Vocalization
Noise	0.99	0.01
Vocalization	0.08	0.92

Using the BOVW classifier after the segmentation algorithm it was possible to diminish the false positive rate, and, therefore, increase the precision value. The results obtained with the segmentation algorithm and the images classifier are listed in Table 6.2. The precision value has increase by 11 % and the recall value decreased by 3%, in comparison with the results obtained with the two thresholds used in the segmentation algorithm set to 0.97.

Table 6.2: Precision and Recall values for the segmentation algorithm plus the BOVW classifier.

Algorithm	Recall	Precision
Segmentation Algorithm (Threshold 1=0.97) + BOVW	0.94	0.74

6.1.2 Segmentation Discussion

Through the analysis of Figure 6.1 it is possible to observe that the value chosen for the second threshold establishes a clear trade-off between precision and recall. The closer this value is to the value of the first threshold (0.97), the higher is the number of vocalizations the algorithm is able to recognize, and, therefore, the higher are the true and false positives rates, which corresponds to a higher recall and lower precision values, accordingly. This happens because spectral entropy translates, in some way, the disorder introduced in the system. So, if the second threshold is higher, the algorithm is more permissive since the sounds it recognizes as syllables do not have to be as strong in terms of power. On the other hand, if the value of the second threshold is lower, the precision value is far superior to the recall one, which means that the majority of the sounds recognized are indeed real vocalizations. However, the algorithm is unable to segment a bigger number of calls, especially the ones with lower spectral power, resulting in a lower recall value.

As previously mentioned, the segmentation algorithm was also tested with the addition of Gaussian white noise to the signals. Through the analysis of Figure 6.1, it is possible to observe that for values of Signal to Noise Ratio (SNR) higher than 60, both the recall and precision values are kept relatively constant for all values of the second threshold used, being the standard deviation between runs very small or even zero. Thus, as the algorithm's performance does not significantly change for low levels of SNR, it is possible to consider that the algorithm is robust to noise.

Using a fully automated segmentation algorithm based on spectral entropy has been proved to be an effective method to isolate mice vocalizations, since the algorithm was able to detected up to 97% of the vocalizations in the signals annotated for this study. However, the proposed method tends to have a higher rate of false positive vocalizations in order to achieve higher recall values, which can be observed in 6.1. This phenomenon can be explained by the fact that the signals used in this work contain lots of noisy sounds with similar power when compared to the real vocalizations, which led to the increase of the false positives rate. However, the rate of false positive vocalizations can diminish with the fine tuning of the lowest threshold used in the algorithm, since it was possible to obtain 0.85 and 0.86 of recall and precision values with the threshold set to 0.94, and 0.90 and 0.83 with 0.95, accordingly, which provides a more balanced, yet good, performance.

The addition of the BOVW classification as a pos-processing technique in order to reduce the the rate of false positive detections has also been shown to be effective since the precision value increased has by 11%. This classifier has presented great results in the training phase, however, when used after the spectral entropy, the results are lower than in training. This behavior is somehow expected, since BOVWs are classifiers which require lots of data in order to produce the dictionaries of visual words. Thus, with the increase of the dataset size, an increase in performance is expected to be observed, as well as the computational resources needed to develop the classifier, potentiating even more this approach.

6.2 Classification

In this section the results obtained regarding the classifications developed models are presented as well as the evaluation of the contour extraction methods and the according discussion of the obtained results.

6.2.1 Classification Results

The first step to undertake in order to develop and test the classification models was the partition of the dataset in two different subsets, a training set containing 70% of the whole data, and a testing set with 30% of the data, accordingly. Therefore, the training set is used to train and validate the developed models while the testing set is used as an independent set, used exclusively to test the models with unseen data. The testing set is never used to perform any type of parameters adjustments while developing the classification models. Both the training and testing sets present the same class prevalence, being the number of samples from each class represented in Table 6.3. Through the analysis of Table 6.3, it is possible to observe that the class distribution is very uneven. Thus, in order to avoid classification bias, all the classification methods used were set with the prior probability of assigning each class as uniform.

Table 6.3: Training and Testing sets classes distribution.

Class	Training Set	Rel. Freq.	Test Set	Rel. Freq.
Complex	33	0.010	13	0.009
1 Frequency Step	464	0.143	198	0.143
M. Frequency Steps	107	0.033	47	0.034
Upward	1246	0.384	534	0.384
Downward	257	0.079	111	0.080
Flat	663	0.204	284	0.204
Short	272	0.084	118	0.085
Chevron	95	0.029	40	0.029
Reverse Chevron	38	0.012	16	0.012
Composite	68	0.021	28	0.020
Total	3243	1	1389	1

Since contour based features play such an important role in USVs classification and several algorithms where developed to extracted it, the first step is to properly assess which of the methods better extracts the time-frequency ridges of the vocalizations. Having this in mind, all the contour extraction algorithms (see subsubsection 5.2.3.1) were used to extract the contour based features, and, afterwards, with the extracted features, classifiers were trained. Therefore, to assess the contour extraction algorithms performance, each of the classifiers was tested with each of the algorithms, repeating a 10-fold cross-validation methodology five times, in order to train and validate the models. This procedure produced fifty accuracy values distributions, one for each of the methods with each of the classifiers. After that, in order to determine which is the best contour extraction method, statistical tests were conducted in order to evaluate the statistical differences between the obtained distributions. The mean accuracy values obtained with each of the Time Frequency Ridge Extraction (TFRE) methods and classifiers are presented in Table 6.4.

Table 6.4: Mean Accuracy values from the distributions for all methods and classifiers (5 times 10-fold Cross-Validation).

	SVM	k-NN	Decision Tree	Fisher-LDA	Tree Emsemble
TFRE max	70.04±2.13	60.01±2.61	65.71±2.43	67±2.47	76.64±2.38
TFRE max tonality	55.54±2.49	44.26±2.58	48.66±2.54	53.97±2.1	63.73±2.83
TFRE max interpolation	73.08±2.17	60.33±2.44	65.09±2.6	58.67±2.42	75.13±2.22
TFRE best path	63.16±2.55	52.31±2.32	53.07±2.86	61.35±2.27	67.55±2.24
TFRE fsst	66.87±2.36	53.97±2.15	55.65±2.95	63.45±2.58	69.79±2.36
TFRE wsst	66.69±2.32	57.84±2.31	58.08±2.39	64.58±2.43	71.87±2.1
TFRE image	67.49±2.66	56.05±2.52	61.91±3.02	58.95±2.79	75.01±2.14

The first step to take in order to study the statistical significance of the difference between the distributions obtained from each method, is to access their normality, so it can be decided whether to use parametric or non-parametric statistical tests. So, the Shappiro-Wilk test was applied with a significance level of 95% ($alpha = 0.05$). The p-value results are presented in Table 6.5. The results obtain with the Shappiro-Wilk statistical test were also corroborated with a visual inspection with the histograms of all the distributions in cause.

Table 6.5: p-Values Shappiro-Wilk normality test.

	SVM	**k-NN**	**Decision Tree**	**Fisher-LDA**	**Tree Emsemble**
TFRE max	0.885	0.196	0.494	0.026	0.494
TFRE max tonality	0.388	0.168	0.002	0.904	0.54
TFRE max interpolation	0.234	0.271	0.092	0.399	0.666
TFRE best path	0.141	0.807	0.162	0.883	0.281
TFRE fsst	0.931	0.616	0.539	0.137	0.532
TFRE wsst	0.819	0.498	0.391	0.5	0.463
TFRE image	0.202	0.007	0.098	0.277	0.019

Through the analysis of Table 6.5 values it is possible to observe that, apart from the SVM case in which all distributions are normal, all classifiers have, at least, a distribution for one of the methods in which the p-value is smaller than the *alpha* value (0.05), meaning that at least one of the distributions can not be assumed as being drawn from a normal distribution. Thereby, to determine if are there any statistical differences between the distributions of all the methods with the different classifiers, non-parametric tests were used, except in the SVM case in which parametric tests were used. Then, to assess the statistical differences, the Kruskal-Wallis and independent-ANOVA tests were used, since the samples are unpaired or unmatched, meaning that the distributions were not obtained under the same conditions. The results are presented in Table 6.6.

Table 6.6: Statistical difference tests.

	SVM	**k-NN**	**Decision Tree**	**Fisher-LDA**	**Tree Emsemble**
p-Value	2.06e-127	8.86e-55	1.20e-60	1.09e-52	4.77e-56
Test	Ind. ANOVA	Kruskal-Wallis	Kruskal-Wallis	Kruskal-Wallis	Kruskal-Wallis

Since all the p-Values in Table 6.6 are smaller than 0.05 (*alpha* value), it is possible to infer that for all classifiers the various distributions present significant statistical differences within themselves. Therefore, in order to determine which time frequency contour extraction performs the best, a pair-wise comparison was performed to the two methods with best performance for each classifier. The two best methods for each classifiers were determined using the boxplots representation (see appendix B) of each accuracy distribution.

Table 6.7: Statistical analysis of the two best methods for every classifier.

	Best Method	2^{nd} **Best Method**	**Test**	**p-Value**	**Sig. Diff**
SVM	TFRE max interpolation	TFRE max	Ind. t-test	2.424e-10	Yes
k-NN	TFRE max interpolation	TFRE max	Mann-Whitney	0.456	No
Decision Tree	TFRE max	TFRE max interpolation	Mann-Whitney	0.223	No
Fisher-LDA	TFRE max	TFRE wsst	Mann-Whitney	1.812e-05	Yes
Tree Emsemble	TFRE max	TFRE max interpolation	Mann-Whitney	1.865e-3	Yes

Through the analysis of Table 6.7, is clearly possible to conclude that the contour extraction methods that performed the best, with almost all classifiers used, were the

Maximum Time-Frequency Ridge Extraction and Maximum Interpolation Time-Frequency Ridge Extraction. Furthermore, it also possible to acknowledge that their performance is quite similar, despite some statistical differences observed in Table 6.7. Having this conclusion in mind, the method used to extract the contour, and afterwards, the development of the classification algorithms, was Maximum Time-Frequency Ridge Extraction, since it was among the top two performers for all the classifiers (see Table 6.7).

The set of features used in the developed Machine Learning (ML) classifications algorithms is composed by an agglomerate of the features described in section 5.2. In total, 128 descriptors were extracted (see Table 6.8), being: 9 from the time domain; 45 from the frequency domain; 74 from the time-frequency domain. Note some of the features described in section 5.2 are a set of multiple values, a distribution, whereat, in those cases, several statistical descriptors were used in order to represent the distribution in a reliable and accurate way.

Table 6.8: Features summary table.

Domain	**Features**
Time	Time-Amplitude Based features
	Zero Crossing Rate
	Tonal Power Ratio
	Short Time Energy
Frequency	PSD Based Features
	Signal Power
	Tonality
	Spectral Centroid
	Spectral Spread
	Spectral Slope
	Spectral Rolloff
	Spectral Flux
	Spectral Decrease
	Spectral Crest Factor
	Spectral Flatness
	Spectral Skewness
	Spectral Kurtosis
	Spectral Entropy
	Spectral Pitch Chroma
	Spectral Edge Frequency
	Harmonic Components
Time-Frequency	Local Binary Patterns
	Contour Based Features
	Number of Frequency Steps
Total Number	*128 Features*

As mentioned before, five different classifiers were trained and tested, but, before directly fed the data into the classifiers, feature selection and feature transformation techniques were used, in an attempt to increase the classification performance. Using the training set and the set of features described in Table 6.8, a method of feature selection joined with analysis of classification performance was used. The feature selection method was implemented with two filter methods (Kruskall-Wallis and ReliefF) and by testing the multiple classification methods with different numbers of features, to determine the best subset of features. The performance results obtained with the feature selection process are represented in Table 6.9 and Table 6.10.

Table 6.9: Weighted mean of F1 Score for feature selection with Maximum Time-Frequency Ridge Extraction and with Kruskal-Wallis (Training Dataset).

# Features	SVM	k-NN	Decision Tree	Fisher-LDA	Tree Ensemble
5	30.40±8.30	54.90±1.30	53.50±1.90	50.10±2.10	61.20±2.40
20	62.70±1.90	65.10±3.40	64.90±2.20	51.60±3.00	73.60±2.50
35	66.90±2.20	62.70±2.50	64.40±3.10	52.80±3.60	75.40±2.40
50	70.30±2.60	63.30±1.70	64.60±1.50	53.50±3.90	75.50±2.50
65	70.60±1.80	62.20±1.60	64.90±2.10	54.00±3.90	76.40±2.20
80	71.80±2.00	60.90±2.30	66.40±2.50	53.50±4.70	75.80±1.80
95	72.90±2.00	60.60±2.10	65.10±2.70	56.00±2.70	76.70±1.50
110	73.80±1.90	59.50±2.20	66.20±2.50	55.20±3.80	76.30±2.10
125	72.60±2.20	55.30±2.00	66.40±2.40	55.20±3.90	76.80±2.30
128	72.20±2.00	55.40±2.50	65.50±1.90	55.60±3.50	76.30±3.30

Table 6.10: Weighted Mean of F1 Score for feature Selection with Maximum Time-Frequency Ridge Extraction and with ReliefF (Training Dataset).

# Features	SVM	k-NN	Decision Tree	Fisher-LDA	Tree Ensemble
5	35.50±7.20	42.20±2.60	50.40±2.60	40.70±3.80	50.50±2.40
20	67.20±2.00	66.00±2.10	67.20±2.50	49.00±1.70	76.50±3.00
35	70.30±2.60	64.50±1.40	66.80±1.70	50.20±3.90	76.80±1.70
50	71.30±2.60	65.30±2.90	67.50±3.00	51.20±2.50	76.50±2.60
65	72.10±1.80	64.80±2.50	67.40±2.60	50.90±2.00	77.10±1.80
80	73.00±2.40	64.80±1.80	67.10±2.90	52.70±3.80	77.50±1.70
95	74.00±1.80	64.30±2.70	66.90±2.20	53.80±4.50	76.60±1.90
110	73.70±2.40	60.00±2.60	66.40±2.30	54.90±4.00	76.80±2.00
125	72.90±3.00	55.60±3.00	65.40±2.20	56.30±3.10	76.20±1.70
128	73.20±2.50	54.60±1.50	65.50±2.50	56.00±2.40	76.40±1.40

The subsets of features obtained are the result of the most commonly selected features across all folds. Then, after the cross-validation process, the n features with higher "ranking" across all iterations are selected. Since the features used can be grouped in distinct groups according to their domain, in Table 6.11 and Table 6.12 the relative frequency (0-1 range) of the features selected from each of the groups considered are presented. Note that in these tables the contour based features were separated from the time-frequency domain in order to assess their importance. Thus, four different groups of features were considered, where: *CB* stands for the contour based, *TF* for time-frequency domain, *F* for frequency domain and *T* for time domain features, accordingly.

Table 6.11: Kruskal-Wallis selected features relative frequency by domain.

#Feat	**CB**	**TF**	**F**	**T**
5	0.23	0	0.02	0.11
20	0.62	0.13	0.07	0.11
35	0.62	0.25	0.22	0.22
50	0.77	0.36	0.33	0.33
65	0.77	0.56	0.38	0.44
80	0.77	0.77	0.42	0.44
95	0.85	0.93	0.49	0.56
110	1	0.98	0.69	0.67
125	1	1	0.96	0.89
128	1	1	1	1

Table 6.12: ReliefF selected features relative frequency by domain.

#Feat	**CB**	**TF**	**F**	**T**
5	0.38	0	0	0
20	0.08	0	0.29	0.67
35	0	0	0.64	0.67
50	0	0.15	0.76	0.78
65	0.23	0.36	0.73	0.78
80	1	0.82	0.33	0.22
95	1	0.62	0.8	0.89
110	1	0.93	0.76	0.67
125	1	1	0.96	0.89
128	1	1	1	1

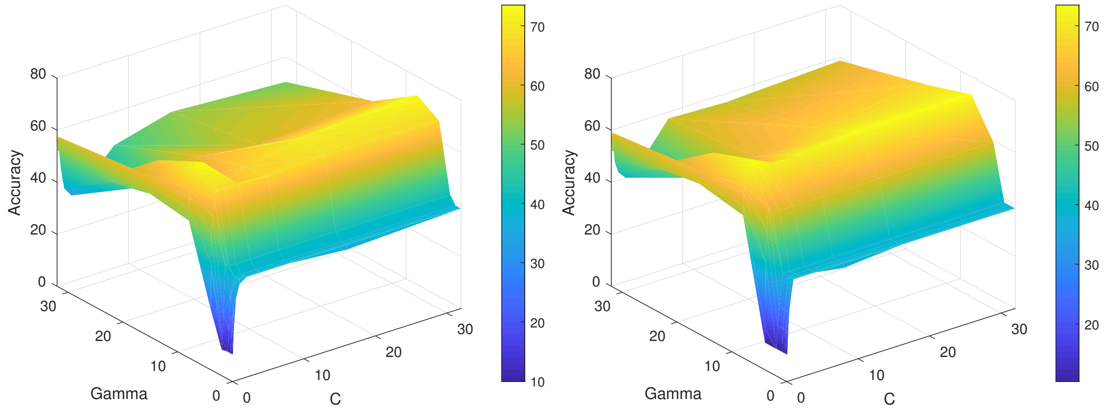
Note that the features selection methods used, namely Kruskal-Wallis and ReliefF, are independent of the classifiers, thus it was expected for the features selected with a certain n for each classifiers to be the same. However, in this does not happen since for all the classifiers the split of the train dataset is different, which led to slight changes in the groups of features depending on the classifier. Yet, these change are not significant whereby in Table 6.11 and Table 6.12 the values are not presented by classifier. In the tables the value 1 means that all features from that domain were chosen, while the value 0 means that none was.

Similarly to what happen with the feature selection methods, a feature transformation method (Principal Component Analysis (PCA)) was also tested in the training data, testing the performance obtained with all the classifiers for different values of retained variance percentage. The performance results obtained with this process are represented in Table 6.13.

Some of the classification models used have parameters that need to be adjusted, which is the case with the Support Vector Machine (SVM) and k-NN. Thus, a grid search optimization process was conducted for these two models. For the SVM, the cost value, C, and the gamma value, γ, were varied, as well as the kernel function. For the k-NN, the number of neighbors and the distance metric were varied.

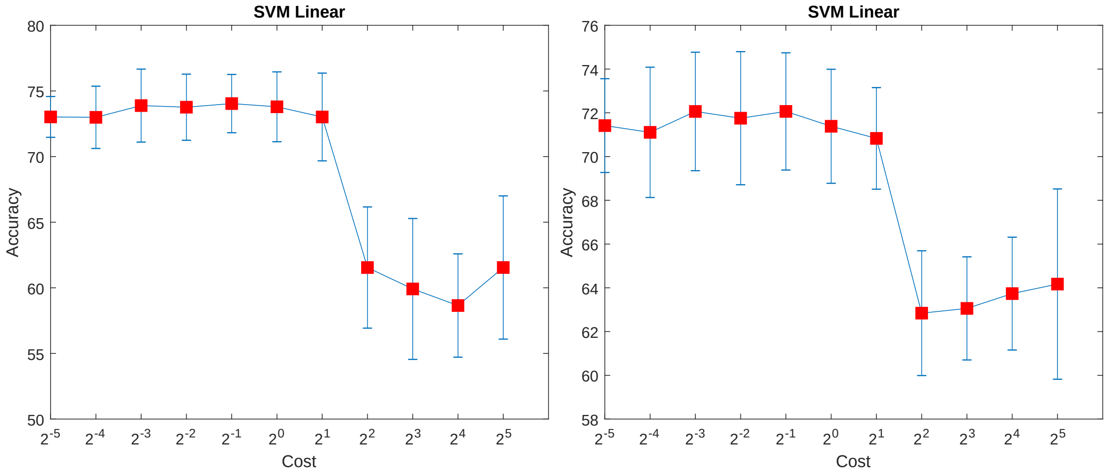
Table 6.13: Weighted Mean of F1 Score for multiple % of variance retained with Maximum Time-Frequency Ridge Extraction and with PCA (Training dataset).

% Variance Retained	SVM	k-NN	Decision Tree	Fisher-LDA	Tree Ensemble
50	14.5 ± 4.6	43.6 ±2.2	39.5±2.7	29.9±2	48.4 ± 2.3
75	23.5±4.9	51.9±1.8	48.5±2.8	38.8±3.5	51.7±2.6
85	33.2±4.8	57.8±1.7	52.2±3	43.5±3	60±2.8
90	34.5±6.9	58±2	53.2±2.5	43.5±2.3	60.6±2.4
95	38.5±5.9	56.7±2.3	52±2	43.4±2.6	60.5±2
99	49.3±5.5	61.2±1.9	55.9±2.8	48.1±1.7	64.2±1.8



(a) 110 features Kruskal-Wallis. **(b)** 95 features ReliefF.

Figure 6.3: SVM RBF grid search.

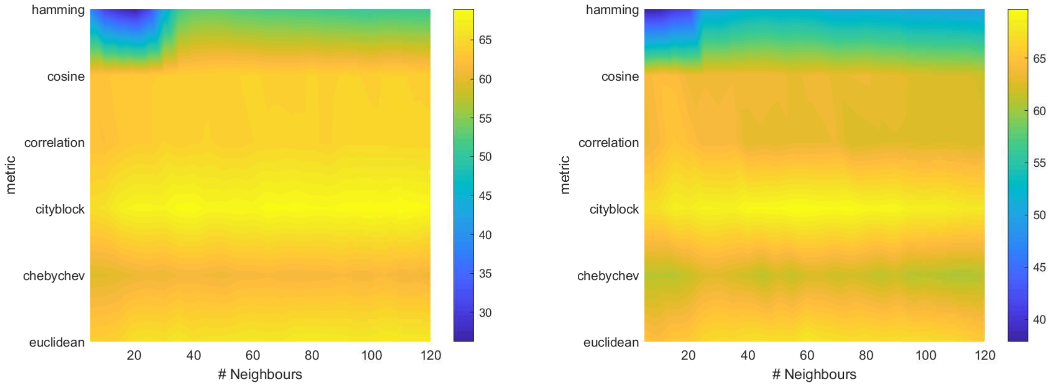


(a) 110 features Kruskal-Wallis. **(b)** 95 features ReliefF.

Figure 6.4: SVM linear grid search.

Table 6.14: Best parameters from grid-search (SVM).

Classifier	Method	Kernel	Cost	Gamma	Mean Acc.
SVM	ReliefF	RBF	2^1	2^3	73.45
SVM	Kruskal-Wallis	RBF	2^1	2^3	73.48
SVM	ReliefF	Linear	2^{-3}	n.a.	72.06
SVM	Kruskal-Wallis	Linear	2^{-1}	n.a.	74.04



(a) 20 features Kruskal-Wallis.

(b) 20 features ReliefF.

Figure 6.5: k-NN grid search.

Table 6.15: Best parameters from grid-search (k-NN).

Classifier	Selection method	# Neigb.	Dist. Metric	Mean Acc.
k-NN	ReliefF	55	City-block	69.69
k-NN	Kruskal-Wallis	110	City-block	68.95

After the determination of the subsets of features for all classifiers, and the hyper-parameters in the SVM and k-NN cases, that allowed to obtain the best performance in the training dataset, the final models were trained with all the training dataset and evaluated in the testing dataset. The results obtained are presented in Table 6.16, being all the metrics presented obtained as the weighted mean of each metric for each class.

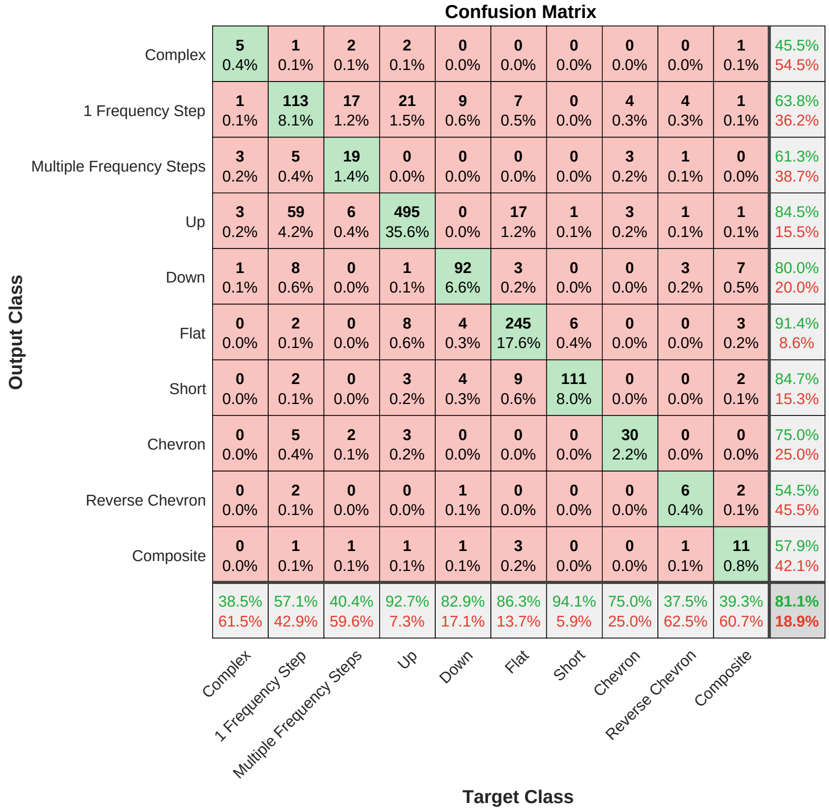
A more complete performance evaluation regarding the two best classifiers in Table 6.16 is presented bellow, where the confusion matrix was determined as well as a table with the Recall, Precision, Specificity and F1 Score values for all the classes separately. The two best methods were the Tree Ensemble and the RFB SVM, with 80 and 95 features selected with ReliefF, accordingly.

Table 6.16: Weighted mean performance values for all classifiers obtained in the test dataset (Underlined values correspond to the two best classifiers).

Classifier	**Parameters**	**Feature selection/ transformation**	**# Features/ % Variance**	**Precision**	**Recall**	**Specificity**	**F1 Score**	**Accuracy**
SVM Linear	$C = 2^{-1}$ One-vs-One	Kruskal	110	76.11	75.16	92.69	75.25	75.2
SVM Linear	$C = 2^{-1}$ One-vs-One	ReliefF	95	75.66	74.37	92.87	74.53	74.4
SVM RBF	$C = 2^1$ $\gamma = 2^3$ One-vs-One	Kruskal	110	74.20	74.30	91.80	74.05	74.3
SVM RBF	$C = 2^1$ $\gamma = 2^3$ **One-vs-One**	**ReliefF**	**95**	**75.27**	**75.95**	**91.79**	**75.38**	**76.0**
SVM	One-vs-One	PCA	99%	60.46	49.46	87.59	50.93	49.5
k-NN	$k = 110$ $metric = cityblock$	Kruskal	20	70.33	70.19	89.16	67.23	70.2
k-NN	$k = 55$ $metric = cityblock$	ReliefF	20	74.18	72.64	91.04	71.12	72.6
k-NN	-	PCA	99%	64.16	58.09	74.47	52.56	58.1
Decision Tree	-	Kruskal	128	70.14	68.40	91.50	68.95	68.4
Decision Tree	-	ReliefF	50	70.11	68.52	91.210	68.98	68.3
Decision Tree	-	PCA	99%	59.43	54.07	86.39	56.24	54.1
Fisher-LDA	Pseudo-Linear Discriminant	Kruskal	95	57.64	49.17	85.00	49.83	49.2
Fisher-LDA	Pseudo-Linear Discriminant	ReliefF	125	60.53	52.12	86.45	53.24	52.1
Fisher-LDA	Pseudo-Linear Discriminant	PCA	99%	56.39	47.95	84.41	48.25	47.9
Tree Ensemble	Num. Trees-250 GentleBoost	Kruskal	125	79.33	80.20	93.00	79.50	80.2
Tree Ensemble	**Num. Trees-250** **GentleBoost**	**ReliefF**	**80**	**80.31**	**81.14**	**93.37**	**80.46**	**81.1**
Tree Ensemble	Num. Trees-250 GentleBoost	PCA	99%	66.45	67.5	87.35	66.72	67.5

Table 6.17: Tree Ensemble Classifier with 80 features selected with ReliefF, 250 trees and GentleBoost method (Test dataset).

Class	**Precision**	**Recall**	**Specificity**	**F1 Score**
Complex	45.5	38.5	99.5	41.7
1 Frequency Step	63.8	57.1	94.1	60.3
Multiple Frequency Steps	61.3	40.4	98.9	48.7
Up	84.5	92.7	87.4	88.4
Down	70.0	82.9	97.8	81.4
Flat	91.4	86.3	97.5	88.8
Short	84.7	94.1	98.1	89.2
Chevron	75.0	75.0	99.1	75.0
Reverse Chevron	54.6	37.5	99.6	44.4
Composite	57.9	39.3	99.3	46.8
Mean	69.9	64.4	97.1	66.5
Weighted mean	80.3	81.1	93.4	80.46



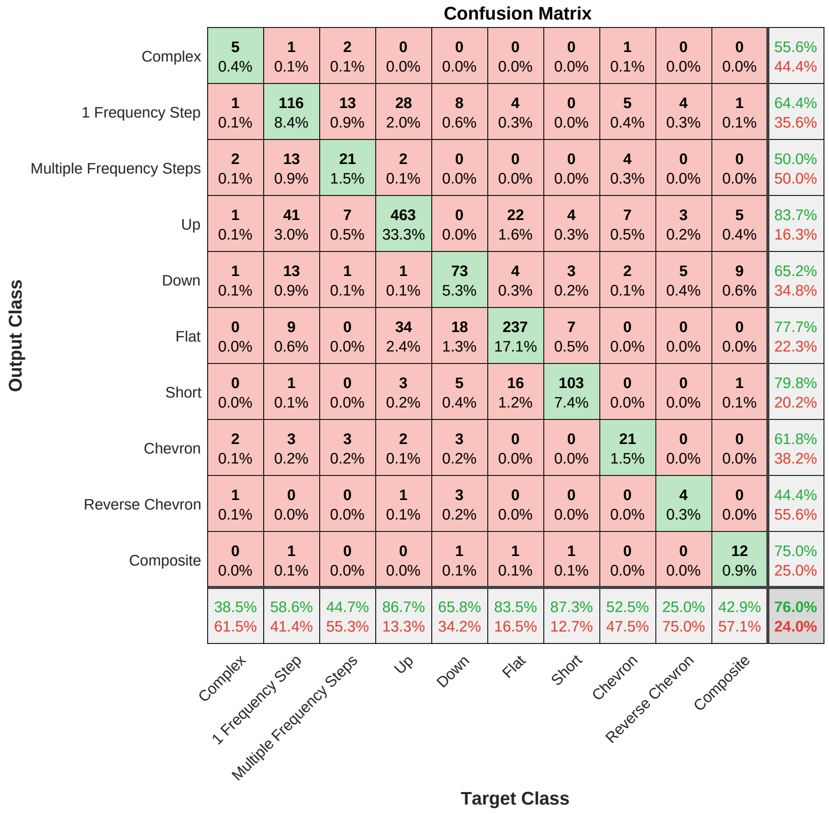
Confusion Matrix

Output Class \ Target Class	Complex	1 Frequency Step	Multiple Frequency Steps	Up	Down	Flat	Short	Chevron	Reverse Chevron	Composite	
Complex	**5** 0.4%	**1** 0.1%	**2** 0.1%	**2** 0.1%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**1** 0.1%	45.5% 54.5%
1 Frequency Step	**1** 0.1%	**113** 8.1%	**17** 1.2%	**21** 1.5%	**9** 0.6%	**7** 0.5%	**0** 0.0%	**4** 0.3%	**4** 0.3%	**1** 0.1%	63.8% 36.2%
Multiple Frequency Steps	**3** 0.2%	**5** 0.4%	**19** 1.4%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**3** 0.2%	**1** 0.1%	**0** 0.0%	61.3% 38.7%
Up	**3** 0.2%	**59** 4.2%	**6** 0.4%	**495** 35.6%	**0** 0.0%	**17** 1.2%	**1** 0.1%	**3** 0.2%	**1** 0.1%	**1** 0.1%	84.5% 15.5%
Down	**1** 0.1%	**8** 0.6%	**0** 0.0%	**1** 0.1%	**92** 6.6%	**3** 0.2%	**0** 0.0%	**0** 0.0%	**3** 0.2%	**7** 0.5%	80.0% 20.0%
Flat	**0** 0.0%	**2** 0.1%	**0** 0.0%	**8** 0.6%	**4** 0.3%	**245** 17.6%	**6** 0.4%	**0** 0.0%	**0** 0.0%	**3** 0.2%	91.4% 8.6%
Short	**0** 0.0%	**2** 0.1%	**0** 0.0%	**3** 0.2%	**4** 0.3%	**9** 0.6%	**111** 8.0%	**0** 0.0%	**0** 0.0%	**2** 0.1%	84.7% 15.3%
Chevron	**0** 0.0%	**5** 0.4%	**2** 0.1%	**3** 0.2%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**30** 2.2%	**0** 0.0%	**0** 0.0%	75.0% 25.0%
Reverse Chevron	**0** 0.0%	**2** 0.1%	**0** 0.0%	**0** 0.0%	**1** 0.1%	**0** 0.0%	**0** 0.0%	**0** 0.0%	**6** 0.4%	**2** 0.1%	54.5% 45.5%
Composite	**0** 0.0%	**1** 0.1%	**1** 0.1%	**1** 0.1%	**1** 0.1%	**3** 0.2%	**0** 0.0%	**0** 0.0%	**1** 0.1%	**11** 0.8%	57.9% 42.1%
	38.5% 61.5%	57.1% 42.9%	40.4% 59.6%	92.7% 7.3%	82.9% 17.1%	86.3% 13.7%	94.1% 5.9%	75.0% 25.0%	37.5% 62.5%	39.3% 60.7%	**81.1%** **18.9%**

Figure 6.6: Confusion Matrix Tree Ensemble Classifier with 80 features selected with ReliefF, 250 trees and GentleBoost method (Test dataset).

Table 6.18: RBF SVM with 95 features selected with ReliefF, $C = 2^1$ and $\gamma = 2^3$ (Test dataset).

Class	Precision	Recall	Specificity	F1 Score
Complex	55.6	38.5	99.6	45.5
1 Frequency Step	64.4	58.6	93.6	61.4
Multiple Frequency Steps	50.0	44.7	98.0	47.2
Up	83.7	86.7	86.8	85.2
Down	65.2	65.8	96.2	65.5
Flat	77.7	83.5	92.3	80.5
Short	79.8	87.3	97.3	83.4
Chevron	61.8	52.5	98.8	56.8
Reverse Chevron	44.4	25.0	99.5	32.0
Composite	75.0	42.9	99.6	54.6
Mean	65.8	58.5	96.2	61.2
Weighted mean	75.3	76.0	91.8	75.4



	Complex	1 Frequency Step	Multiple Frequency Steps	Up	Down	Flat	Short	Chevron	Reverse Chevron	Composite	
Complex	5 0.4%	1 0.1%	2 0.1%	0 0.0%	0 0.0%	0 0.0%	0 0.0%	1 0.1%	0 0.0%	0 0.0%	55.6% 44.4%
1 Frequency Step	1 0.1%	116 8.4%	13 0.9%	28 2.0%	8 0.6%	4 0.3%	0 0.0%	5 0.4%	4 0.3%	1 0.1%	64.4% 35.6%
Multiple Frequency Steps	2 0.1%	13 0.9%	21 1.5%	2 0.1%	0 0.0%	0 0.0%	0 0.0%	4 0.3%	0 0.0%	0 0.0%	50.0% 50.0%
Up	1 0.1%	41 3.0%	7 0.5%	463 33.3%	0 0.0%	22 1.6%	4 0.3%	7 0.5%	3 0.2%	5 0.4%	83.7% 16.3%
Down	1 0.1%	13 0.9%	1 0.1%	1 0.1%	73 5.3%	4 0.3%	3 0.2%	2 0.1%	5 0.4%	9 0.6%	65.2% 34.8%
Flat	0 0.0%	9 0.6%	0 0.0%	34 2.4%	18 1.3%	237 17.1%	7 0.5%	0 0.0%	0 0.0%	0 0.0%	77.7% 22.3%
Short	0 0.0%	1 0.1%	0 0.0%	3 0.2%	5 0.4%	16 1.2%	103 7.4%	0 0.0%	0 0.0%	1 0.1%	79.8% 20.2%
Chevron	2 0.1%	3 0.2%	3 0.2%	2 0.1%	3 0.2%	0 0.0%	0 0.0%	21 1.5%	0 0.0%	0 0.0%	61.8% 38.2%
Reverse Chevron	1 0.1%	0 0.0%	0 0.0%	1 0.1%	3 0.2%	0 0.0%	0 0.0%	0 0.0%	4 0.3%	0 0.0%	44.4% 55.6%
Composite	0 0.0%	1 0.1%	0 0.0%	0 0.0%	1 0.1%	1 0.1%	1 0.1%	0 0.0%	0 0.0%	12 0.9%	75.0% 25.0%
	38.5% 61.5%	58.6% 41.4%	44.7% 55.3%	86.7% 13.3%	65.8% 34.2%	83.5% 16.5%	87.3% 12.7%	52.5% 47.5%	25.0% 75.0%	42.9% 57.1%	**76.0%** **24.0%**

Figure 6.7: Confusion Matrix RBF SVM with 95 features selected with ReliefF, $C = 2^1$ and $\gamma = 2^3$ (Test dataset).

In Table 6.19, the features that were used to train the two best models in Table 6.17 and Table 6.18 are presented. Furthermore, a graphical representation of each domain selection feature percentage is also presented in Figure 6.8 and Figure 6.9.

Table 6.19: Features selected for the training of the two best classifiers.

Tree Ensemble (80 Feat ReliefF)	Trend, Number Dir Changes, Number Steps, Harmonic Components, Diff Max End, Diff Min Init, Initial Frequency, Duration, Frequency Bandwith, LBP37, Final Frequency, Spectral Edge Frequency, Freq max PSD, Peak Frequency, LBP41, Mean Freq PSD, Diff Min End, Diff Max Init, Minimum Frequency, LBP35, LBP39, Diff End Init, LBP3, LBP7, Mean Frequency, LBP9, Mean Spectral Skewness, Non instantanous Spectral Entropy, LBP36, LBP5, Mean Tona Power Ratio, Mean Spectral Slope, LBP51, LBP8, LBP15, Std Spectral Entropy, LBP55, LBP4, LBP27, LBP30, Mean Spectral Centroid, LBP12, LBP57, LBP31, LBP26, LBP53, LBP16, Max power PSD, LBP11, Mean Spectral Spread, LBP14, LBP29, Mean ZCR, LBP33, LBP22, LBP32, LBP28, LBP46, LBP1, Min Spectral Entropy, LBP24, LBP59, LBP10, LBP42, LBP45, Median Spectral Entropy, LBP49, LBP20, LBP25, LBP40, LBP18, LBP23, Mean Spectral Entropy, LBP47, LBP56, LBP13, LBP21, LBP44, Signal Power, LBP54
SVM (95 Feat ReliefF)	Short Time Energy, Max Tonality, Min Tonality, Std Tonality, Mean Amplitude, Median Amplitude, Std Amplitude, Skewness Amplitude, Min power PSD, Freq min PSD, Std power PSD, Mean Spectral Flatness, Mean Spectral Flux, Mean Spectral Pitch Chroma1, Mean Spectral Pitch Chroma2, Mean Spectral Pitch Chroma3, Mean Spectral Pitch Chroma4, Mean Spectral Pitch Chroma5, Mean Spectral Pitch Chroma6, Mean Spectral Pitch Chroma7, Mean Spectral Pitch Chroma8, Mean Spectral Pitch Chroma9, Mean Spectral Pitch Chroma10, Mean Spectral Pitch Chroma11, Mean Spectral Pitch Chroma12, Mean Spectral Rolloff, Mean Spectral Decrease, Mean Time STD, LBP34, LBP38, Trend, Number Dir Changes, Number Steps, Harmonic Components, Diff Max End, Diff Min Init, Initial Frequency, Frequency Bandwith, Duration, Kurtosis Amplitude, Mean Spectral Kurtosis, LBP37, Final Frequency, Freq max PSD, Spectral Edge Frequency, Peak Frequency, LBP41, Diff Max Init, Mean Freq PSD, Diff Min End, Max Spectral Entropy, LBP39, Minimum Frequency, LBP35, Diff End Init, LBP7, LBP3, Mean Frequency, LBP9, Mean Spectral Skewness, LBP5, LBP36, Mean Tona Power Ratio, Non instantanous Spectral Entropy, LBP8, LBP15, LBP51, Std Spectral Entropy, Mean Spectral Slope, LBP55, LBP4, LBP27, LBP30, LBP12, Mean Spectral Centroid, LBP57, LBP16, LBP53, LBP31, LBP26, LBP11, Mean Spectral Spread, Max power PSD, LBP29, LBP14, LBP33, LBP32, LBP28, Mean ZCR, LBP22, LBP46, LBP1, LBP42, LBP59, LBP24

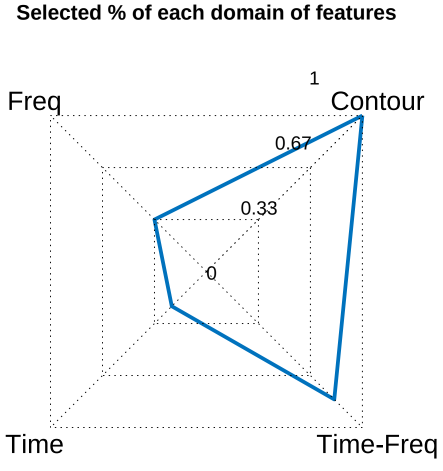


Figure 6.8: Domain percentage selection of the 80 features selected with ReliefF (Tree Ensemble Classifier).

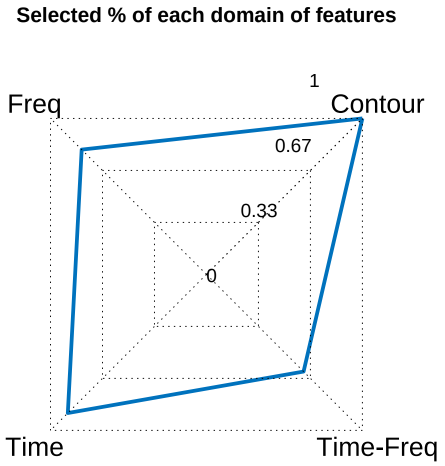


Figure 6.9: Domain percentage selection of the 95 features selected with ReliefF (SVM).

6.2.2 Classification Discussion

The classification results are divided in two different parts. Firstly, an evaluation of the performance of the multiple algorithms to extract the time-frequency ridges of the vocalizations was made. After that, the best method was selected and used to develop the classifications algorithms themselves and the according results in the test set were determined.

Given the fact that several contour extraction algorithm were developed, a statistical analysis was conducted to assess which of the methods performed the best with the multiple classifiers used. As presented in subsection 6.2.1, not all contour extraction methods performed equally with the multiple classifiers, whereby, the two strongest performing methods across all classifiers were selected and their performance distributional differences was analyzed (see Table 6.7). Through the analysis of Table 6.7 is possible to observe that the only contour extraction method that was consistently among the two better ones, was the Maximum Time-Frequency Ridge Extraction. Thus, it was the one used to extract the descriptors set to be used, joined with the remaining ones described in section 5.2.

As it is possible to see in Table 6.9 and Table 6.10, the importance of feature selection is demonstrated, however, the performance improvements are not that pronounced. This is generally truth for all classifiers, which suggests that the classification problems are originating from the features themselves, and not from the way in which they are selected or from the classification methods. Furthermore, through the analysis of Table 6.11 it is possible to observe that Time-frequency domain features, which includes the contour based ones, tend to have a higher percentage of selection. In Table 6.12, some irregular behaviors regarding the selected descriptors are patented. Also, time domain descriptors are all select for almost every number of selected features. Contour based features also tend to be all select with the ReliefF method, which demonstrate the importance that those features since they are the ones used in te manual annotation process. Using a different method to reduce the problem dimensionality, representing the data where it varies the most (PCA), the conclusions drawn are similar to the ones with the feature selection techniques.

Moving to the performance analysis of the models developed with the best parameters defined with the help of the training dataset, a clear trend emerges. Looking at the results in the test dataset (see Table 6.16), it is noticeable that the performance does not decrease that much when compared to the results obtained in the models validation and adjustment (see Table 6.9, Table 6.10 and Table 6.13). In some cases, the performance obtained in the test phase has even supplanted the performance in the models validation phase. This fact stresses out three important aspects. First, it highlights the quality of the used features. Secondly, it means that the selection of the better parameters and number of features for the classification was made in such a way that eventual overfitting phenomena may have been avoided. Thirdly, it means that in this particular problem, the classifiers benefits from the use of more data, since in the validation phase they were typically trained with 9/10 of the total data composing the training dataset while in the test phase the whole training set was used.

Another conclusion that can be drawn is related to the classifiers which have showed better results, being the SVM and the Tree Ensembles the best methods. Thus, in order to better understand how these two models behave, the confusion matrix and the according Recall, Precision, Specificity and F1 Score values for each class, were represented in Figure 6.6, Figure 6.7, Table 6.17 and Table 6.18.

Analyzing the Tree Ensemble method, Figure 6.6 and Table 6.17, is possible to observe that the method can discriminate the classes Upward, Downward, Flat and Short quiet well. This can be explained by the fact that these classes have strict definitions and lots of samples, from which the classifier can learn. On the other hand, the classes that were the hardest for the classifier to correctly identify were the ones with less samples, Complex, Reverse Chevron and Composite. Observing the confusion matrix, it is clear that the class that was confused more often was the 1-Frequency Step, being the more expressive confusion with the Upward class. Even-tough the 1-Frequency Step class has a very strict and unequivocal identifier, which is the number of frequency breaks or “steps”, more precisely one step, this feature is pretty hard to extract correctly, namely due to the quality of some vocalizations themselves and the excessive noise. This fault in the extraction of the descriptor gives origin to several missclassification cases. The same thing happens with the class 2-Frequency Steps. Still, for a 10 class multi-class problem with very uneven class distribution, the performance obtain with the Tree Ensemble method is quiet satisfactory, being the weighted mean of the Recall, Precision and F1 Score values all around 80%, as well as the global accuracy.

Looking now to the performance of the RBF SVM, Figure 6.7 and Table 6.18, is also perceptible that the classes in which the classifier struggled the most where the ones with less representation. Also, in a similar way, the most mismatched class was the 1-Frequency Step, which was mostly confused with the Upward class. The SVM presented a slightly lower performance values than the Tree Ensemble, however, this should already be expected since the way the USV classes are defined and manually annotated resembles a top down process with several conditions "if ... then ... else ...", which, basically, works like these classification trees.

In conclusion, the best classification methods presented good results in all the evaluated metrics. Plus, while the state of art classification methods presented [20, 25] can distinguish only between five different classes, the classifiers used in these work can classify ten USVs classes. Since in the state of the art literature, no performance evaluation of the classification was made, a proper comparison with the developed methods could not take course. Moreover, the classes classified on the state of the art do not have correspondent definitions with the ones classified in this work, making it even harder to perform a comparative analysis.

Limitations

Regarding the classification itself, the main reason why the results were not higher is, certainly, related to the quality of the extracted features. Since the USVs classes definitions follow such strict and well defined rules, if the descriptors were extracted in a perfect way, the classification would also be perfect. Thus, in relation to the classification problem, the main obstacle was, for sure, the features extraction, mostly due to the low quality of some vocalizations and the enormous variability within each class. The mentioned problems made it even harder to develop proper algorithms to extract the features to be used in classification, being this mostly truth to the features related to the contour of the USVs.

Looking back to all the classification related work, there are several takeaways that are worthy to be mentioned and referenced. The first is related to one of the main obstacles and difficulties in rodents USVs studies, which is the fact that a standard set of vocalizations does not exist. This means that is hard to list all types of sounds produced by rodents, mainly because these sounds can vary a lot with several factors, such as the species of the rodents, age, sex and other factors. Other of the main struggles in rodents USVs studies is, definitely, the lack of public accessible annotated data, which makes it harder to perform external validation of the developed classification algorithms. The same is true for the segmentation algorithms. However, there have been some recent developments regarding this topic, as an online platform called MouseTube[1] was created. This platform allows researchers to make public their own acquisitions of mouse USVs and describe the protocols used to acquire them. Even-tough there is no annotation regarding the vocalization of the multiple signals in the platform, the simple fact of sharing the data allows for other researchers to understand what types of protocols generate more USVs, for instance. Platforms like MouseTube are the result of an increasingly search and interest

[1] https://mousetube.pasteur.fr/

in USVs. These type of studies have been more and more common over the years, since they present a simple and easy tool to access the animal models.

Chapter 7

Conclusion

The starting global premise of this work was fairly simple and straightforward, being the main objective the development of a system capable of processing raw audio files, segmenting their Ultrasonic Vocalizations (USVs) and then, categorize them.

Having the initial objectives in mind, it can be said they were fulfilled, since a fully automated pipeline capable of processing the audio files was developed, being the final result an application developed in MATLAB. The developed software considers all algorithms that compose this pipeline.

With this work a new approach regarding USV segmentation has been developed. The segmentation algorithm has shown great results, being able to detect most of the USVs. Furthermore, the association with a post-processing technique, which in these case was the Bag of Visual Words (BOVW), has also proved to be a very interesting approach, making the segmentation process more robust.

Also, a new approach was taken regarding the sets of features used. This new approach consisted in the fusion of features information from multiple domains, including the addition of image specific descriptors.

In relation to the developed classification algorithms, the performance obtained can be acknowledged as satisfactory, mainly due to the fact that the problem was a multi-class classification one, with ten different classes. Regarding this topic, is worth notice that the freely available state of the art methods are only capable of discriminate between five different class, or group the vocalizations in classes without labeling them. Thus, significant improvements were made in the USVs classification topic.

In conclusion, with this work several advances have been made and the objectives that were set as a goal, when the project started, were reached. Even if the developed segmentation and classification algorithms are not perfect, they are arguably a step in the right direction.

Future Work

Since USVs analysis constitute a growing field of study, several things can be done in the future based on the work developed in this thesis. Perhaps, one of the most interesting approaches to be studied in the future, would be, definitely, an approach based on computer vision and deep-learning techniques. The computer vision based techniques and deep-learning have experienced a rapid grow in demand and use, showing great results in image classification problems. Hence, this path is, undoubtedly, worthy to be explored. As demonstrated by Coffey et al. 2019 [20], deep-learning techniques have great potential in USVs studies.

Also, as future work, a more exhaustive statistical analysis should be conducted, in order to unclear the relations between vocalizations and emotions, as well as the relation with the neurodevelopmental state of the animals, among other factors, in order to uncover the full potential of USVs as a biomarker.

Appendix A

Graphical Interface

Using the platform App Designer from MATLAB 2018b, a graphical interface was developed, as a final product of the project. The interface allows the complete processing of raw audio signals, since it integrates the developed algorithms, starting from the segmentation of the Ultrasonic Vocalizations (USVs) to their according classification.

The software presents a simple graphical design, being divided in three main menus or tabs. Following the typical chronological order in bioacoustic studies, the first menu is related to the segmentation of raw audio files. Then, the second tab offers the ability to rectify or accept the output of the segmentation algorithm. At last, using the segmentation output, in the classification menu it is possible to run the segmented calls to through the developed classification systems.

To use the interface a version of MATLAB has to installed. The MATLAB distribution must be the 2018b version.

Segmentation Menu

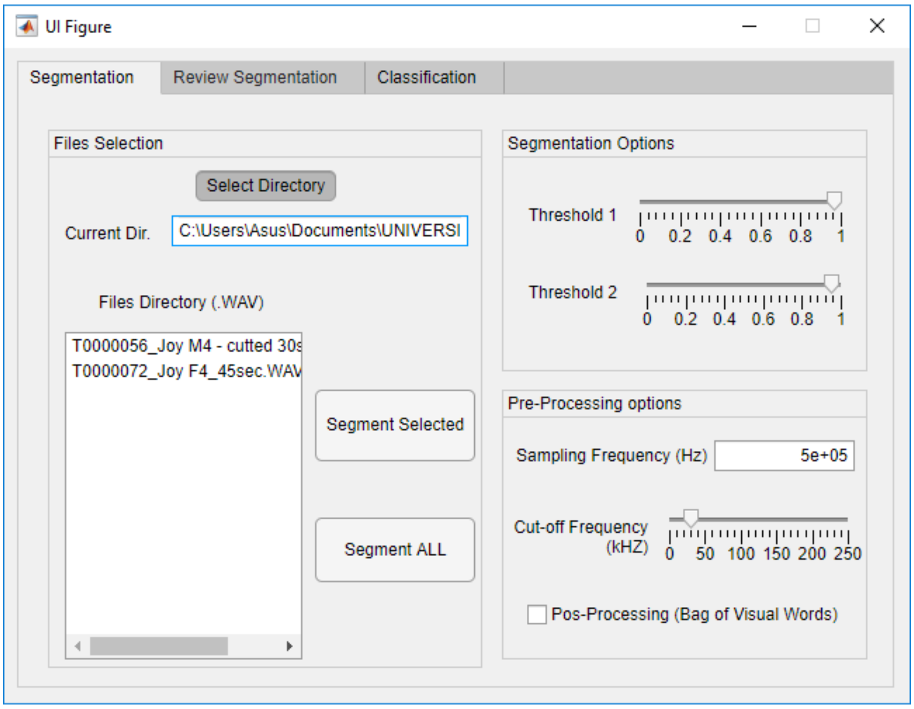
The segmentation menu has two core functionalities, which are the isolation of the USVs in the audio signals and the extraction of some relevant characteristics regarding them.

This tab has a simple layout and to perform the segmentation of audio signals it is only required to select a directory where the raw audio files are located, and then, select which ones to choose. Furthermore, the user also has the possibility to change some parameters related to the segmentation itself, such as the values of each of the thresholds used, the filtering cut of frequency and the activation or not of the Bag of Visual Words (BOVW) noise classifier.

The interface outputs a *.xls* (see Figure A.2) file containing the date when the signal was segmented concatenated with its name. The *.xls* file lists all the detected vocalizations with the extracted characteristics, which are:

1. Duration;

2. Peak Frequency;
3. Minimum Frequency;
4. Mean Frequency;
5. Initial Frequency;
6. Final Frequency;
7. Frequency Bandwidth;
8. Short Time Energy;
9. Energy.



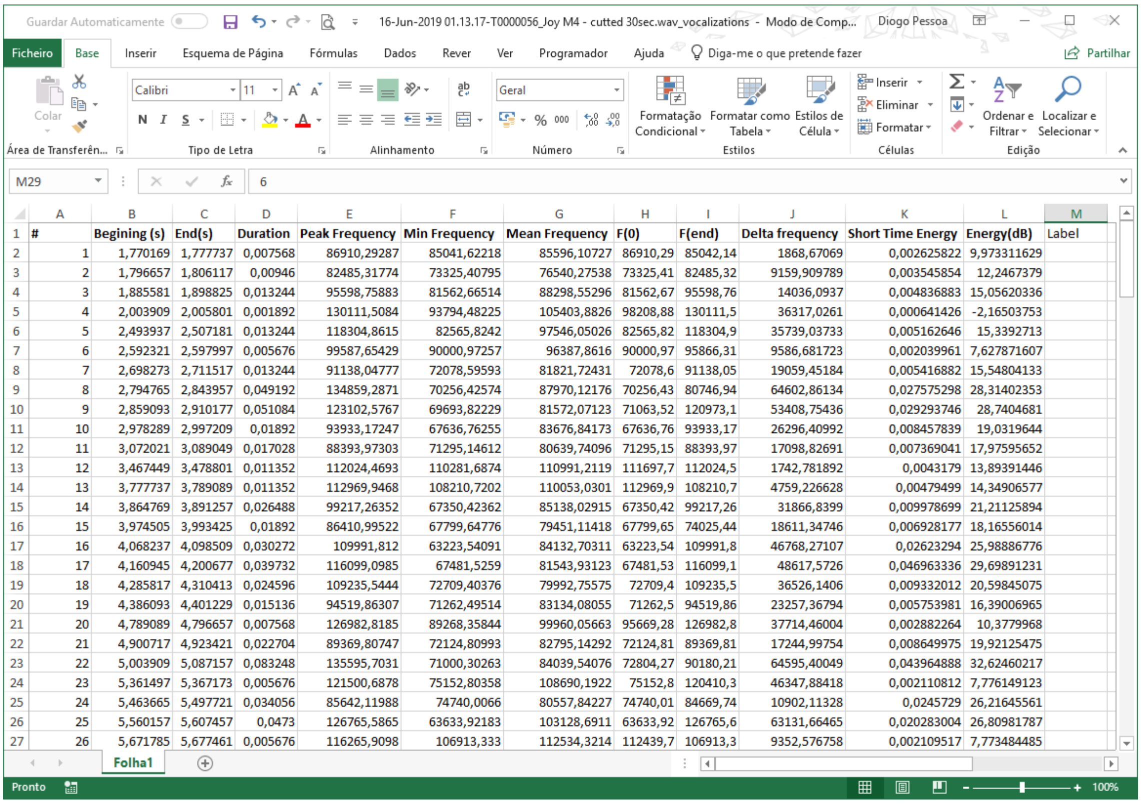
Current Dir. C:\Users\Asus\Documents\UNIVERSI

T0000056_Joy M4 - cutted 30s
T0000072_Joy F4_45sec.WAV

Sampling Frequency (Hz) 5e+05

☐ Pos-Processing (Bag of Visual Words)

Figure A.1: Tab Segmentation.



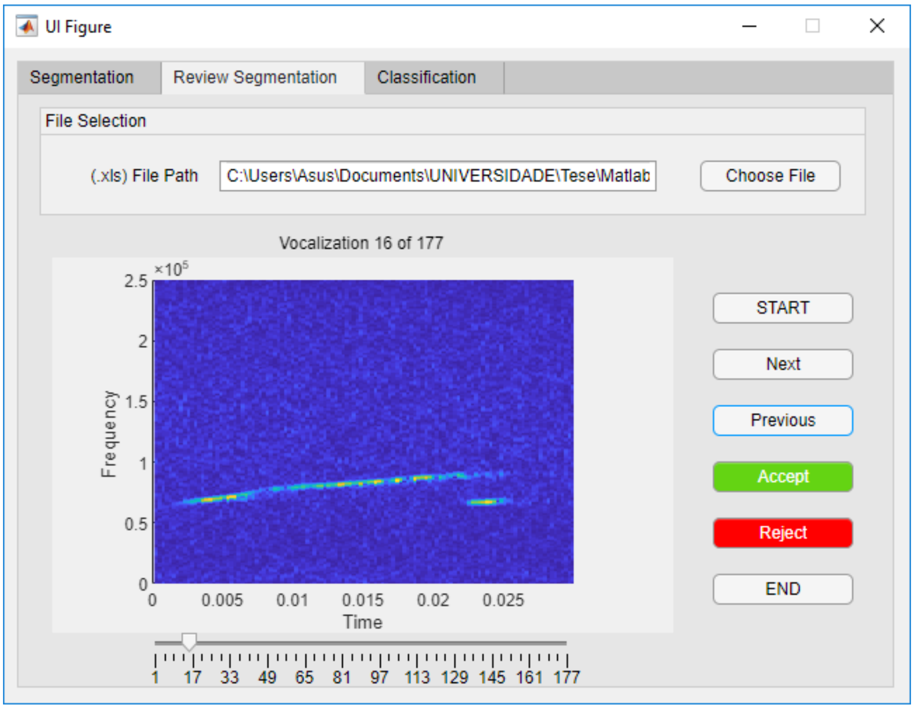
M29 | 6

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	#	Begining (s)	End(s)	Duration	Peak Frequency	Min Frequency	Mean Frequency	F(0)	F(end)	Delta frequency	Short Time Energy	Energy(dB)	Label
2	1	1,770169	1,777737	0,007568	86910,29287	85041,62218	85596,10727	86910,29	85042,14	1868,67069	0,002625822	9,973311629	
3	2	1,796657	1,806117	0,00946	82485,31774	73325,40795	76540,27538	73325,41	82485,32	9159,909789	0,003545854	12,2467379	
4	3	1,885581	1,898825	0,013244	95598,75883	81562,66514	88298,55296	81562,67	95598,76	14036,0937	0,004836883	15,05620336	
5	4	2,003909	2,005801	0,001892	130111,5084	93794,48225	105403,8826	98208,88	130111,5	36317,0261	0,000641426	-2,16503753	
6	5	2,493937	2,507181	0,013244	118304,8615	82565,8242	97546,05026	82565,82	118304,9	35739,03733	0,005162646	15,3392713	
7	6	2,592321	2,597997	0,005676	99587,65429	90000,97257	96387,8616	90000,97	95866,31	9586,681723	0,002039961	7,627871607	
8	7	2,698273	2,711517	0,013244	91138,04777	72078,59593	81821,72431	72078,6	91138,05	19059,45184	0,005416882	15,54804133	
9	8	2,794765	2,843957	0,049192	134859,2871	70256,42574	87970,12176	70256,43	80746,94	64602,86134	0,027575298	28,31402353	
10	9	2,859093	2,910177	0,051084	123102,5767	69693,82229	81572,07123	71063,52	120973,1	53408,75436	0,029293746	28,7404681	
11	10	2,978289	2,997209	0,01892	93933,17247	67636,76255	83676,84173	67636,76	93933,17	26296,40992	0,008457839	19,0319644	
12	11	3,072021	3,089049	0,017028	88393,97303	71295,14612	80639,74096	71295,15	88393,97	17098,82691	0,007369041	17,97595652	
13	12	3,467449	3,478801	0,011352	112024,4693	110281,6874	110991,2119	111697,7	112024,5	1742,781892	0,0043179	13,89391446	
14	13	3,777737	3,789089	0,011352	112969,9468	108210,7202	110053,0301	112969,9	108210,7	4759,226628	0,00479499	14,34906577	
15	14	3,864769	3,891257	0,026488	99217,26352	67350,42362	85138,02915	67350,42	99217,26	31866,8399	0,009978699	21,21125894	
16	15	3,974505	3,993425	0,01892	86410,99522	67799,64776	79451,11418	67799,65	74025,44	18611,34746	0,006928177	18,16556014	
17	16	4,068237	4,098509	0,030272	109991,812	63223,54091	84132,70311	63223,54	109991,8	46768,27107	0,02623294	25,98886776	
18	17	4,160945	4,200677	0,039732	116099,0985	67481,5259	81543,93123	67481,53	116099,1	48617,5726	0,046963336	29,69891231	
19	18	4,285817	4,310413	0,024596	109235,5444	72709,40376	79992,75575	72709,4	109235,5	36526,1406	0,009332012	20,59845075	
20	19	4,386093	4,401229	0,015136	94519,86307	71262,49514	83134,08055	71262,5	94519,86	23257,36794	0,005753981	16,39006965	
21	20	4,789089	4,796657	0,007568	126982,8185	89268,35844	99960,05663	95669,28	126982,8	37714,46004	0,002882264	10,3779968	
22	21	4,900717	4,923421	0,022704	89369,80747	72124,80993	82795,14292	72124,81	89369,81	17244,99754	0,008649975	19,92125475	
23	22	5,003909	5,087157	0,083248	135595,7031	71000,30263	84039,54076	72804,27	90180,21	64595,40049	0,043964888	32,62460217	
24	23	5,361497	5,367173	0,005676	121500,6878	75152,80358	108690,1922	75152,8	120410,3	46347,88418	0,002110812	7,776149123	
25	24	5,463665	5,497721	0,034056	85642,11988	74740,0066	80557,84227	74740,01	84669,74	10902,11328	0,0245729	26,21645561	
26	25	5,560157	5,607457	0,0473	126765,5865	63633,92183	103128,6911	63633,92	126765,6	63131,66465	0,020283004	26,80981787	
27	26	5,671785	5,677461	0,005676	116265,9098	106913,333	112534,3214	112439,7	106913,3	9352,576758	0,002109517	7,773484485	

Figure A.2: Segmentation Output.

Segmentation Correction Menu

The second menu allows the user to review the segmentation output, as it offers the possibility to accept or rejects the detected calls. If the user rejects a vocalization, it is removed from the *.xls* files. Thus, this menu has a very simple layout, enabling not only the possibility of review the segmented USVs, but also to see their time-frequency representation.



(.xls) File Path C:\Users\Asus\Documents\UNIVERSIDADE\Tese\Matlab

Vocalization 16 of 177

Figure A.3: Tab Review.

Classification Menu

The classification menu allows the user to run a classifier through the output of the segmented vocalizations. In this menu, the user only has to choose a single or multiple *.xls* files, which resulted from the segmentation process, and the classifier in order to classify. The available classifiers are SVM and Tree Ensemble.

After the classification process a *.xls* file is generated containing some relevant characteristics regarding the vocalizations distribution. The output statistics are:

1. Absolute frequency of each class;

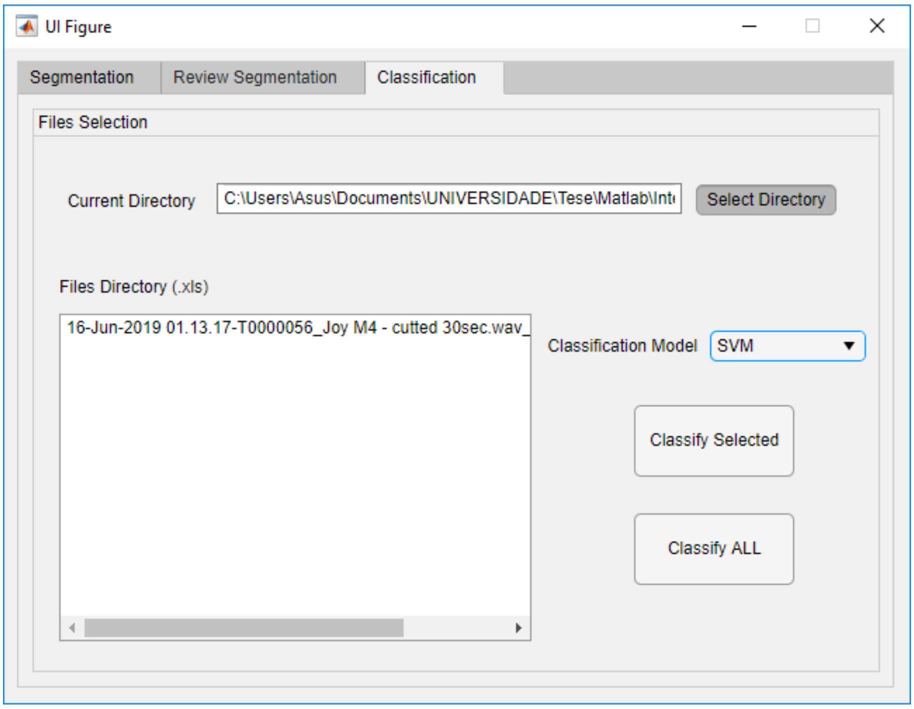
2. Rate of each class;

3. Inter vocalization time of each class;

4. Total Count of the number of vocalization;

5. Global vocalizations rate;

6. Global inter vocalizations time.



UI Figure

Files Selection

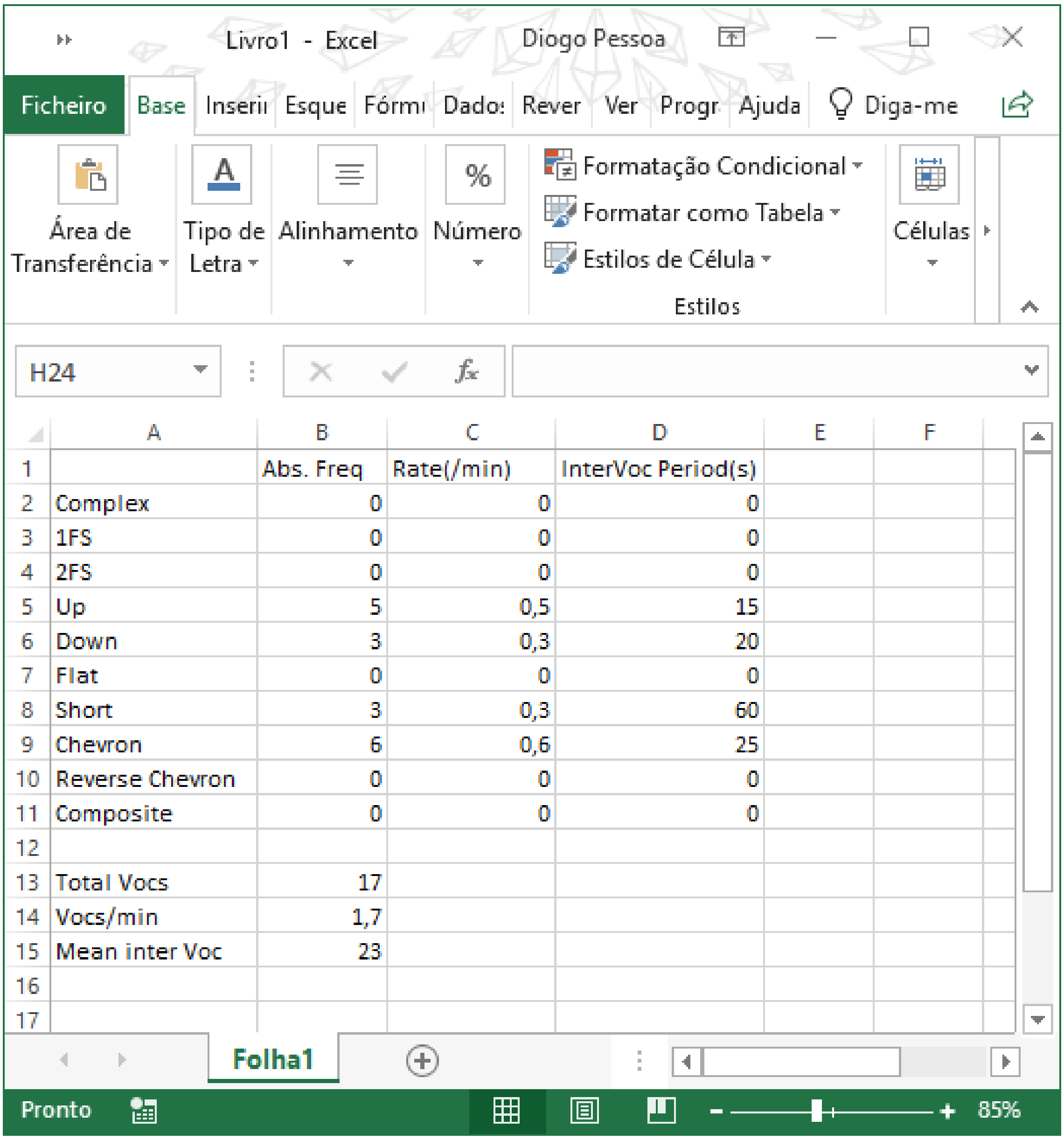
Current Directory: C:\Users\Asus\Documents\UNIVERSIDADE\Tese\Matlab\Int

Files Directory (.xls)

16-Jun-2019 01.13.17-T0000056_Joy M4 - cutted 30sec.wav_

Classification Model: SVM

Figure A.4: Tab Classification.



	A	B	C	D	E	F
1		Abs. Freq	Rate(/min)	InterVoc Period(s)		
2	Complex	0	0	0		
3	1FS	0	0	0		
4	2FS	0	0	0		
5	Up	5	0,5	15		
6	Down	3	0,3	20		
7	Flat	0	0	0		
8	Short	3	0,3	60		
9	Chevron	6	0,6	25		
10	Reverse Chevron	0	0	0		
11	Composite	0	0	0		
12						
13	Total Vocs	17				
14	Vocs/min	1,7				
15	Mean inter Voc	23				
16						
17						

Figure A.5: Classification Output.

Appendix B

Contour Extraction Evaluation

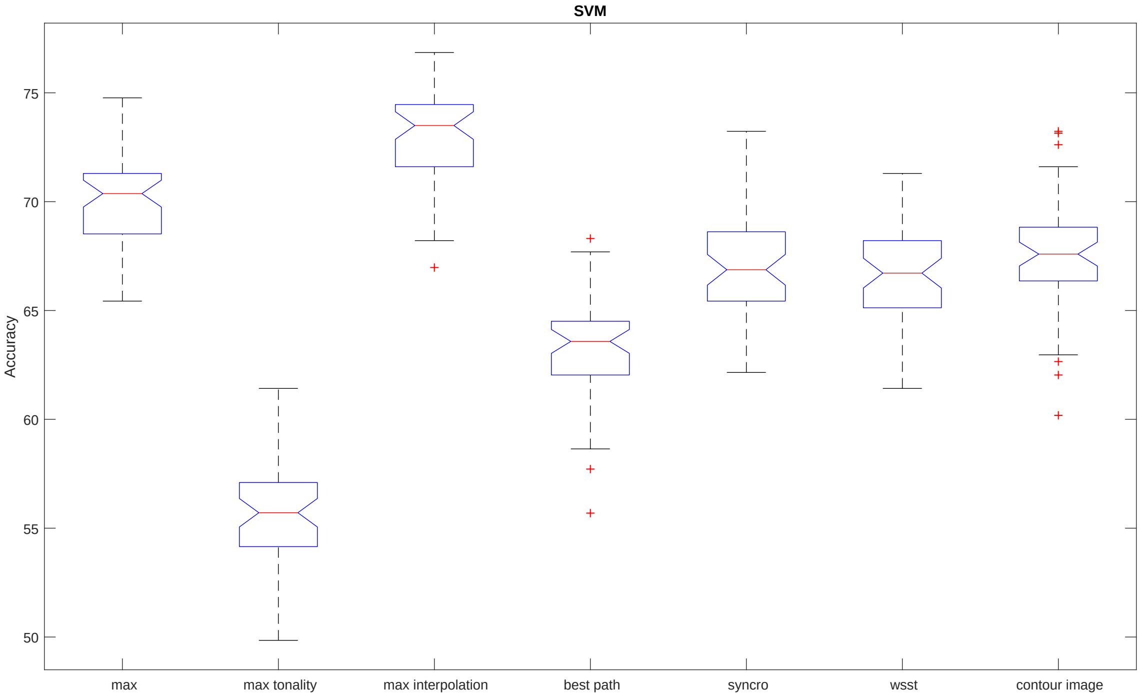


Figure B.1: Boxplot SVM.

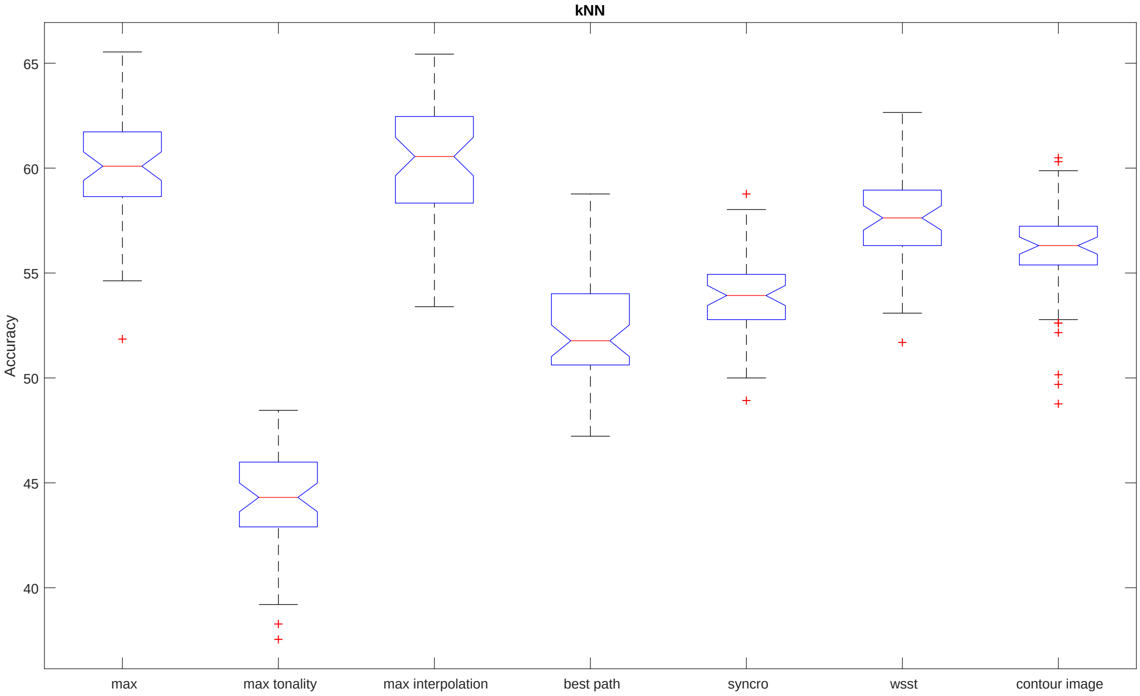


Figure B.2: Boxplot k-NN.

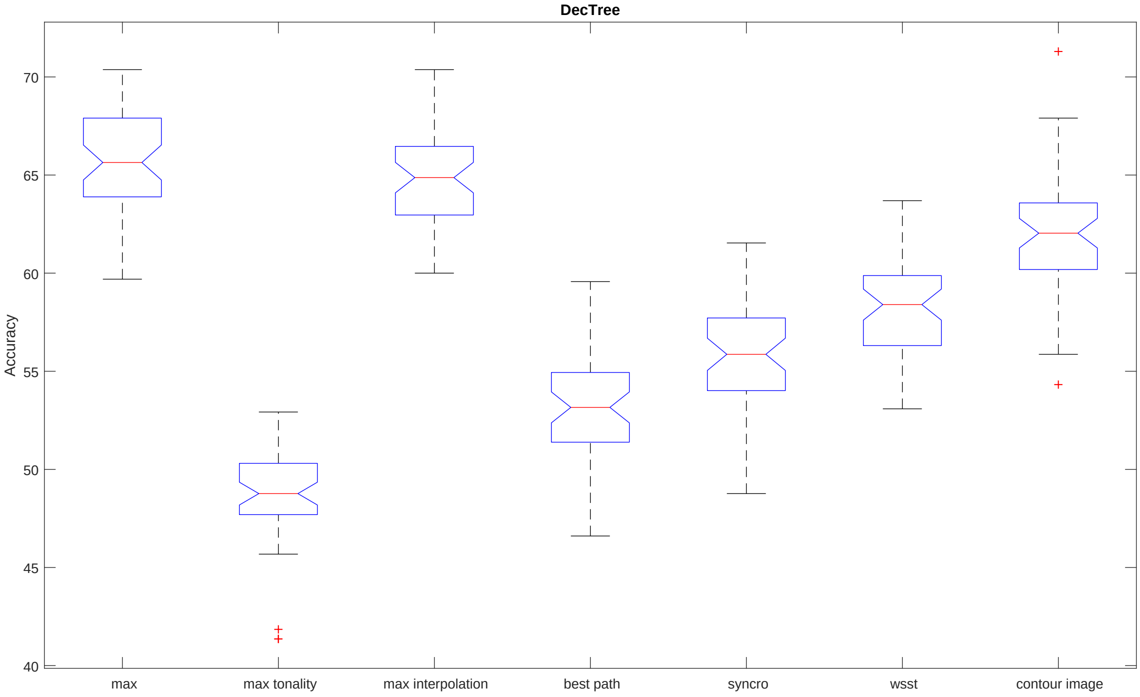


Figure B.3: Boxplot Decision Tree.

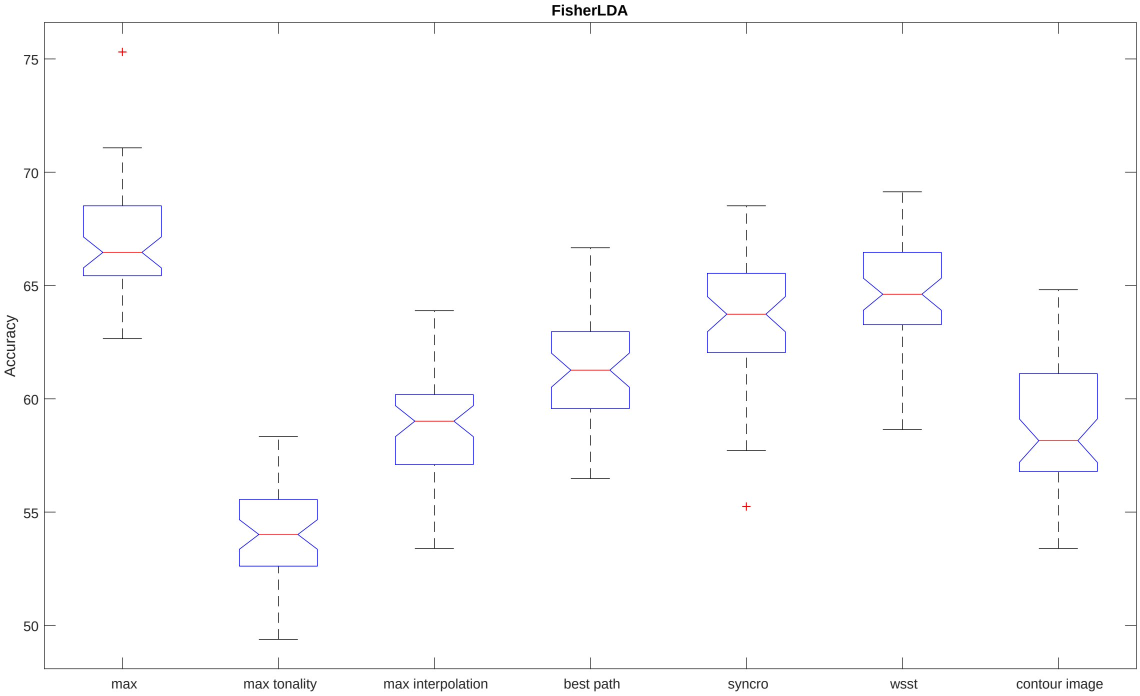


Figure B.4: Boxplot Fisher-LDA.

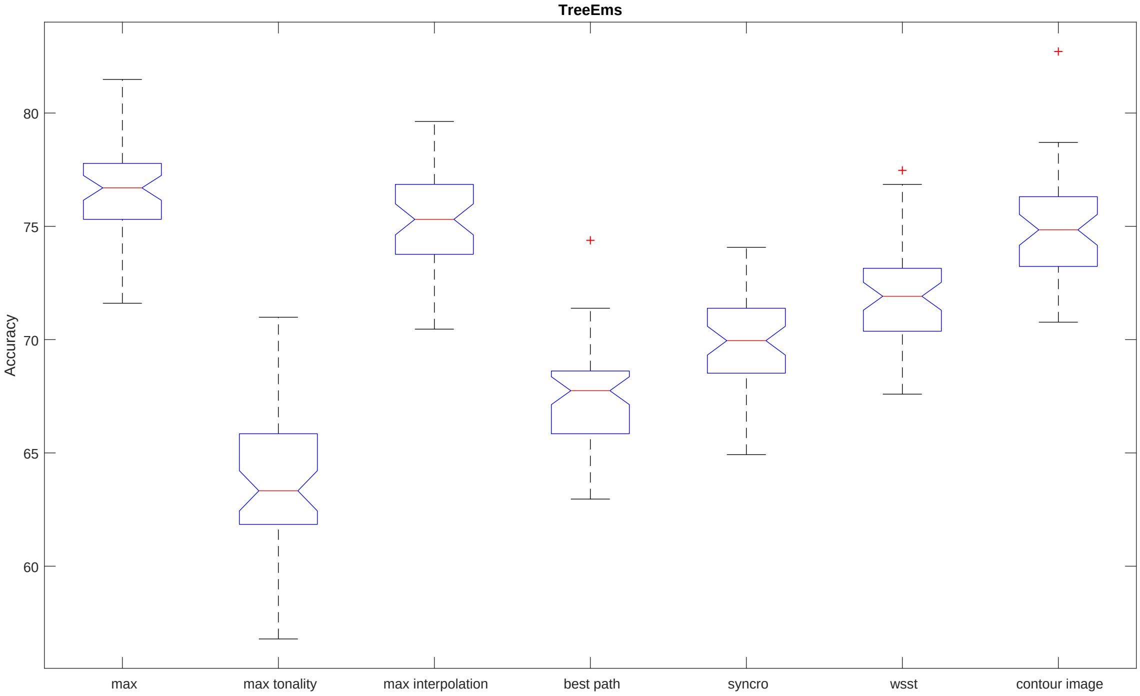


Figure B.5: Boxplot Tree Ensemble.

References

[1] M. C. Kalcounis-Rueppell, J. D. Metheny, and M. J. Vonhof, "Production of ultrasonic vocalizations by Peromyscus mice in the wild," *Front. Zool.*, vol. 3, p. 3, dec 2006. 1

[2] M. F. Bear, B. W. Connors, and M. A. Paradiso, *Neuroscience: Exploring the brain, 3rd ed.* Philadelphia, PA, US: Lippincott Williams & Wilkins Publishers, 2007. 5

[3] C. Belzung, S. Leman, P. Vourc'h, and C. Andres, "Rodent models for autism: A critical review," *Drug Discov. Today Dis. Model.*, vol. 2, pp. 93–101, jun 2005. 5, 6

[4] L. Vinet and A. Zhedanov, *Rodent Bioacoustics*, vol. 67 of *Springer Handbook of Auditory Research.* Cham: Springer International Publishing, feb 2018. 5, 6, 44

[5] Www.nabr.org, "Mice & Rats: The Essential Needs for Animals in Medical Research," *Natl. Assoc. Biomed. Res.*, pp. 23–24, 2016. 6

[6] Y. K. Matsumoto and K. Okanoya, "Mice modulate ultrasonic calling bouts according to sociosexual context," *R. Soc. Open Sci.*, vol. 5, no. 6, pp. 1–15, 2018. 6

[7] M. L. Scattoni, S. U. Gandhy, L. Ricceri, and J. N. Crawley, "Unusual Repertoire of Vocalizations in the BTBR T+tf/J Mouse Model of Autism," *PLoS One*, vol. 3, p. e3067, aug 2008. 6, 7, 11, 18

[8] J. M. S. Grimsley, J. J. M. Monaghan, and J. J. Wenstrup, "Development of Social Vocalizations in Mice," *PLoS One*, vol. 6, p. e17460, mar 2011. 6, 7

[9] M. Wöhr and R. K. W. Schwarting, "Affective communication in rodents: ultrasonic vocalizations as a tool for research on emotion and motivation," *Cell Tissue Res.*, vol. 354, pp. 81–97, oct 2013. 6, 11

[10] M. Towsey, B. Planitz, A. Nantes, J. Wimmer, and P. Roe, "A toolbox for animal call recognition," *Bioacoustics*, vol. 21, pp. 107–125, jun 2012. 11

[11] V. B. Deecke and V. M. Janik, "Automated categorization of bioacoustic signals: Avoiding perceptual pitfalls," *J. Acoust. Soc. Am.*, vol. 117, pp. 2470–2470, apr 2005. 11

[12] A. Wisler, L. J. Brattain, R. Landman, and T. F. Quatieri, "A Framework for Automated Marmoset Vocalization Detection and Classification," in *Interspeech*, pp. 2592–2596, sep 2016. 11

[13] P. J. Clemins and M. T. Johnson, "Automatic type classification and speaker identification of african elephant (Loxodonta africana) vocalizations," *J. Acoust. Soc. Am.*, vol. 113, pp. 2306–2306, apr 2003. 11

[14] J. C. Brown and P. J. O. Miller, "Automatic classification of killer whale vocalizations using dynamic time warping," *J. Acoust. Soc. Am.*, vol. 122, no. 2, pp. 1201–1207, 2007. 11

[15] J. Xie, M. Towsey, J. Zhang, and P. Roe, "Frog call classification: a survey," *Artif. Intell. Rev.*, vol. 49, pp. 375–391, mar 2018. 11

[16] J. Chabout, A. Sarkar, D. B. Dunson, and E. D. Jarvis, "Male mice song syntax depends on social contexts and influences female preferences," *Front. Behav. Neurosci.*, vol. 9, pp. 1–16, apr 2015. 12

[17] C. V. Portfors, "Types and functions of ultrasonic vocalizations in laboratory rats and mice.," *J. Am. Assoc. Lab. Anim. Sci.*, vol. 46, no. 1, pp. 28–34, 2007. 12

[18] J. Chabout, P. Serreau, E. Ey, L. Bellier, T. Aubin, T. Bourgeron, and S. Granon, "Adult Male Mice Emit Context-Specific Ultrasonic Vocalizations That Are Modulated by Prior Isolation or Group Rearing Environment," *PLoS One*, vol. 7, p. e29401, jan 2012. 12

[19] M. L. Scattoni, J. Crawley, and L. Ricceri, "Ultrasonic vocalizations: A tool for behavioural phenotyping of mouse models of neurodevelopmental disorders," *Neurosci. Biobehav. Rev.*, vol. 33, no. 4, pp. 508–515, 2009. 12

[20] K. R. Coffey, R. G. Marx, and J. F. Neumaier, "DeepSqueak: a deep learning-based system for detection and analysis of ultrasonic vocalizations," *Neuropsychopharmacology*, pp. 1–10, jan 2019. 12, 13, 14, 19, 41, 87, 90

[21] J. Ramirez, J. M., and J. C., "Voice Activity Detection. Fundamentals and Speech Recognition System Robustness," in *Robust Speech Recognit. Underst.* (M. Grimm and K. Kroschel, eds.), ch. 1, Rijeka: I-Tech Education and Publishing, jun 2007. 12, 16

[22] A. Harma, "Automatic identification of bird species based on sinusoidal modeling of syllables," in *2003 IEEE Int. Conf. Acoust. Speech, Signal Process. 2003. Proceedings. (ICASSP '03).*, vol. 5, pp. V–545–8, IEEE, 2004. 12

[23] C. Paper, H. Jaafar, D. A. Ramli, B. A. Rosdi, and S. Shahrudin, *The 8th International Conference on Robotic, Vision, Signal Processing & Power Applications*,

vol. 291 of *Lecture Notes in Electrical Engineering*. Singapore: Springer Singapore, 2014. 12, 13

[24] S. M. Zala, D. Reitschmidt, A. Noll, P. Balazs, and D. J. Penn, "Automatic mouse ultrasound detector (AMUD): A new tool for processing rodent vocalizations," *PLoS One*, vol. 12, no. 7, pp. 3–9, 2017. 12, 13, 17, 19

[25] T. E. Holy and Z. Guo, "Ultrasonic Songs of Male Mice," *PLoS Biol.*, vol. 3, p. e386, nov 2005. 12, 13, 14, 18, 19, 87

[26] I. Potamitis, S. Ntalampiras, O. Jahn, and K. Riede, "Automatic bird sound detection in long real-field recordings: Applications and tools," *Appl. Acoust.*, vol. 80, pp. 1–9, jun 2014. 12, 13

[27] M. Van Segbroeck, A. T. Knoll, P. Levitt, and S. Narayanan, "MUPET—Mouse Ultrasonic Profile ExTraction: A Signal Processing Tool for Rapid and Unsupervised Analysis of Ultrasonic Vocalizations," *Neuron*, vol. 94, no. 3, pp. 465–485.e5, 2017. 12, 14, 16, 19

[28] J. Xie, M. Towsey, J. Zhang, and P. Roe, "Acoustic classification of Australian frogs based on enhanced features and machine learning algorithms," *Appl. Acoust.*, vol. 113, pp. 193–201, dec 2016. 13, 61

[29] C.-h. Lee, C.-H. Chou, C.-C. Han, and R.-Z. Huang, "Automatic recognition of animal vocalizations using averaged MFCC and linear discriminant analysis," *Pattern Recognit. Lett.*, vol. 27, pp. 93–101, jan 2006. 13

[30] C.-j. Huang, Y.-J. Yang, D.-x. Yang, and Y.-j. Chen, "Frog classification using machine learning techniques," *Expert Syst. Appl.*, vol. 36, pp. 3737–3743, mar 2009. 13

[31] J. Noda, C. Travieso, and D. Sánchez-Rodríguez, "Fusion of Linear and Mel Frequency Cepstral Coefficients for Automatic Classification of Reptiles," *Appl. Sci.*, vol. 7, p. 178, feb 2017. 13, 59, 61

[32] D. A. Ramli and H. Jaafar, "Peak Finding Algorithm to Improve Syllable Segmentation for Noisy Bioacoustic Sound Signal," *Procedia Comput. Sci.*, vol. 96, pp. 100–109, 2016. 13

[33] W.-P. Chen, S.-S. Chen, C.-C. Lin, Y.-z. Chen, and W.-c. Lin, "Automatic recognition of frog calls using a multi-stage average spectrum," *Comput. Math. with Appl.*, vol. 64, pp. 1270–1281, sep 2012. 13

[34] A. Ivanenko, P. Watkins, M. A. J. van Gerven, K. Hammerschmidt, and B. Englitz, "Classification of mouse ultrasonic vocalizations using deep learning," *bioRxiv*, p. 358143, jan 2018. 13, 37

[35] G. Arriaga, E. P. Zhou, and E. D. Jarvis, "Of Mice, Birds, and Men: The Mouse Ultrasonic Song System Has Some Features Similar to Humans and Song-Learning Birds," *PLoS One*, vol. 7, p. e46610, oct 2012. 13, 18

[36] Z. D. Burkett, N. F. Day, O. Peñagarikano, D. H. Geschwind, and S. A. White, "VoICE: A semi-automated pipeline for standardizing vocal analysis across models," *Sci. Rep.*, vol. 5, p. 10237, sep 2015. 18, 19

[37] M. L. Scattoni, L. Ricceri, and J. N. Crawley, "Unusual repertoire of vocalizations in adult BTBR T+tf/J mice during three types of social encounters," *Genes, Brain Behav.*, vol. 10, pp. 44–56, feb 2011. 18

[38] L. R. Saraiva, K. Kondoh, X. Ye, K.-h. Yoon, M. Hernandez, and L. B. Buck, "Combinatorial effects of odorants on mouse behavior," *Proc. Natl. Acad. Sci.*, vol. 113, no. 23, pp. E3300—-E3306, 2016. 23

[39] M. E. Wang, N. P. Fraize, L. Yin, R. K. Yuan, D. Petsagourakis, E. G. Wann, I. A. Muzzio, and W. E. T. Al, "Differential Roles of the Dorsal and Ventral Hippocampus in Predator Odor Contextual Fear Conditioning," vol. 466, pp. 451–466, 2013. 23

[40] C. V. Portfors and D. J. Perkel, "The role of ultrasonic vocalizations in mouse communication," *Curr. Opin. Neurobiol.*, vol. 28, pp. 115–120, oct 2014. 23

[41] S. Vajda, C. E. Shannon, and W. Weaver, "The Mathematical Theory of Communication," *Math. Gaz.*, vol. 34, p. 312, dec 1950. 36

[42] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Work. Stat. Learn. Comput. Vision, ECCV*, pp. 1–22, 2004. 38

[43] Y. Zhang, R. Jin, and Z.-H. Zhou, "Understanding bag-of-words model: a statistical framework," *Int. J. Mach. Learn. Cybern.*, vol. 1, pp. 43–52, dec 2010. 38, 39

[44] A. Lerch, *An Introduction to Audio Content Analysis.* Hoboken, NJ, USA: John Wiley & Sons, Inc., jul 2012. 40, 41, 42, 43, 44

[45] M. Jalil, F. A. Butt, and A. Malik, "Short-time energy, magnitude, zero crossing rate and autocorrelation measurement for discriminating voiced and unvoiced segments of speech signals," in *2013 Int. Conf. Technol. Adv. Electr. Electron. Comput. Eng.*, no. m, pp. 208–212, IEEE, may 2013. 41

[46] J. Xie, M. Towsey, J. Zhang, X. Dong, and P. Roe, "Application of image processing techniques for frog call classification," in *Proc. - Int. Conf. Image Process. ICIP*, vol. 2015-Decem, pp. 4190–4194, IEEE, sep 2015. 46

[47] M. A. Roch, T. Scott Brandes, B. Patel, Y. Barkley, S. Baumann-Pickering, and M. S. Soldevilla, "Automated extraction of odontocete whistle contours," *J. Acoust. Soc. Am.*, vol. 130, pp. 2212–2223, oct 2011. 46

[48] A. Mallawaarachchi, S. H. Ong, M. Chitre, and E. Taylor, "Spectrogram denoising and automated extraction of the fundamental frequency variation of dolphin whistles," *J. Acoust. Soc. Am.*, vol. 124, no. 2, pp. 1159–1170, 2008. 46

[49] H. Ou, W. W. L. Au, L. M. Zurk, and M. O. Lammers, "Automated extraction and classification of time-frequency contours in humpback vocalizations," *J. Acoust. Soc. Am.*, vol. 133, no. 1, pp. 301–310, 2013. 46

[50] M. Esfahanian, H. Zhuang, and N. Erdol, "Sparse representation for classification of dolphin whistles by type," *J. Acoust. Soc. Am.*, vol. 136, pp. EL1–EL7, jul 2014. 46

[51] D. Iatsenko, P. V. McClintock, and A. Stefanovska, "Extraction of instantaneous frequencies from ridges in time-frequency representations of signals," *Signal Processing*, vol. 125, pp. 290–303, 2016. 46

[52] P. Flandrin, F. Auger, and E. Chassande-Mottin, "Time,ÄìFrequency Reassignment," no. January, pp. 179–203, 2010. 46

[53] T. Oberlin, S. Meignen, T. Oberlin, S. Meignen, T. F.-b. Synchrosqueezing, T. Oberlin, S. Meignen, and V. Perrier, "The Fourier-based Synchrosqueezing Transform erie Perrier To cite this version :," 2013. 46

[54] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006. 51

[55] P. F. Alcantarilla, A. Bartoli, and A. J. Davison, "KAZE Features," pp. 214–227, 2012. 55, 56

[56] S. A. K. Tareen and Z. Saleem, "A comparative analysis of SIFT, SURF, KAZE, AKAZE, ORB, and BRISK," *2018 Int. Conf. Comput. Math. Eng. Technol. Inven. Innov. Integr. Socioecon. Dev. iCoMET 2018 - Proc.*, vol. 2018-Janua, pp. 1–10, 2018. 55

[57] P. Martins, P. Carvalho, and C. Gatta, "On the completeness of feature-driven maximally stable extremal regions," *Pattern Recognit. Lett.*, vol. 74, pp. 9–16, apr 2016. 56

[58] R. Valdes-perez, "Theoretical and Empirical Analysis of ReliefF and RReliefF," pp. 23–69, 2003. 58

[59] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006. xvi, 59, 61, 64

[60] Chih-Wei Hsu and Chih-Jen Lin, “A comparison of methods for multiclass support vector machines,” *IEEE Trans. Neural Networks*, vol. 13, pp. 415–425, mar 2002. 59

[61] S. Gunasekaran and K. Revathy, “Content-based classification and retrieval of wild animal sounds using feature selection algorithm,” in *ICMLC 2010 - 2nd Int. Conf. Mach. Learn. Comput.*, pp. 272–275, 2010. 61

[62] J. R. Quinlan, “Induction Fo Decion Trees,” *Mach. Learn.*, vol. 1, pp. 81–106, 1986. 62