NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

LUÍS FILIPE SOBRAL DO ROSÁRIO

Degree in Computer Science and Engineering

# VOCAL SIGNATURE FEATURE SET FOR THE DISTINCTION OF MACARONESIAN DOLPHIN SPECIES

MASTER IN COMPUTER SCIENCE

NOVA University Lisbon
November, 2021

NOVA SCHOOL OF SCIENCE & TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE

# VOCAL SIGNATURE FEATURE SET FOR THE DISTINCTION OF MACARONESIAN DOLPHIN SPECIES

**LUÍS FILIPE SOBRAL DO ROSÁRIO**

Degree in Computer Science and Engineering

**Adviser:** Joaquim Francisco Ferreira Da Silva
*Assistant Professor, NOVA University of Lisbon*

**Co-adviser:** Sofia Carmen Faria Maia Cavaco
*Assistant Professor NOVA University of Lisbon*

**Examination Committee:**

**Chair:** Carlos Augusto Isaac Piló Viegas Damásio
*Associate Professor, NOVA University of Lisbon*

**Rapporteur:** Luís Miguel Parreira e Correia
*Associate Professor, Faculdade de Ciências of Universidade de Lisboa*

**Adviser:** Joaquim Francisco Ferreira Da Silva
*Assistant Professor, NOVA University of Lisbon*

MASTER IN COMPUTER SCIENCE

NOVA University Lisbon
November, 2021

**Vocal Signature Feature Set for the Distinction of Macaronesian Dolphin species**

# Acknowledgements

Here I would like to thank several people that were instrumental for the development and success of this dissertation. Firstly, I would like to thank the entire research team of the Madeira Whale Museum for all of their collaboration and shared knowledge besides the fundamental help they presented in obtaining the needed cetacean recordings for this work.

Then, I would like to thank both my adviser Prof. Dr. Joaquim Silva and co-adviser Prof. Dr. Sofia Cavaco, for all the continuous support, motivation and exceptional guidance given to me during the last year. Without them this work would not be possible. I would like to extend my gratitude towards the Department of Computer Science of NOVA School of Science  Technology and NOVA LINCS, for all of the great learning structure they provided me over the past years.

I want to thank my family for all of their support, love and encouragement shown during my academic path. Reaching this point would not be possible if it was not for them. I also want to thank all the friends I made throughout these years, thanks for all the great moments and support you gave me. Finally, I want to thank my girlfriend for all the love, patience and support shown to me during this last year.

*"If you no longer go for a gap that exists, you are no longer a racing driver" (Ayrton Senna)*

# Abstract

In this dissertation we approach the problem of performing bioacoustic classification of four different small dolphin species by using their vocalizations.

Cetaceans, (the taxonomic order which dolphins are part of) live in complex social societies and have been known to possess remarkable cognitive skills, being praised to have great intelligence capabilities. Cetaceans are most well known for their intricate communication patterns, which serve different purposes from mating advertisement to individual recognition. The analysis of these vocalizations, due to their intricacy has been for decades a laborious manual task, which takes a long time for specialists to perform. Our interest is in aiding researchers by developing machine learning methods capable of the analysis and classification of great volumes of cetacean recordings.

We propose a four stage method which is capable of extracting relevant features from dolphin vocalizations making it possible to identify the corresponding species with great accuracy (achieving model accuracies above 95%). Although the resulting model is tailored to the classification of cetacean species indigenous to the Madeira Archipelago, which is expected to help the Madeira Whale Museum's conservation efforts of these animals, it can be the foundation for future classifications of other cetacean species.

**Keywords:** Bioacoustic Classification, Cetaceans, Marine bioacoustic signal processing, Supervised classification, Denoising, Convolution neural networks

# Resumo

Esta dissertação aborda a temática da classificação bioacústica de quatro espécies de golfinhos com base nas suas vocalizações.

Os cetáceos (infraordem taxonómica os golfinhos se encontram) vivem em complexos aglomerados sociais e demonstram elevadas capacidades cognitivas, sendo considerados animais altamente inteligentes. Estas espécies são especialmente conhecidas pelos seus intrincados chamamentos que, podem servir diversos propósitos tais como para acasalamento e identificação. A análise destas vocalizações, devido à sua complexidade, foi desde sempre uma tarefa manual laboriosa que demora grandes períodos de tempo a ser realizada. De modo a facilitar este processo, é do nosso interesse a produção de um modelo de aprendizagem automática, capaz de analisar e classificar grandes volumes de gravações de cetáceos.

Propomos a produção de um método a quatro fases capaz de obter features relevantes a partir de vocalizações de golfinhos, tornando assim possível a sua distinção com uma grande precisão (chegando-se a alcançar precisões médias acima de 95%). Independentemente de o modelo resultante estar otimizado para a classificação de espécies indígenas do arquipélago da Madeira, algo que poderá ajudar os esforços de conservação destas espécies levados a cabo pelo Museu da Baleia da Madeira, também poderá servir como base para outros trabalhos futuros na área da classificação bioacústica.

**Palavras-chave:** Classificação Bioacústica, Cetáceos, Processamento de sinais bioacústicos marinhos, Classificação supervisionada, Remoção de ruído, Redes neuronais convolucionais

# Contents

# List of Figures

# List of Tables

# Acronyms

**ANN** Artificial Neural Network 21, 65

**CNN** Convolution Neural Networks ix, xii, xiii, xiv, 4, 20, 21, 26, 27, 46, 47, 48, 49, 50, 52, 63, 64, 79, 80

**DBSCAN** Density-based spatial clustering of applications with noise 18, 49
**DCT** Discrete Cosine Transform 36
**DTAGs** Digital Acoustic Recording Tags 2

**FFT** Fast Fourier Transform 10, 34

**ICA** Independent Component Analysis xiv, 19, 46, 50, 51, 52, 53, 63

**K-NN** K-Nearest Neighbours xiv, xv, 4, 14, 26, 46, 47, 49, 50, 51, 53, 54, 55, 56, 63, 64

**LBP** Local Binary Patterns 25
**LFCC** Linear Cepstral Coefficients 25, 26

**MFCC** Mel-Frequency Cepstral Coefficients xii, xiv, 4, 12, 25, 26, 34, 35, 36, 37, 40, 49, 51, 52, 53, 54, 55, 56, 63, 64
**MLP** Multilayer Perceptron 23

**PAM** Passive Acoustic Monitoring 2, 63
**PCA** Principal Component Analysis 19, 46, 51, 52

**STFT** Short-time Fourier Transform xii, xiii, 4, 10, 23, 26, 34, 35, 36, 37, 40, 50, 51, 52, 53, 54, 55, 56, 61, 62, 63, 64

**SVM** Support Vector Machines xv, 4, 15, 16, 25, 26, 46, 47, 49, 50, 51, 53, 55, 56, 57, 58, 59, 63, 64

**WPT** Wavelet Packet Transform 23

**WT** Wavelet Transform 26, 27

# 1

# Introduction

## 1.1 Motivation

Cetaceans are a group of aquatic mammals which belong to the taxonomic order Cetacea and can be further divided in two sub orders, the Mysticeti (cetaceans with baleen plates) and Odontoceti (toothed cetaceans) [1]. This taxon, which is composed by every species of whales, dolphins and porpoises, displays a wide variety of complex social structures, social behaviors and communication patterns [2]. As a result, these animals have been a target of innumerous studies for several decades with the intention of better understanding their behaviours. One area of study that has always been seen with great interest by scientists is the study of their vocalizations.

Cetaceans produce a wide range of different vocalizations, each with a different call signature and purpose [2]. Humpback whale songs, which are one of the most researched vocalizations made by cetaceans, have been found to be mostly produced by lone males during the breeding season, which suggests these may be some kind of mating advertisement [3]. These songs are relatively complex, consisting of a juxtaposition of individual patterns which are arranged in a repeating manner to form themes. However, there is still a big debate in the scientific community regarding the purpose of these songs and their common structure [2]. Bottlenose dolphins produce whistles which can be used for individual recognition, as they can be perceived by other individuals due to its frequency modulation pattern [2, 4]. These whistles can also be used to maintain group cohesion, which is essential to an individual's security. Sperm whales also employ vocalizations to identify individuals and even groups, by using standardized patterns of clicks called codas, which are shared between social groups, facilitating their recognition [2, 3]. This phenomenon has also been observed in other species which have highly stable matrilineal communities, as is the case of orca whales. Another common vocalization among some cetacean species is the occurrence of narrow band high frequency clicks, which are used either for echolocation and/or communication [2]. These are suggested to be an anti-predator adaptation, taking advantage of the lack of hearing sensitivity of their natural predators (e.g. orca whales) at high frequencies (above 100 kHz). These calls are mostly

used at short distance, as they suffer from range-dependent absorption due to their high frequency rates [5]. Contrarily, some cetacean species like the blue and fin whales produce sustained low frequency calls which are used for long distance communication that, due to their low energy, can reach up to 90 km in an ocean basin [6].

Developments in the study of cetacean vocalizations are crucial to assess the impact of anthropogenic noise in cetacean feeding and breeding patterns [2]. This in turn could help to improve some conservation efforts for these species, as well as to better understand the impact of vocal production learning in the communication of several cetacean species. Some species have been recorded to learn vocal patterns from other individuals [2]. Examples of this are, the bottlenose dolphins being able to label objects with newly learnt whistle patterns; orcas and beluga whales being able to imitate other species signal calls and a reported synchronous change in humpback whale song elements in a population [2]. These behaviours are extremely rare and only have been observed in few other mammals like bats, elephants, seals and humans [2, 7], rendering their study important to establish comparisons between the communication of cetaceans and the human language. This could prove helpful to theorize on how the human language has evolved and what unique aspect separate it from the communication of species that evolved in different environments [2].

In order to study these vocalizations, biologists can make use of Passive Acoustic Monitoring (PAM) techniques to obtain large bioacoustic recording archives. These techniques can be seen in Figure 1.1, and may vary from using Digital Acoustic Recording Tags (DTAGs), which are placed on marine animals, to the usage of arrays of hydrophones attached to buoys or mounted to the seafloor [8–11]. These approaches have the benefit of being non-invasive and capable of recording animals with as few external disturbances to their natural habitats as possible, greatly improving the chances of capturing calls in an animals' natural behavioural context [12, 13]. However, even with call detection algorithms, these methods produce many hours of recording, that must be analysed by experts. The recordings may be used for many tasks like species-identification, localization-tracking, behaviour-analysis, or population monitoring [11].

Due to their specificity, these analysis have always been a manual task, and with the increasing size of new datasets it has become more time consuming than ever. This increase in size can also lead to classification inconsistencies depending on the expertise and fatigue of the analyst [11]. Additionally, the presence of noise in the recordings, which can be caused by underwater background noises, boat sounds or microphone artifacts, can difficult its analysis.

Due to these problems, there is an increasing interest in using automated computational methods which could mitigate the aforementioned problems, improving the efficiency and accuracy of these tasks, while removing human bias [15]. As such, this could lead to significant advances in the study of cetacean communication, with all its benefits.

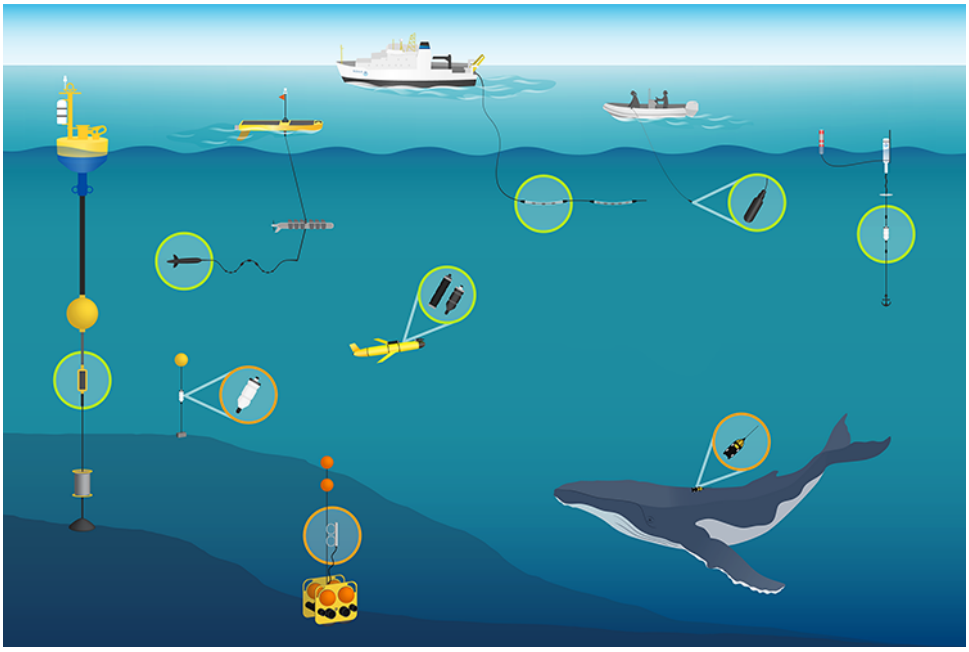


Figure 1.1: Representation of several acoustic monitoring techniques [14].

## 1.2 Objectives

This dissertation aims to contribute to the development of the cetacean biocoustics field and consequently, the conservation effort of these species. For this purpose, we propose the creation of a set of tailor-made features, which in conjunction with machine learning classifiers are capable of accurately distinguish the vocalizations of four different small species of dolphins, even in the presence of substantial amounts of noise.

In addition, we also propose a comparative study of several technical approaches, studying different feature extraction and classification methods. By doing this, we can assess the benefits and drawbacks that come with each approach, which can then be used as a reference for some future work.

This work is made in collaboration with marine biologists from the Madeira Whale Museum, which provided cetacean vocalization recordings, obtained for the purpose of this project. The area surrounding the Madeira archipelago, presents itself as a passing, feeding and reproduction ground for at least 20 different cetacean species [16]. The utilization of recordings of species indigenous to the area also imposes the objective of obtaining a tailored model for the identification of cetacean vocalizations in this area, which in turn can be used to aid the museum's conservation efforts. For this purpose, we decided on using only vocalizations of 4 dolphin species, which will produce a model capable of distinguishing these species: common dolphin, bottlenose dolphin, atlantic spotted dolphin and striped dolphin.

## 1.3 Proposed solution - Four stage classification method

In order to accomplish the mentioned objectives, we propose a four stage method, which is depicted in Figure 1.2. On the first stage (preprocessing stage), the raw recordings which were collected at sea are sliced into smaller equal sized segments contain cetacean vocalizations. These are then subjected to a denoising procedure, which improves the quality of their signal. Then, on the second stage (time-frequency representation) each segment is represented by an adequate time-frequency representation which then undergoes a feature extraction method on the third stage. The feature extraction stage obtains a feature array (for each data segment) characterizing the vocalization present on that specific segment. These feature arrays are then used on the fourth and final stage of our method, which culminates in the usage of supervised classification methods to distinguish the vocalizations of four small dolphin species.

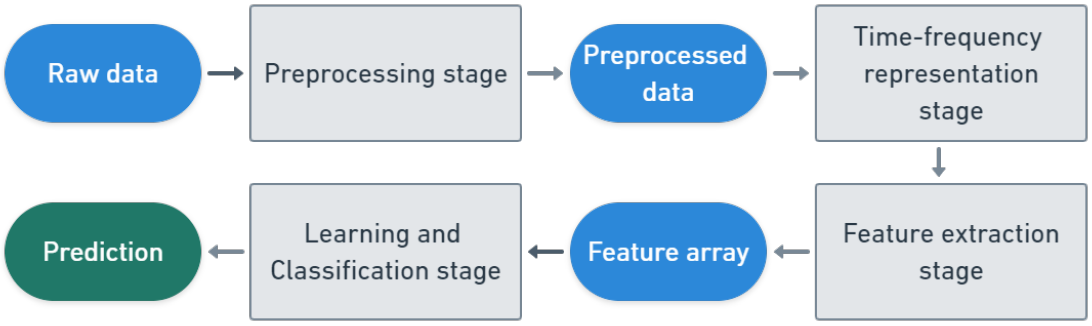



Figure 1.2: Pipeline of the proposed four stage method to classify small dolphin vocalizations. Each gray rectangle represents a process (one of the stages) to be applied over data (blue pills)

As to fulfil the objective of performing a comparative study of different technical approaches, several classification models (SVM, K-NN, CNN), as well as two distinct time-frequency representation (STFT and MFCC) will be tested in the proposed method. From the resulting work produced by this dissertation, we higlight the following contributions:

- Research concerning several different feature combinations and classification algorithms, applied to the domain of dolphin bioacoustic classification which resulted in a set of features capable of characterizing and distinguishing the vocal signatures of different dolphin species;
- A robust identification method capable of distinguishing four different small dolphin species by using their vocalizations;
- A submitted conference paper regarding this research.

## 1.4 Document outline

To better understand the proposed work, this dissertation is divided into the following chapters:

- **Chapter** 1, provides a contextualization of the problem at hand, explaining the motivation behind the need to solve it. A brief overview of the proposed solution is also presented here;
- **Chapter** 2, presents some fundamental concepts regarding sound and machine learning, which may be important to fully comprehend this work;
- **Chapter** 3 introduces some of the current state of the art in the field of bioacoustic classification, presenting some past attempts at solving similar problems to the one stated in this work. Here, several classification approaches and preprocessing techniques will be highlighted and compared;
- **Chapter** 4 highlight the technical approach used to solve the problem presented in this dissertation.
- **Chapter** 5 presents the obtained experimental results during the development of this work.
- **Chapter** 6 presents some closing remarks regarding the performed work while also alluding to some potential future development of this research.

# 2

# Fundamental Concepts

In order to better understand the work and research done, this chapter presents some background knowledge about some fundamental concepts on sound and machine learning that will be used throughout this dissertation. A reader more familiarized with the these topics can skip to Chapter 3, however if any of these topic are not mastered by the reader, their reading is advised. The information compiled in Section 2.1 is not included in the current course syllabus, being mostly sourced from [17]. The contents of Section 2.2 are conteplated in [18] and [19].

## 2.1 Sound

In this section we will delve down into some basic notions of sound, such as some of its properties and how sound is propagated. Here, we will also touch on some concepts intrinsic to sound analysis/processing such as frequency analysis, the Mel-frequency cepstrum and noise filtering.

### 2.1.1 Basic notions

Sound can be seen as a consequence of setting an object into a vibrating motion given that the object has the properties of elasticity and inertia [17]. Inertia referring to a force that was exerted on the object for it to move and elasticity being the ability of an object to return to an initial state after it has been set into motion. Therefore, any object which has these properties has potential to produce sound.

However, this does not guarantee that the produced sound is audible. For that to happen, firstly sound needs a medium to be propagated through in order for it to reach our eardrums (*tympanic membrane*) [17]. Sound is a pressure wave that travels by bumping adjacent molecules into each other as they vibrate. These molecules will subsequently propagate these vibrations, moving in the same direction the sound wave is moving, until it reaches our auditory system. This is why sound does not propagate in vacuum, as it is a medium which lacks the properties of inertia and elasticity needed for molecules in that medium to pass along a wave of motion.

When propagating a sound wave, molecules in the propagation medium are moved back and forth through several stages of condensation and rarefaction [17]. The condensation stage corresponds to a high pressure stage where the molecules are compressed together as the vibrating wave moves away from its resting state. When the vibrating object starts to return to its resting state, the molecules start to decompress, filling the vacant space previously occupied by the wave. When the wave goes back past its resting state it translates in a decreased pressure state where molecules will become more disperse corresponding to a rarefaction stage. This can be seen in Figure 2.1.

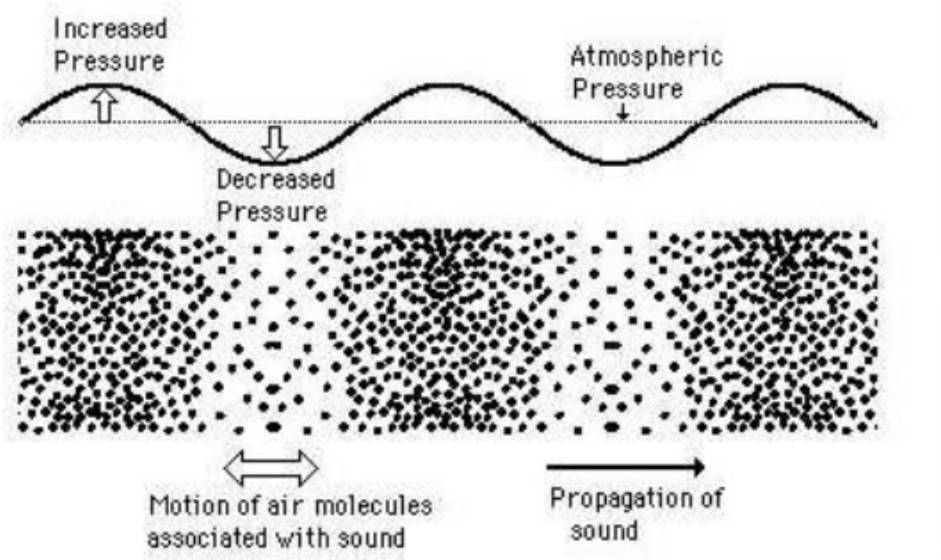



Figure 2.1: Propagation of a sound wave using air as a transmission medium. Here we can observe how the condensation and rarefaction stages are related to atmospheric pressure [20].

Different mediums of propagation will have different effects in sound propagation [21, 22]. In water, the molecules are generally closer together than in air, making it is much denser (800 times more). The density of the medium, which is affected by temperature and pressure, and its compressibility, i.e. the capability of a solid or liquid changing its volume as a response to pressure, have a direct impact on the speed of sound. In general, the less compressible and dense a medium is, the faster the speed of sound throughout it. Due to being harder to compress, liquids normally present a higher speed of sound in its medium than gases.

Other properties like the viscosity of the medium, which translates the rate at which sound is attenuated also must be considered, due to it impacting the absorption of sound and consequently, the distance a sound can be propagated [23, 24]. In general, sound absorption is lower in water than in the air (being remarkably low within seawater due to a chemical relaxation process), and it is mostly predominant with higher frequency waves. Due to this, generally sound can be propagated through longer distances in water.

#### 2.1.1.1 Sinusoids

Sound can be seen as a sum of particular types of vibration called sinusoidal vibrations [17]. A sinusoid is the simplest type of vibration and describes relationship between displacement and time. This vibration has a constant continuous back and forth oscillation between a maximum and minimum oscillation point as seen in Figure 2.2. Displacement is measured as the distance between the resting point and the current oscillating point for a given time instance.

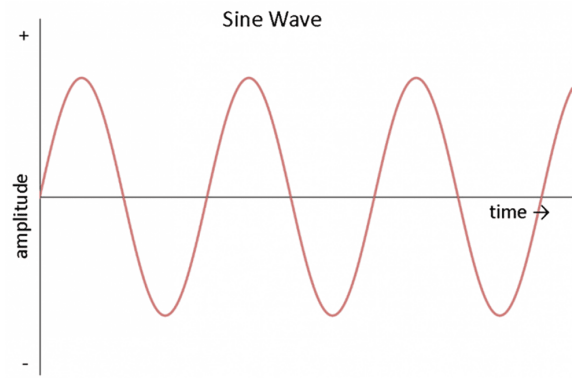



Figure 2.2: Sinusoid or sine wave, which describes relation between displacement (Amplitude) and time [25].

Sinusoids can be described by their properties, namely **Frequency**, **Amplitude** and **Starting phase**. Because any vibration consists of a sum of one or more sinusoids, it is possible to describe any vibration by stating these properties of the various sinusoids that constitute it. A vibration consisting of more than one sinusoid is referred to as a **complex vibration** whereas a vibration consisting of a single sinusoid is a **simple vibration**.

#### 2.1.1.2 Frequency

Frequency of a sinusoid is defined by the number of complete cycles, i.e number of times a sine wave begins and ends in the same point of displacement, per second [17]. The frequency of a sine wave is expressed in Hertz (Hz), meaning that a wave which completes 10 cycles per second has a frequency of 10 Hz. Frequency is inversely proportional to the **period** of a wave (expressed in seconds), which refers to the amount of time a wave takes to complete one cycle. This relation between frequency and period can be seen in Equation 2.1:

$$period = \frac{1}{frequency} \tag{2.1}$$

Because of this, both period and frequency can be used to describe the oscillation of a sine wave.

#### 2.1.1.3 Amplitude

Amplitude corresponds to the vibratory displacement of a given wave [17]. This property can be stated in different ways. When referring to a time-varying displacement we are referring to **instantaneous amplitude**, which translates into the displacement for a given time instance t. Instantaneous amplitude is expressed by equation 2.2, where $A$ corresponds to the maximum amplitude, $f$ is the frequency of the wave, $t$ is the time instance and $\theta$ corresponds to the starting phase [17].

$$D(t) = A\sin(2\pi f t + \theta) \tag{2.2}$$

When referring to a non-time-varying amplitude we are either referring to **peak amplitude**, **peak-to-peak amplitude** or **root-mean-square amplitude**. Peak amplitude is the maximum displacement a wave achieves in one period. Peak-to-peak amplitude corresponds to the distance between the maximum positive displacement and the maximum negative displacement of a waveform in one period. Both types of amplitude are represented in Figure 2.3.

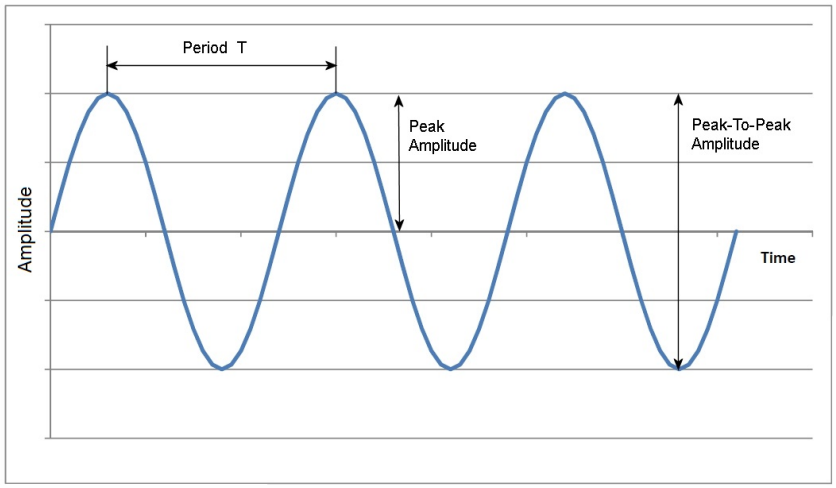



Figure 2.3: Representation of peak amplitude and peak-to-peak amplitude [26].

However, these two approaches have the problem of not properly describing the amplitude of complex non-sinusoidal waveforms with time varying peak amplitudes [17]. In these cases, we might want to use the average amplitude to establish a proper comparison between waveforms, however for sine waves (like the one in Figure 2.3) this would not work due to the average being zero. In order to solve this we use **root-mean-square amplitude**, which calculates the square of every instantaneous amplitudes of a waveform and then computes the average of these values followed by the square root of this average, giving us a non zero result which is directly proportional to the peak amplitude or peak-to-peak amplitude values.

#### 2.1.1.4 Starting phase

The starting phase corresponds to the displacement point where a waveform starts when an object starts to vibrate [17]. This value is expressed in degrees of rotation, with 0º corresponding to the start of the initial period and 360º the end of the same period.

If two sinusoids have the same frequency but different starting phases they are said to be out of phase as seen in Figure 2.4. In this case the starting phase difference will always be equal to the instantaneous phase difference between the two sinusoids.

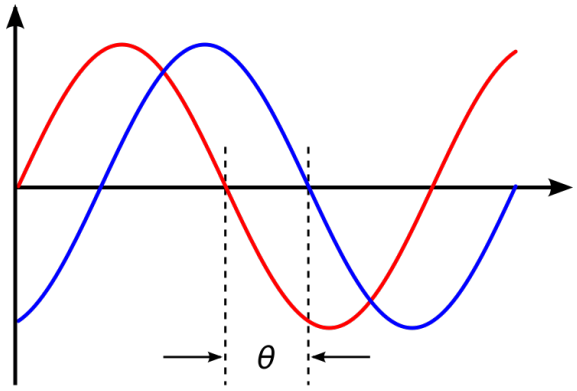



Figure 2.4: Here we can see two out of phase sinusoids. The red waveform has a starting phase of 0 and the blue one has a starting phase $\theta$ [27].

### 2.1.2 Frequency analysis

Fourier analysis states that a waveform can be decomposed into the sum of several other constituent vibrations [17]. The Fourier Transform is a method which can convert a signal from a time-domain to a frequency-domain, that can better reflect the constituent frequencies of that signal. This frequency domain can be represented in the form of a spectrum. Apart from better representing a signal's frequency, it also can reflect the magnitude of each given frequency [28].

Fast Fourier Transform (FFT) is an algorithm that efficiently calculates the Discrete Fourier Transform of a discrete signal. It starts by converting it into its frequency constituents into several equal intervals much like the Fourier Transform does for continuous signals [28].

Short-time Fourier Transform (STFT) is a method that consists in the segmentation of a signal into temporal segments, followed by the application of the FFT with a given set of parameters, to each of it [17]. These parameters refer to the frame size, window size and hop size (which determines the amount of overlap between frames) and their purpose is better ilustrated later in Section 4.3. Doing this results in a spectrum for each of the time segments, which together allows for the representation of a frequency-domain (frequency and amplitude) in a temporal context. This allows for the creation of spectrograms.

A spectrogram is a graphical representation of a waveform's frequency energy (intensity) over a given time [29]. Spectrograms can be two-dimensional graphs with a third dimension in the form of a color, representing the sound intensity, or it can be a regular three-dimensional graph.

Time is represented in the horizontal axis, while the recorded frequency range is represented in the vertical axis. The intensity of a given frequency in a given time is represented by a color scale gradient according to its magnitude at the time.

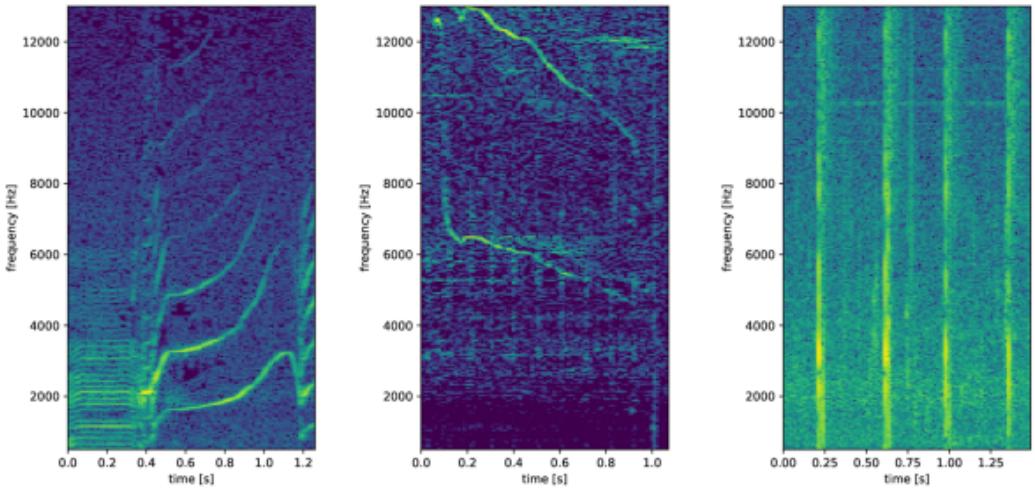



Figure 2.5: Three spectrograms of different killer whale characteristic sounds. From left to right, a pulsed call, a whistle and a echolocation click [3].

Spectrograms are very useful in the study of bioacoustics as their analysis can lead to a better understanding of animal communications and their vocalization characteristics. To support this claim, in Figure 2.5 we can observe three spectrograms of different killer whale calls. By analysing each one we can easily identify the three distinct calls as they possess clear distinguished sound profiles with different frequency ranges and time domains. For example, the pulsed call has a narrower frequency range than the echolocation pulse, while being a longer call.

### 2.1.3 Mel-frequency cepstrum

Pitch is a subjective property of sounds that expresses the perceived frequency by a listener [17]. A change in pitch traduces itself to a change along many sound property dimensions (non-linear), and thus cannot be solely attributed to a change in frequency. This is one of the reasons that our perception of pitch is extremely subjective, varying from person to person. However, a strong correlation between the pitch of a sound with the spectral location of its frequencies can be established.

The Mel scale is based on a mapping between actual frequency and perceived pitch as the human auditory system does not perceive pitch in a linear manner [17]. By doing this, it scales the frequency in order to match more closely the way the human ear processes

sound, with the mapping being linear bellow 1000 Hz and logarithmic above that. This is evident in Figure 2.6, where Mel bands are mapped from frequency spectra, with the spectrum sizes increasing in a logarithmic fashion.

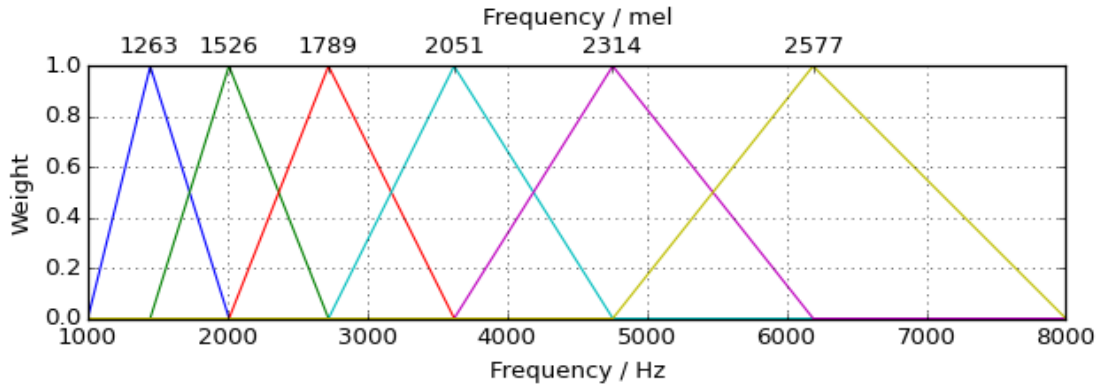



Figure 2.6: Mel filter bank [30]

Mel-frequency cepstrum is a representation of the energy present in the different Mel-frequency bands, which in turn can express the features of a sound [31]. The variation of this energy can be expressed by Mel-Frequency Cepstral Coefficients (MFCC) which contain the information of rate of change in the Mel-frequency bands. These coefficients are logarithmic in order to match the non-linearity of the frequency perception capabilities of the human ear. These log-Mel features are a widely used time-frequency feature representation, especially for speech signals, however these have achieved promising results when used with underwater signals in deep-learning methods [31].

.

### 2.1.4 Denoising signals

When dealing with sound processing sometimes we have to work with noisy signals, i.e. signals that contain unwanted secondary noise. It is possible to distinguish between two main types of noise: (1) random noise, which is characterized by an instantaneous amplitude that varies over time in a random manner; and (2) background noise, which is caused by external factors such as noises made by boat motors or rain [17]. If the random noise instantaneous amplitude changes according to a probabilistic distribution such as Gaussian or normal distribution, we have **Gaussian noise**. In the particular case that our signal has a persistent average intensity with a flat power spectrum present in a given frequency range, we would be dealing with **white noise** [17].

In most cases it is desired to remove these unwanted noisy components in order to have a noise free signal to work with. To do so, several different signal processing techniques exist to tackle this problem. If we identify that the noise is being caused by a

specific band of frequencies a filter can be applied. These filters attenuate the amplitudes of a given band of frequencies while maintaining the remaining signal frequencies untouched. Some examples of these filters are the low-pass filter, high-pass filter, band-reject filter and band-pass filters. The practical utilization of other denoising methods will be presented in more detail in Section 3.2.

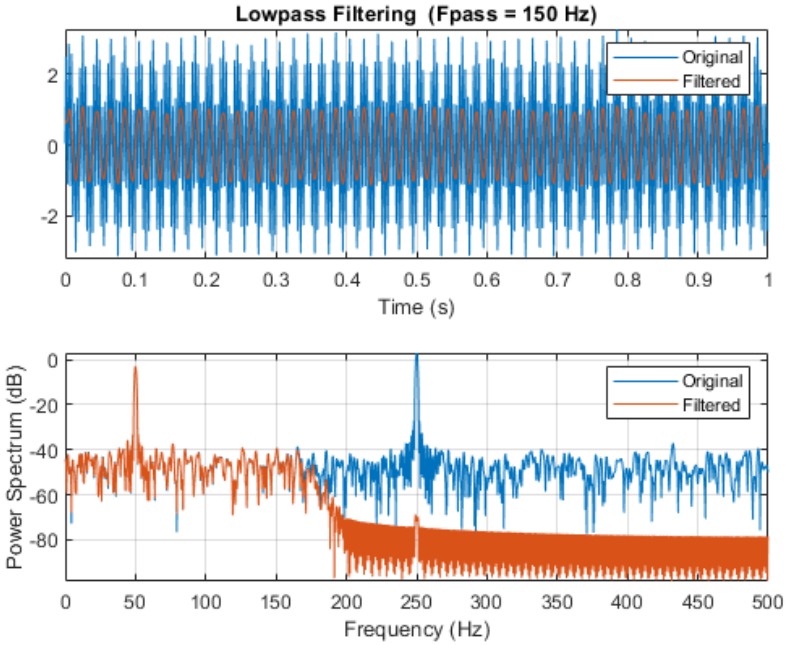



Figure 2.7: Example of the utilization of a low-pass filter at 150Hz to a noisy signal. We can observe the reduction in amplitude for any frequency greater than 150Hz in the filtered signal [32].

A **low-pass filter** (Figure 2.7), will pass all sinusoids with frequencies below a particular value while attenuating the amplitudes of the remaining frequencies above it. A **High-pass filter** would do the opposite of the low-pass filter, while a **band-reject filter** would attenuate the amplitudes of all frequencies inside a given frequency interval.

## 2.2 Machine learning

This section introduces some machine learning concepts mostly about supervised learning classifiers and clustering. Other more specific topics like deep learning and feature selection are also mentioned.

### 2.2.1 Supervised learning

Supervised learning consists in learning a function which is able to map input to output values from a set of training samples containing valid input-output mappings [18]. As what we intend our model to learn is implied in the training features, during the learning

process we can compare the predicted values with its expected output. Doing this makes it possible to empirically establish the error of each hypothesis which acts as a performance measure of the model [18]. In this case we have the goal of finding a hypothesis which minimizes the empirical error for any examples, even for those not contained in our training data.

#### 2.2.1.1 K Nearest Neighbours

The K-Nearest Neighbours (K-NN) classification algorithm [18] might be a good option to be used in the context of this dissertation. This is the case because, this algorithm presents itself as an example of a lazy learning method rather than an eager learning one. Instead of the process where training data is used to fit a model and form a hypothesis on how the features present relate the models' predictions (eager learning), we present new instances and compare them directly with the ones in our training set (instance learning) [18]. This comparison between data points, in this context, can be seen as comparing a given vocalization to the ones already seen. As the patterns produced by vocalizations of a single species might be similar, this similarity would be conveyed by the K-NN algorithm.

K-NN labels new points based on the label of the majority of $K$ points nearest to the new point, being $K$ an uneven number from 1 to $K$. This will result in the creation of decision hyperplanes that will separate the classes based on the frontier in which the distance of two points of distinct classes is equal.

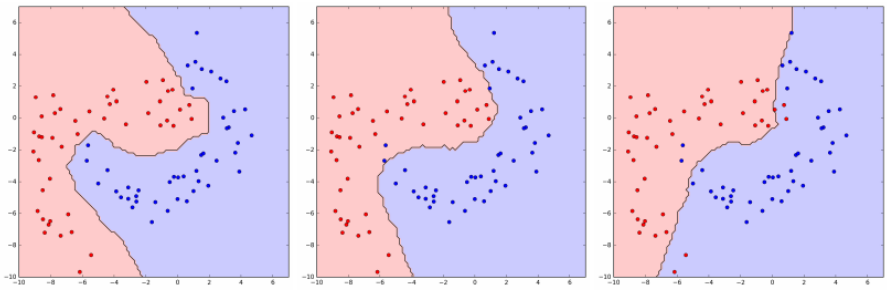


Figure 2.8: Comparison of the K-NN algorithm for different $K$ values (1, 13 and 25) [18].

As we can observe in Figure 2.8, we have to be careful when choosing the $K$ parameter, as it will influence how well our model will correctly label our new samples. If we have a small $K$ value, the model tends to overfit as the more data points we introduce, subsequent classifications will be easily influenced by past ones. When using a large $K$ value the opposite will happen, as we will incur in underfitting, as in general, less relevant (distant) points will have a bigger influence on the classification of new points which can lead to miss classifications.

Another aspect of the algorithm that needs to be considered is how the distance between points is calculated. Different distance calculation methods might be ideal for

some particular cases such the e *Minkowski distance* for continuous numerical features, or the *Hamming distance* for categorical features. Other widely used distances are the *Manhattan distance* and the *Euclidean distance*.

#### 2.2.1.2 Naive Bayes

Another classification algorithm to consider is the Naive Bayes classifier [18]. In spite the significance it imposes to the conditional independence between features, it might present satisfactory results in the multiclass classification problem that this dissertation deals with. Due to its availability and support among Python libraries, it presents itself as a fitting classifier to be tested. Although, when features are not independent, other classifiers are preferred in advance.

This algorithm is a probabilistic classifier based on the Bayes theorem, which as said, considers a conditional independence between features. This assumption is in place as it would not be feasible to compute the joint probability of every combination of features and classes which could be translated to a very large number of combinations.

In order to predict the class of a data instance, as with the Bayes classifier, we need to determine the maximum joint probability for each class for the features of the given data instance. The maximum join probability achieved amongst all classes will be the predicted class of that instance. Unlike the Bayes classifier the Naive Bayes classifier only needs to find the probability of each feature given the class.

In order train the classifier we need to determine the conditional probability distribution of each feature given each class. This will depend on if we are dealing with features which are continuous values or categorical values.

For the first case we could use a parametric model, where it is assumed that the features are normally distributed random variables when conditioned on the class, thus their probability distribution can be obtained by using a normal distribution [18]. An alternative to this would be the utilization of a non-parametric model like using a kernel density estimator, which determines the distribution of each feature by using histograms that depict the density of our data for a given feature in a given class. When dealing with categorical features it is recommended the utilization of histograms (non-parametric model), in order to derive a distribution of each feature from them, much like in kernel density estimation.

#### 2.2.1.3 Support Vector Machines

Support Vector Machines (SVM) [18, 33], like the Naive Bayes classifier has good availability and support among Python libraries. It can be seen as an appropriate classifier for the task at hand, as it is adequate to use in classification tasks while having great flexibility due to the use of different possible kernels (ex: Gaussian kernel), which can facilitate the classing of our vocalizations.

SVM when performing a classification task defines a hyperplane which separates the samples of different classes in our data sample. The hyperplane maximizes the distance of the two closest points from each class to the decision boundary [18, 33]. This maximum distance is the margin of the classifier and the points used to define it are represented in a vector form and defined as support vectors. This can be seen in Figure 2.9.

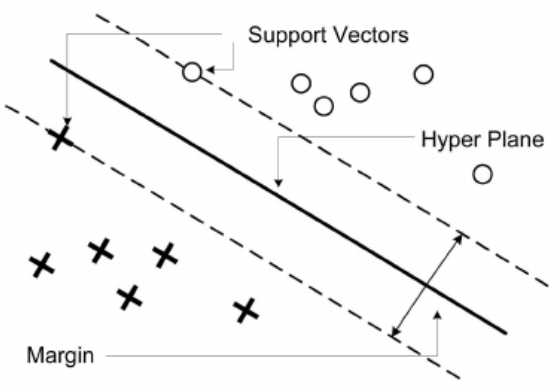



Figure 2.9: Representation of a support vector machine [34].

The goal of this classifier is the maximization of the margin. This ensures that for subsequent runs we will always have the hyperplane in the same position. In that case the loss function might not be affected by placing the hyperplane in a different position, which could result in placing it too close to some points, leading to overfitting. So, by maximizing the margin, we reduce overfitting as a result of constraining the frontier. In order to calculate the margin distance, so that it can then be maximized, we use the absolute distance between a given vector and the hyperplane.

This can be rewritten into a constraint optimization problem, that can be solved using Lagrange multipliers and thus defining a hyperplane which maximizes the maximum margin distance [18]. However, this will only be achievable if our classes are linearly separable. To tackle this, a slack variable is introduced to our constraint, which would allow vectors to penetrate the margins. This slack value represents the distance between the margins and the vector.

In spite of allowing the margins to be violated, the points that do so are given a penalty bounded by a regularization parameter $c$, with its value being directly proportional to the penalty weight.

This technique is called **Soft Margins** and is capable of solving slight overlaps in the data, however for sets whose data has bigger overlaps another technique is required [18]. To solve this, we could expand our feature space to a higher dimension where both classes could be linearly separable (Figure 2.10). We can achieve this by using a kernel function that computes the inner product between the feature vectors, which in turn can be used as an additional dimension of our feature vector.

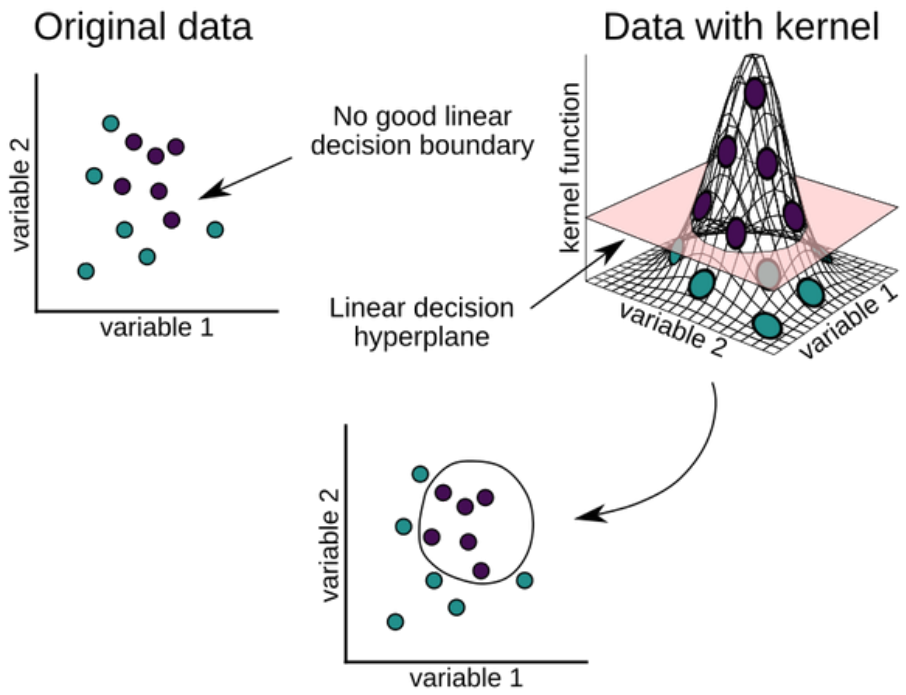



Figure 2.10: Utilization of a kernel function to establish decision frontier in a non-linearly separable data set [35].

### 2.2.2 Unsupervised learning

Due to the logistics concerning the capture of the vocalizations, most of the obtained data will be raw long duration recordings, which will need to be preprocessed and labeled to be used with supervised classifiers. However, unsupervised learning, which consists in the description of data based solely on its structure and features, does not require the use of labeled data [18], which may be useful in some future work related with this dissertation. Regarding this absence of class labels, a clustering phase can be implemented, where clusters for each of the existing classes will be built.

#### 2.2.2.1 Clustering

Clustering is a machine learning technique where we have the goal of dividing similar data points into groups (clusters), based on their similarities. Doing this will result in having data points that share the same proprieties or features in the same clusters, while the points dissimilarities are highlighted by belonging to different clusters [36]. Therefore, we can use clustering to group similar vocalizations into clusters, which in theory, would be comprised of vocalizations made from a single species. There are several different types of clustering algorithms. For the context of this work only a few will mentioned.

**K-means clustering** is a prototype based clustering approach and one more well known, and simple to implement clustering algorithms. Due to this, it provides a good

starting approach regarding clustering in a non supervised learning context. In short it divides the data in K clusters, each with a given prototype that corresponds to a mean vector of that cluster's members [18]. At first it establishes the prototypes randomly and in each round the closest point to each prototype will join its cluster, and then the prototype is recalculated. It performs this action until it converges or it reaches a stopping criteria. This algorithm has the drawback of needing to know a priori how many clusters exist or are desired, which is not suitable to every kind of problem. Another drawback of k-means and other is the tendency to produce n–dimension spherical clusters with similar volume, which do not match with real clusters in many cases, as it might happen in our problem [18].

**DBSCAN** [18, 37], is a density based clustering solution that solves the problem of irregular clusters, being a good alternative to k-means to be tested in this dissertation. Apart from this advantage, which can result in better classification performance, it does not require to know any information about the number of clusters a priori. It works by firstly selecting a random point in the data and assessing its neighbourhood. If there are at least $minPoints$ points at a distance closer than $\epsilon$, those points (and the first selected one) are introduced to the cluster, with the original point marked as visited. The algorithm will then repeat this process for the newly joined data points until all points are visited. In case a point does not meet the minimum number of neighbours required and it does not belong to any cluster, it will be considered noise (might be later introduced to a cluster). When a point that belongs to a cluster different than the one we are evaluating, is in its neighbourhood (with at least $minPoints$) both clusters are merged. When there are no more points to assess in a cluster, we sample an unvisited point and try to create a new cluster, this process will happen until all points are visited.

**Hierarchical clustering** is a clustering method that produces a series of nested clusters in a hierarchical fashion [18]. This method might be of interest for some future work regarding this dissertation, in order to correlate the obtained clusters of each vocalization with the taxonomic relationships of some cetacean species. This can be observed in the resulting dendrogram, which represents the successive similarity links established between the clusters. In order to establish a relation of similarity or dissimilarity between clusters (and samples) we need a metric that would represents these associations [18]. As a mean to compare examples we could express similarity/dissimilarity based on a distance metric between two points or by a normalized metric between 0 and 1. When dealing with similarity between clusters we use a given type of linkage such as complete linkage which measures the distance between the most distant points of each cluster [18]. In Hierarchical clustering, we have two standard approaches to perform the clustering: *Agglomerative clustering*, which is a bottom-up approach (from singleton clusters to one general cluster), or *Divisive clustering*, which is a top-down approach (several divisive iterations from a general cluster to smaller clusters).

### 2.2.3 Feature selection

When dealing with data that is comprised of many features, we have to take into account that most of them might not be desirable due to them being uninformative. Even if none of the features existent in our data is irrelevant, by having too many features we might incur in overfitting more easily [18].

Due to this it is important either in supervised and unsupervised learning to make a selection of the most important features, while discarding the rest. This can be done by examining each feature individually, **univariate filtering**, or by comparing with the other existing features, **multivariate filtering** [18].

One example of univariate filtering for supervised learning is the utilization of the $\chi^2$ test. This test is used to assess the statistical independence of a given feature to a class, as features that are statistically independent from the class will have a low $\chi^2$ value therefore are not desirable [18].

Another example, is the ANOVA F-test, which compares the variance of the feature values between groups, with the average variance within them [18]. Features that present similar values in different groups or features whose values deviate the most within a group are irrelevant, being represented by a low test value. The value resulting from this test represents a feature's discriminatory power regarding the available classes.

We can define a feature as redundant if it strongly correlates to other features. In order to observe these correlations, we can use some visualization methods like using parallel coordinates, frequency histograms, scatter matrix plots.

Another alternative is the utilization of Principal Component Analysis (PCA) in order to remove the redundancy in the features by finding a new low-dimension set of axes that summarize our data [38]. PCA takes into account the variance of each new produced feature, prioritizing features that present high variance as they are more likely to be able to distinguish different classes.

Independent component analysis (ICA), can be seen as a multipurpose statistical technique which allows for the extraction of maximally independent non-gaussian components that denote underlying factors in our data. This method relies on the idea that the observed data consists of a linear mixture of latent variables (independent components) where its mixing process is unknown. The standard use for this method is its application to solve the audio source separation problems, which consist in the separation of a sound mixture (such as the sound in a room full of people) into isolated sounds from individual sources (the sound each individual in the room produces). However, this technique can also be useful in other contexts such as the study of economic indicators or other type of data, as a way to better understand hidden aspects in the data structure. The independent components can be obtained by using the FastICA algorithm which relies on maximizing the non-Gaussianity of the data by maximizing its negentropy and kurtosis [39, 40].

### 2.2.4 Deep learning

Deep learning is a branch of machine learning which deals with algorithms that attempt to draw conclusions by continually analyzing data with a given logical structure. The algorithms usually have neural networks at its core. This consists in a structure of multiple layers, which try to learn representations of the data through hierarchical composition of relatively simple non-linear modules that transform features into progressively higher levels of abstraction [41]. These representations will then serve as input for the task we intent to accomplish (ex. classification).

Traditional machine learning algorithms such the ones presented above are flat algorithms, which means that they normally cannot be directly applied to raw data and must endure a pre-processing stage to retrieve its features [42]. Deep learning algorithms incorporate this phase within the algorithm itself, which can result in features that can better capture the structure of the data and that ultimately will result in a better performing algorithm. However, these algorithms are dependent on great volumes of data in order to be properly trained, which in some cases can be hard to gather.

#### 2.2.4.1 Convolution Neural Networks

Convolution Neural Networks (CNN) are an end-to-end deep neural network architecture with the premise of pattern identification by the successive application of convolution layers to its input. These layers extract and combine low-level features, such as edges and color, to more complex higher-level one's that will then be used to perform a classification task [3, 43]. Due to its popularity in computer vision, CNN are designed for working with two-dimensional image data, although they can be used with one-dimensional and three-dimensional inputs. These properties make CNN extremely interesting to use with the spectrograms of our vocalizations, mainly to test their capability to recognize their patterns and using the resulting features for our classification approaches.

In this explanation images are used as input. We can identify two distinct stages in a CNN model which are, the feature extraction/learning stage and the classification stage 2.11.

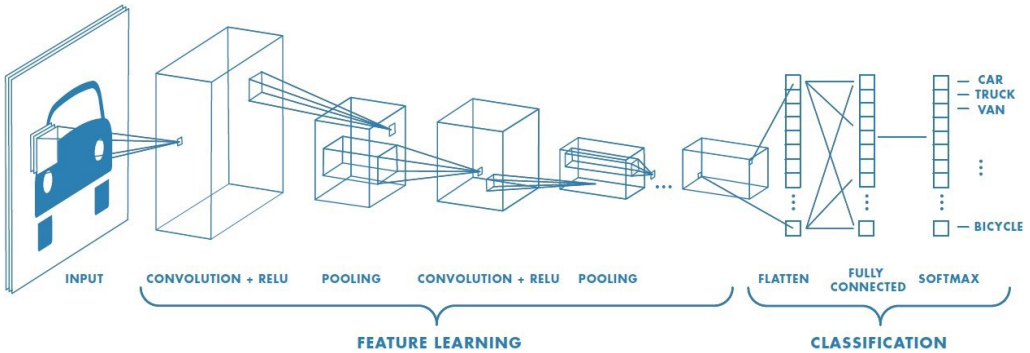



Figure 2.11: Schematic of an example of a CNN model [44].

This first stage, as briefly mentioned, is characterized by the reduction of the input to a form which is easier to process while not losing any important feature information. We achieve this by applying several successive convolution layers to the input. In these layers, convolution operation which consist in the multiplication of a set of weights to a given input, are done through the usage of a kernel/filter [45]. This filter is applied to each section of the input that it covers (receptive field), and performs a matrix multiplication between itself and this subsection of the input. The filter will move according to a stride value, which corresponds to the number of pixels shifts over the input matrix. Once the input is completely transversed, from these convolutions we obtain a single feature map. The resulting feature map will be of a smaller dimensionality than the original input unless some padding was applied to it during the convolution process, which would result in an equal or higher dimensionality matrix [44]. The convolution process can be seen in Figure 2.12.

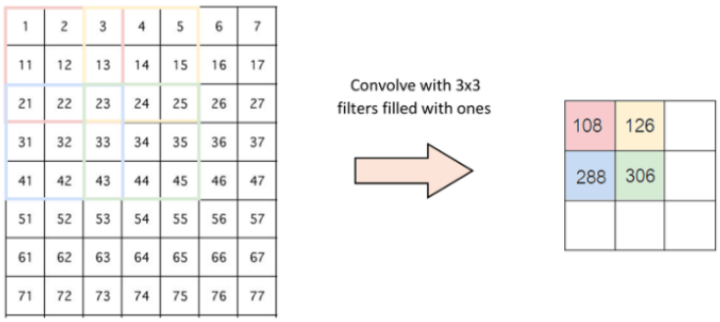



Figure 2.12: Convolution of 8x7 input with 3x3 kernel (highlighted on the left image, by a 3x3 sub matrix in color) and a stride of two pixels [46].

CNN architectures also employ the usage of pooling layers which reduce the dimensions of the convoluted features even further. This can be done with a type of pooling such as Max pooling, which returns the maximum value from the portion of the image covered by the Kernel. This pooling type in particular can also act as a noise suppressant [44].

CNN's use several filters in parallel for a given input resulting in different "specialized"features for the same input, which in turn will provide a more complete and diverse final set of features to use during the classification phase. Also, by subsequently taking an extracted feature map from one convolution layer to be the input of the next, we are allowing a hierarchical decomposition of the input [45]. This in turn will result in the deeper layers of the network extracting higher-level features that will help in the next stage.

The second stage, which is the classification stage takes the flattened extracted features from the model and feeds them to a feed-forward neural network. This Artificial Neural Network (ANN) will be trained using backpropagation providing feedback by evaluating the error between predictions and ground truth, enabling the balancing of the neurons' weights. Depending on the type of problem the CNN is trying to solve, a different

activation function on the last layer will be required, which can be a softmax activation (n-ary classification problem), a sigmoid activation (binary classification) or none in case of a regression problem. In the case of a classification problem, the model outputs predictive probability vector representing the probability of the input belonging to any of our target classes.

# 3

# Related work

This chapter presents the related work literature that was surveyed in order to have a clearer picture of the task at hand. We start by highlighting past approaches that were employed in solving problems of bioacoustic classification (Section 3.1), introducing some key challenges that come with it. Due to the high change of working with noisy signals, we will also touch on some past work which tackles this problem (Section 3.2).

## 3.1 Bioacoustic classification

Bioacoustics is an area of scientific study which crosses the study of biology and acoustics. To be more precise, it studies how humans and other animals use sound and acoustical perception, while trying to understand how the various acoustical adaptations reflect their relationships with their habitat and surroundings.

One common task researchers in this field perform is the categorization of vocalizations produced by a certain species of animals [3, 9, 15, 47, 48]. Sometimes even going as far as identifying which species it originated from [8, 49–57]. For the past decades there have been numerous studies which, with the aid of machine learning, accomplish this.

In 2010, Bahoura and Simard [9], used different feature characterization methods based on the STFT and the Wavelet Packet Transform (WPT), to classify distinct blue whale call types while using a Multilayer Perceptron (MLP) as a classifier. This resulted in an average accuracy of 86%, with STFT achieving a slightly better performance than WPT. In order to perform a similar task in real time but on Beluga whales, Miralles-Ricós et al. [47] proposed a new model which deviated from the typical approaches based on pattern recognition of time frequency representations of cetaceous sound [9]. Instead, they used 15 hand selected features which summarized the resonant frequencies and bandwidths of beluga multi-tonal vocalizations and other high order statistical information. This choice of features reduces the computational cost, facilitating real time processing. In conjunction with a Naive Bayes classifier they achieved an overall correct classification percentage of 88.3%.

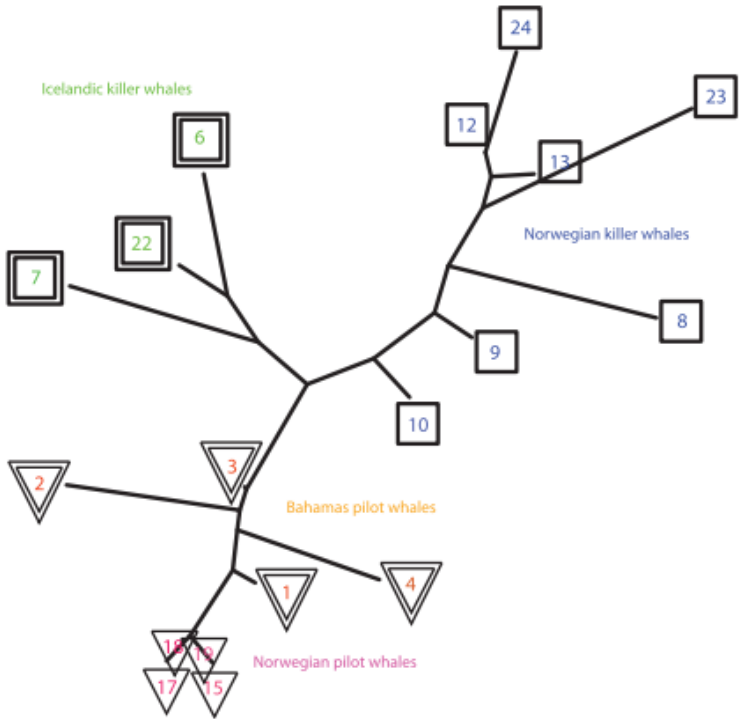



Figure 3.1: Evolutionary tree generated of resulting clustering algorithm [8].

Shamir et al. [8] address a different task, the classification of a species given a vocalization. This research presents a model to classify the similarities between calls of killer and pilot whales (from different regions) in both a supervised and unsupervised fashion. This is accomplished by firstly assigning a Fisher discriminant score to content descriptors, which depict the input 2D spectrograms in a numerical fashion. The top 15% of features with the highest score are considered the most informative and will be used to construct a feature vector, which in turn will be used to compare samples. The similarity between each pair of whale calls can be estimated by the weighted distance between two feature vectors. The supervised approach had the objective of distinguishing between both species of whales. By comparing the similarities in their calls it achieved an accuracy of 92%. The clustering algorithm (unsupervised) with this similarity criteria achieved a classification accuracy ranging from 44% to 62% in determining sub-groups within regional samples of a specimen (pods), which is not perfect but better than random guessing. By observing the evolutionary tree 3.1 we can observe the resulting cluster showing the similarities between pods of each specimen.

Also within the realm of cetacean interspecies bioacoustic classification, several other works stand out. In 2010, Baumann-Pickering et al. [51] performed a discriminative analysis of the echolocalization clicks of three distinct cetacean species (melon-headed whales, bottlenose dolphins and gray spinner dolphins). By performing a statistical analysis of the frequency aspects of each species, several descriptors were obtained which

were then used in conjunction with a discriminant function analysis to determine their efficacy. This method proved itself useful in distinguishing the melon headed whale clicks, having an accuracy ranging from 87% to 93%. However, for the bottlenose and gray's spinner dolphins only accuracies ranging from 34% to 54% and 54% to 75% were obtained respectively, highlighting the difficulty of the problem of dolphin vocalization distinction. More recently, a similar approach was used by Amorim et al. [57] which also obtained vocalization whistle and echolocation features of several odontocete species by performing a statistical analysis of the vocalizations. Similarly as a classification approach a discriminant function analysis was also performed obtaining results ranging from 86.3% to 100%, however the researches highlight the fact that the used limited data might have contributed to these results.

In 2013, Gillespie et al. [52] also performed the detection and classification of whistles of several odontocete species. In their work the vocalization contours were first extracted from spectrograms and then underwent a statistical parameter extraction stage, from where 9 parameters describing the vocalizations contours were obtained (mean frequency, slope over time, curvature...). During the classification stage, the vocalizations were divided into 4 different datasets regarding their species natural habitat: Polar Atlantic (4 species), Atlantic Frontier (8 species), Gulf of Mexico (11 species) and Tropical Atlantic (12 species). The classification was performed using the whistle classifier available in the PAMguard [58] passive acoustic monitoring software. In spite of a good mean correct classification rate for the species in the Polar Atlantic (94.5%), the remaining results provided worse results: 67.5% in the Atlantic Frontier, 60.6% in the Gulf of Mexico and 58.6% in the Tropical Atlantic. Similarly, Erbs et al. [56] also made use of the PAMguard software to detect and classify vocalizations of three dolphin species (common dolphin, bottlenose dolphin, Indo-Pacific bottlenose dolphin) obtaining a mean correct classification rate of 87.3% and 78.4% when additionally trying to distinguish two different populations of Indo-Pacific bottlenose dolphins.

More recently in 2019, Nadir et al. [55] used a different approach based on using features obtained by the fusion of 1D Local binary patterns (1D-LBP) and MFCC to classify 6 different species of odontocetes (5 dolphin species and a melon head whale). This approach used SVM as a classifier which was trained using 5 fold cross-validation. The obtained results were satisfactory, with a general model accuracy of 89.6% and individual species acuraccies ranging from 74% (spinner dolphin) to 100% (bottlenose dolphin). However these also correspond to the species with the most amount of recordings, 114 samples for spinner dolphins and 24 for bottlenose dolphins, which raises the same concerns of a limited dataset as the work of Amorim et al. [57].

In spite the works being presented until now only refer to the classification of cetacean species, the area of bioacoustic classification spans a broader scope. An example of this is the work of Noda et al. [49] which performed inter species classification, more specifically between anurans and reptile species. This study uses as features MFCC ( 2.1.3), Linear Cepstral Coefficients (LFCC) (provide better resolution at higher frequencies) and a fusion

of both MFCC and LFCC in a feature matrix. For the classification stage, three algorithms were compared: K-NN, SVM and random forests. On average the obtained accuracy was 95% for the anurans species and 98% for the reptile species, with the combination of both MFCC and LFCC features along with SVM classifier providing the best results. The combination of both MFCC and LFCC provides a clear characterization of high and low frequency components, being more robust against noise which can explain the high classification accuracy.

In recent years, due to the advancements in deep learning there has been a trend in applying it to solve some of the problems mentioned above. Zhang et al. [48], presented a comparative study of two different pre-trained convolution neural network models (ResNext101 and Xception) with two different types of input features, a 1D waveform input and a 2D log-mel spectrogram. The generated feature maps then serve as input to a fully connected layer with a softmax activation in order to perform a classification task. This study uses the same data as the one by Shamir et al. which was mentioned above [8]. Besides performing the same tasks as Shamir et al., they also carry out an additional one which had the intent to separate each species into its 4 regional groups. In every task all CNN models tested outperformed the solution proposed in [8], with the better performing combination being the ResNext101 with the 2D log mel spectrogram, which achieved accuracies of 99.7% (binary classification of species), 99.2% (separation in regional groups) and 97.6% (pod classification). It is of importance to note that the 1D waveform input also produced satisfactory results (99.5%, 97.7%, 91.8% in the correspondent tasks).

Other examples of usage of CNN's in bioacoustic problems are present by Bermant et al. [15], where it is used to detect whale echolocation clicks in spectrogram images (time frequency-domain). CNN's do not always need to be applied in an end-to-end way as showed by Dorian et al. in [43]. In their work, a pre-trained CNN's is only used in order to obtain the feature maps of the input spectrograms, with the classification task being performed by a traditional SVM algorithm. This study applied this model to determine how environmental noise (such as noise from boats, rain, etc...) affect whale call detection.

## 3.2 Denoising of signals

As mentioned above (in Subsection 2.1.4) when dealing with audio recordings it is likely to encounter some form of noise. If we have prior knowledge of which is the frequency range of our target vocalization we can apply a high-pass filter, like Miralles-Ricós et al. [47] did to remove low frequency interference noise outside the beluga whale frequency range.

Both Priyadarshani et al. in [59] and Veeraiyan in [60] present several methods for denoising underwater acoustic signals. From these methods it is possible to highlight the utilization of the Wavelet Transform (WT) , which allows the usage of different sized time windows for dissimilar bands of frequency unlike what is done with the STFT method.

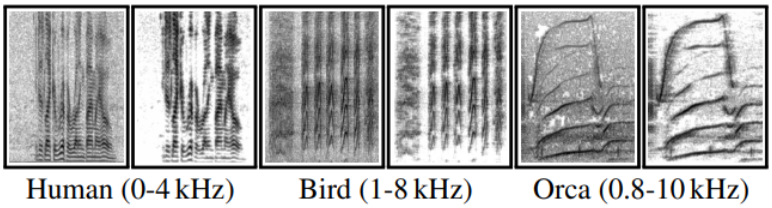



Figure 3.2: Original versus denoised spectrograms that resulted from OrcaClean [12].

This in turn provides a higher resolution for lower frequencies and lower resolution for higher ones, making it ideal to model non-linear signals. The WT condenses the signal's features in wavelet coefficients. By applying a thresholding function to the coefficients, the ones that represent noise (which typically have lower values) will have a lower weight, possibly close to zero. After that the signal can be reconstructed by using an inverse wavelet transform, which in turn will return the denoised signal. This method was used by Priyadarshani et al. [61] in order to denoise signals of bird calls in conjunction with band-pass or low-pass filtering, successfully managing to wash out stationary noise from the recordings without distorting the bird calls.

Bergler et al. [12] propose an alternative way to denoise a signal with the aid of CNN's without requiring any clean ground-truth. Here, a deep denoising network is trained with orca vocalization spectrograms injected with several additive noise variants, in order to produce a noise-free representation. To help remove as much noise as possible without compromising the vocalization quality, an attention mechanism in the form of binary masks is introduced to distinguish the signal from the noise. By doing this the model will more accurately distinguish spectrally strong and weak signal regions, assuming that the emitted orca calls have stronger spectral intensities. This solution was tested for cross-domain generalization and transferability with positive results, experimenting on also human voices and bird calls which can be seen in Figure 3.2. By observing the resulting spectrograms, we can see that the signal representation was preserved and the noise almost completely removed.

# 4

# Technical Approach

This chapter highlights the proposed technical approach used to accomplish the task of bioacoustic classification proposed in this dissertation. Here, we will go through each one of the stages in the proposed method shown in Figure 4.1. First, all of the sourced data for this work is presented (Section 4.1), followed by the preprocessing tasks to be performed over it, which make up the first stage of the method (Section. 4.2). Following this, the used time-frequency representations on the second stage are highlighted (Section.4.3). These representations will be the foundation to obtain our features in the feature extraction stage in Section 4.4 (third stage). Finally in Section 4.5 (fourth stage), we will delve down into the details of the training and classification stages of the work.

## 4.1 Data

The initial objective of this dissertation aimed at the distinction of a wide variety of cetacean species existent in the Madeira archipelago by using bioacoustic classification models. For this purpose, recordings of 14 different species (4 belonging to Mysticeti and 11 to Odontoceti) which have been reported in the Madeira archipelago [16] were initially sourced. These recordings were obtained either from the Watkins Marine Mammal Sound Database (WMMSD) issued by the Woods Hole Oceanographic Institution [62] or from recordings made by the Madeira Whale Museum (MWM) scientific team for the purpose of this dissertation. All of the compiled recordings can be seen in Table 4.1.

The recordings conducted by the MWM were carried out in dedicated boat surveys while using PAM techniques. Whenever there was visual confirmation of a species of interest and the weather conditions were favorable, a compact self-contained underwater sound recorder (SoundTrap 300 series, model HF, recording the 20 Hz to 150 kHz bandwidth) was deployed at a depth of 10 m in continuous recording mode. In order to minimize the external environmental noise caused by the vertical movement of the recording device when submerged, a group of buoys connected by an elastic rope to the recorder were used, which counteracted this movement. During the recording process the boat kept its engine off while at a distance of 100 m of the device.

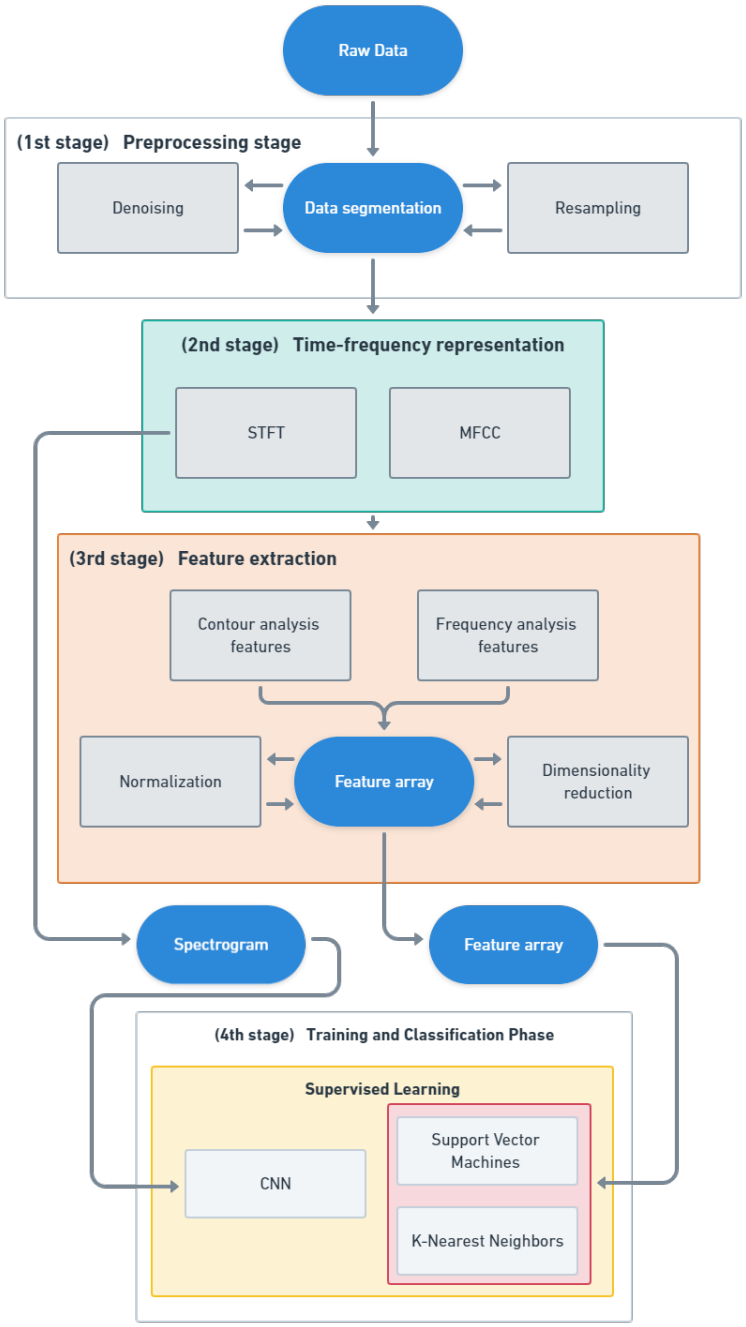



Figure 4.1: Pipeline of the proposed four stage method.

In spite of having a diverse set of recordings containing several different cetacean species, the objective of the dissertation shifted towards performing the distinction between only four different small dolphin species (common dolphin: *delphinus delphis*; bottlenose dolphin: *tursiops truncatus*; atlantic spotted dolphin: *stenella frontalis*; striped dolphin: *stenella coeruleoalba*). This was due to **three main factors**. The first **(1)** being the recordings sampling rate. Since most Mysticeti species predominantly produce low

Table 4.1: Compiled cetacean vocalization data. The first four species correspond to Mysticetes, while the remaining species are Odontocetes. The highlighted species correspond to the four species selected for the final dataset. All data was sourced from the Watkins Marine Mammal Sound Database[†] and from recordings provided by the Madeira Whale Museum[⋆].

| Species | nº recs | min len (sec) | max len (sec) | min SR (Hz) |
|---|---|---|---|---|
| Northern Right Whale[†] | 54 | 1 | 5 | 5120 |
| Fin, Finback Whale[†] | 54 | 5 | 101 | 600 |
| Humpback Whale[†] | 64 | 2 | 709 | 5120 |
| Minke Whale[†] | 17 | 1 | 2 | 1280 |
| Sperm Whale[†] | 76 | 2 | 1260 | 40000 |
| Orca Whale[†] | 35 | 2 | 15 | 20480 |
| Short-Finned Pilot Whale[†⋆] | 66 | 1 | 1260 | 30000 |
| False Orca Whale[†] | 59 | 1 | 6 | 30000 |
| Grampus, Risso's Dolphin[†] | 67 | 1 | 36 | 30000 |
| Bottlenose Dolphin[†⋆] | 25 | 1 | 1135 | 40000 |
| Rough-Toothed Dolphin[†] | 50 | 1 | 2 | 81920 |
| Common Dolphin[†⋆] | 54 | 1 | 991 | 43900 |
| Striped Dolphin[†] | 81 | 1 | 25 | 60600 |
| Atlantic Spotted Dolphin[†⋆] | 55 | 1 | 536 | 43900 |

*contains recordings provided by the Madeira Whale Museum

frequency vocalizations, these are usually recorded using small sampling rate values. For example, Table 4.1 shows that the available recordings for the finback whale have a sample rate of just 600 Hz, and the recordings of the other Mysticeti species have sampling rates equal or lower than 5120 Hz. This limits our frequency comparison scope, as we can only compare vocalizations up to a frequency corresponding to half of the used sampling rate (Nyquist Theorem [63]). Due to this scale difference to other species, the Mysticeti recordings were discarded. The other factors include **(2)** the lack of recordings for some other Odontocete species and ultimately **(3)** the desire of the MWM to focus on the distinction of species which are more commonly sighted in the archipelago while also being harder to distinguish by their vocalizations, culminated in the selection of these four species.

As a way to reduce location and recording bias during the classification process, at least two distinct recording locations for each of the species present in the dataset are ensured. This premise is guaranteed at the start by the usage of data contained in the WMMSD, which for each of the selected species provides recordings that span multiple locations, that will then be complemented by the MWM data.

## 4.2 Data preprocessing

Having compiled the species dataset to be used in this dissertation, the next task to undergo aimed at transforming the raw recorded vocalizations into equal data points from where to extract features from. This is done by obtaining equal length and sampled recording slices from the original recordings. These tasks make up the first stage of our proposed method depicted in Figure 1.2. Each of its individual tasks are represented in Figure 4.2.

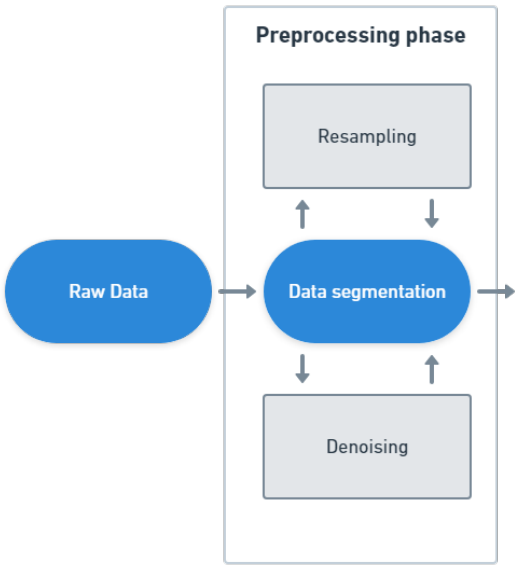



Figure 4.2: Preprocessing stage tasks.

### 4.2.1 Data segmentation

This is the first task at hand, which consists in obtaining several data points from the raw data, to be able to train the classification models down the line. To do this, a segmentation of the raw recordings into smaller 1 second slices is performed, while providing an adequate species label for further training and validation. This slicing process was carried out with the aid of Phyton modules in the form of the AudioSegment module from Pydub.

To ensure that all the segments contain part of a vocalization, an empirical observation of the corresponding audio slice spectrograms was made, which resulted in the rejection of segments which only contained background noise. This segmentation of the recordings achieved the final dataset to be used in this dissertation, which is comprised by 910 one-second recording samples and can be seen in Table 4.2. This analysis was facilitated by the usage of the audio editing software "Audacity"(Figure 4.3).

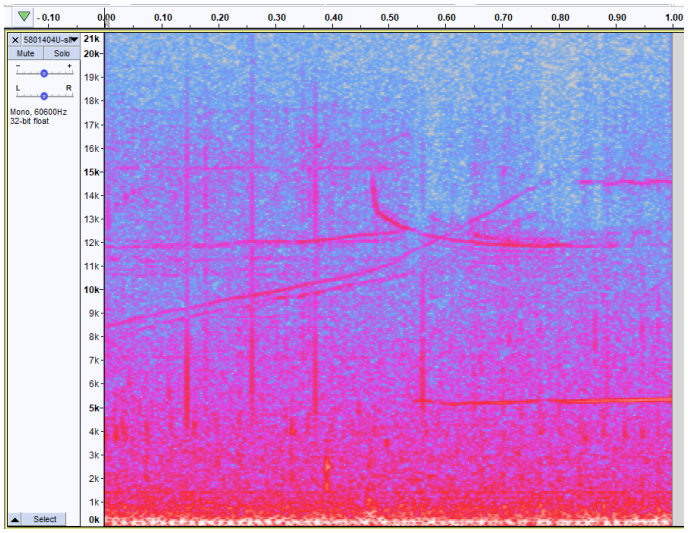



Figure 4.3: Spectrogram analysis of a recording slice of a Bottlenose dolphin/*Tursiops truncatus* vocalizations in Audacity

.

Table 4.2: Final dataset of the obtained recording samples after the segmentation was performed.

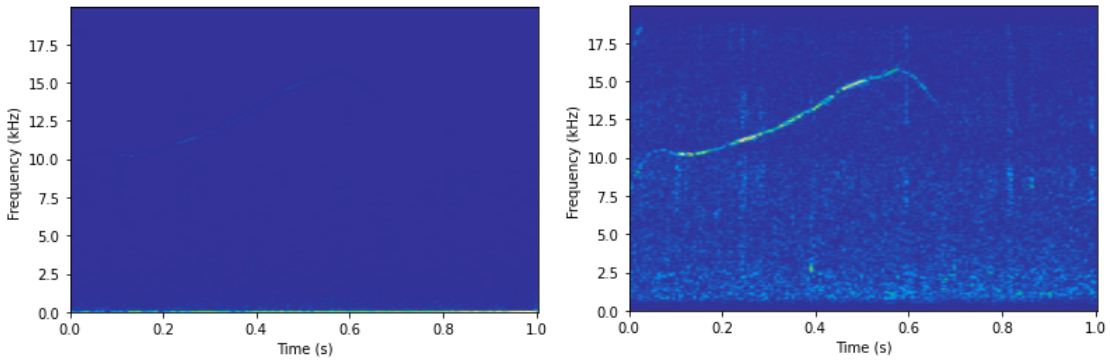
| Species | WMMSD | MWM | Total |
|---|---|---|---|
| Common dolphin/*delphinus delphis (Dd)* | 238 | 164 | 402 |
| Atlantic spotted dolphin/*stenella frontalis (Sf)* | 165 | 31 | 196 |
| Bottlenose dolphin/*tursiops truncatus (Tt)* | 42 | 136 | 178 |
| Striped dolphin/*stenella coeruleoalba (Sc)* | 134 | 0 | 134 |

### 4.2.2 Denoising

As there is some degree of noise in our recordings, a denoising task must be performed as a way to enhance the quality of our data. As mentioned above, several methods can be applied to do this. Depending on the type of vocalization and amount of noise, we can employ simple denoising filters (Section 2.1.4). This solution can be effective if the noise is not present in the same frequency bands as the vocalizations, which happens to be the case as the used recordings mostly have a constant low frequency noise in some examples.

Taking this into account, to each of the recordings to be used, a 4th order Butterworth high-pass filter was applied with a cutoff frequency of 1000 Hz. By doing this, the intensity of the frequency bands above 1000 Hz is enhanced while and the the ones bellow are attenuated. Due to the wide frequency range the vocalizations of the species in the dataset can reach, a higher cutoff value was not used, as it could have withhold relevant information. The application of this denoising filter was made possible by using the signal processing package from the scipy module.

The effect of this chosen approach can be seen in Figure 4.4. Here, two spectrograms




(a) (b)

Figure 4.4: Spectrograms of a vocalization of a bottlenose dolphin with background noise before (a) and after (b) applying a high-pass filter with cutoff frequency of 1000 Hz.

of a vocalization of a bottlenose dolphin/*tursiops truncatus* are shown before and after applying the high-pass filter. It is possible to observe that the first spectrogram (a) is dominated by the high intensity of the low frequency noise bellow 1000 Hz (near the bottom of the spectrogram). This makes it very difficult to distinguish any vocalization pattern existent in the recording, with only a faint contour being visible around 10000 Hz. However, we can see that by applying the high pass filter in spectrogram (b) a vocalization pattern is now visible between the 10000 Hz to 15000 Hz frequency bands. This highlights the viability of this approach to not only reduce low frequency noise but also to enhance the vocalization signal and will be used throughout this work.

### 4.2.3 Resampling

As alluded to previously in Section 4.1, the sampling rate of a given recording is directly tied to the maximum recorded frequency by the Nyquist Theorem, where it corresponds to half of the used sampling rate. Knowing this, it is easy to understand that a recording which used a higher sampling rate will carry more information than one which used a lower sampling rate. This causes an issue when loading recordings into data structures, as recordings with different sampling rates but with equal time length, would not produce data structures of equal size, which rises the need to resample the used dataset to a common sampling rate.

Taking into account the minimum common sampling rate among the selected species in Table 4.1, our recordings were downsampled to 40 kHz. This sampling rate allows to express vocalizations up to 20 kHz, which for the selected species is enough to cover most of the frequency bandwidth of the species social calls. This can be seen by examining Table 4.3 which shows the minimum and maximum produced frequencies by a given species based on some previous work on whistle analysis. It must be noted that these values do not account for narrowband high frequency clicks that most of these species perform, which can reach frequencies above 100 kHz, as the focus of this work lies on

mostly whistle recordings which are deemed to be more informative.

Table 4.3: Whistle frequency bandwidth of four dolphin species detected in previous works.

| Species | Min Frequency (kHz) | Max Frequency (kHz) | Refs |
|---|---|---|---|
| Common dolphin | 2 | 20.21 | [64, 65] |
| Atlantic spotted dolphin | 5.24 | 23.44 | [65, 66] |
| Striped dolphin | 7.11 | 20.47 | [65] |
| Bottlenose dolphin | 7 | 15 | [67] |

## 4.3 Time-frequency representation

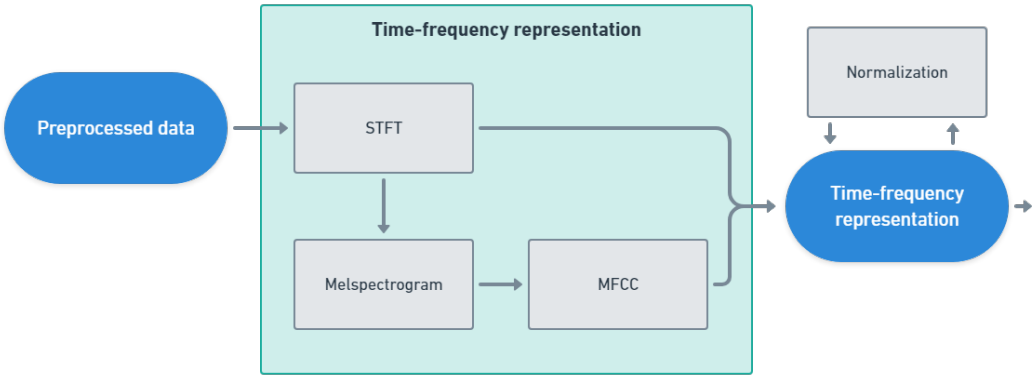



Figure 4.5: Time-frequency extraction process.

In order to extract features from each of the audio recordings, they must first be converted to a time-frequency representation, from where the features will be derived. In this work two different approaches for this representation were used: the first uses the magnitude spectrogram of the STFT, while the second uses the MFCC of the signal.

For both of these approaches, a windowing and segmentation of the signal must be performed which requires to select both a window and frame size. The window size is essentially the amount of samples to be windowed at a segment, whereas the frame size corresponds to the number of frames contained in a segment to which the FFT will be applied (Figure 4.6). Generally both of these values are the same, however the frame size can be bigger than the window size, which in this case the excess frames would be zero padded.

Another particularity of these parameters is their influence in both the frequency and time resolution of the resulting time-frequency representation. This correlation can be observed by looking at the shapes of the resulting spectral representations in Table 4.4. These highlight the trade-off between having a better frequency resolution (higher number of frequency bins) by increasing the frame size or a better time resolution (more time frames) by reducing the window size.

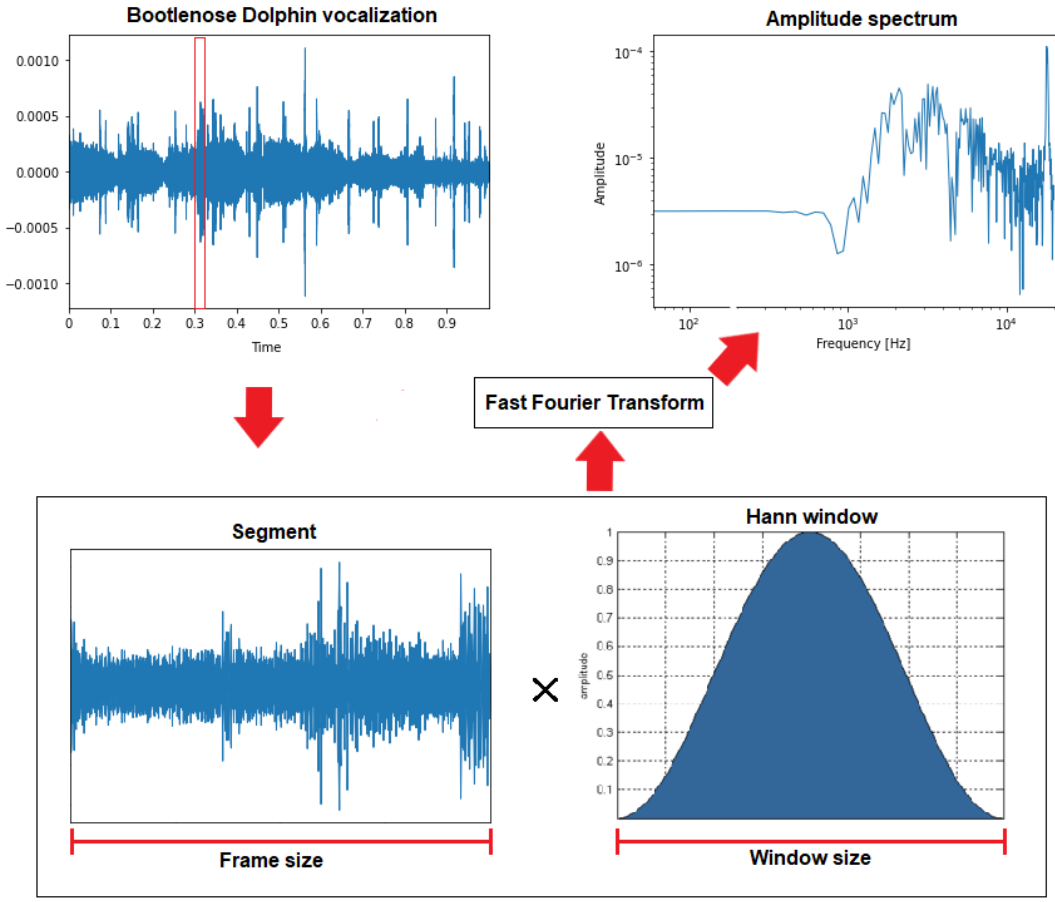



Figure 4.6: Visualization of the segmentation and windowing of part of a vocalization.

Due to this, we intend to test three distinct combinations of frame and window sizes which tend to favour better frequency resolutions without a showing a significant decline in the temporal resolution of the vocalizations. This choice allows us to mainly focus on the extraction of detailed frequency modulation features, while still having a significant temporal context in the time-frequency representation to allow the analysis of the vocalization contours. The tested combinations can be seen in Table 4.4. Additionally in Figure 4.7, the magnitude spectrograms obtained with these STFT parameter combinations can also be seen.

Table 4.4: Tested parameter combinations of frame and window sizes to be tested.

| Frame size | Window size | Shape (Freq, time) |
|---|---|---|
| 512 | 256 | (257, 314) |
| 512 | 512 | (257, 158) |
| 1024 | 512 | (513, 158) |

Another variable that also needs mentioning is the number of extracted MFCC. As mentioned in Section 2.1.3, these highlight the rate of change in the Mel-frequency bands

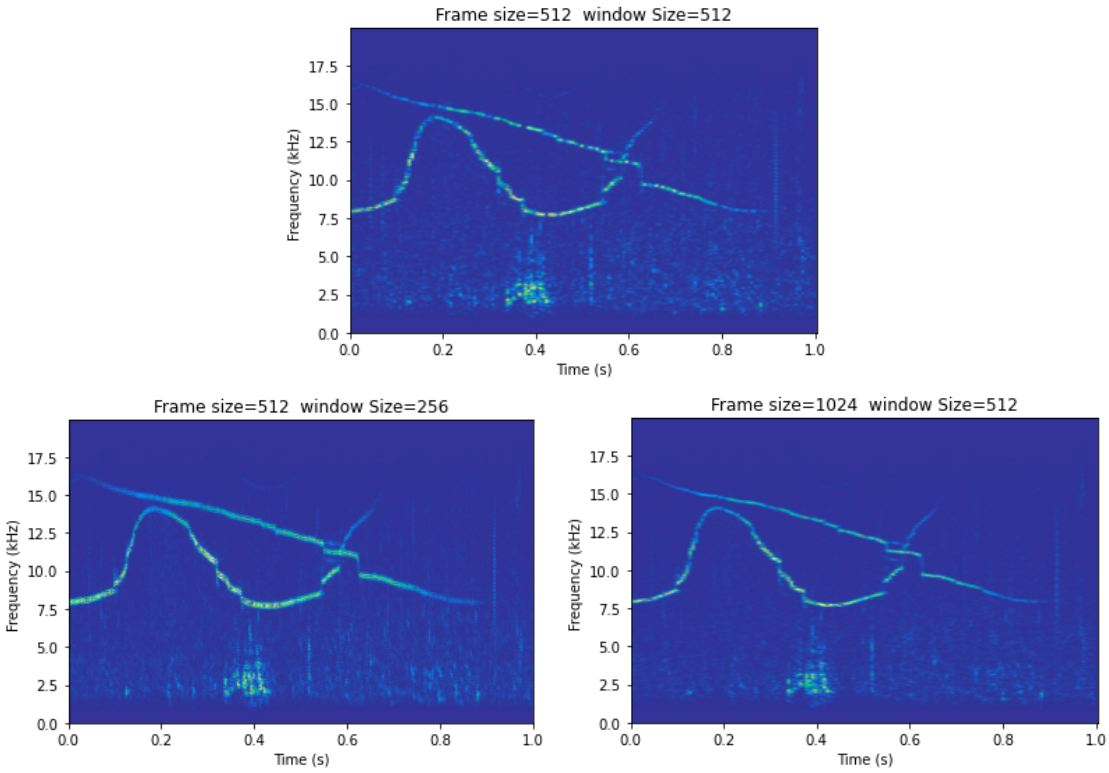



Figure 4.7: Spectrograms of striped dolphin vocalization obtained by using the tested combinations of frame size and window size parameters when performing the STFT.

which are denoted by the number of used filters in the Mel filter bank. Due to the logarithmic nature of the Mel scale which mimics the perceived pitch by the human ear, as we move along the frequency axis the less definition we will get, as the filters in the filter bank become broader and more isolated as we can see in Figure 2.6.

In the context of this work this might be an issue, as we intend to maintain as much definition on the higher frequencies as possible due to the high probability of it containing useful information (vocalizations). With this in mind, Mel filter banks with different amounts of filters were tested which enables the extraction of different MFCC values corresponding to the amount of filters used (20, 40, 60 or 120). The used filters can be seen in Figure 4.8. These filters were used to obtain Melspectrograms of each of the vocalizations from where the discrete cosine transform is applied (DCT) allowing the extraction of the MFCC. After obtaining any of spectral-representations, a min-max normalization is performed in order to have every data point within the same scale.

## 4.4 Feature extraction

Feature extraction is one of the most important tasks to be performed during the scope of this work. This is the case as the obtained features will be the sole representants of the vocalizations used to classify the intended species. In other words, the used features

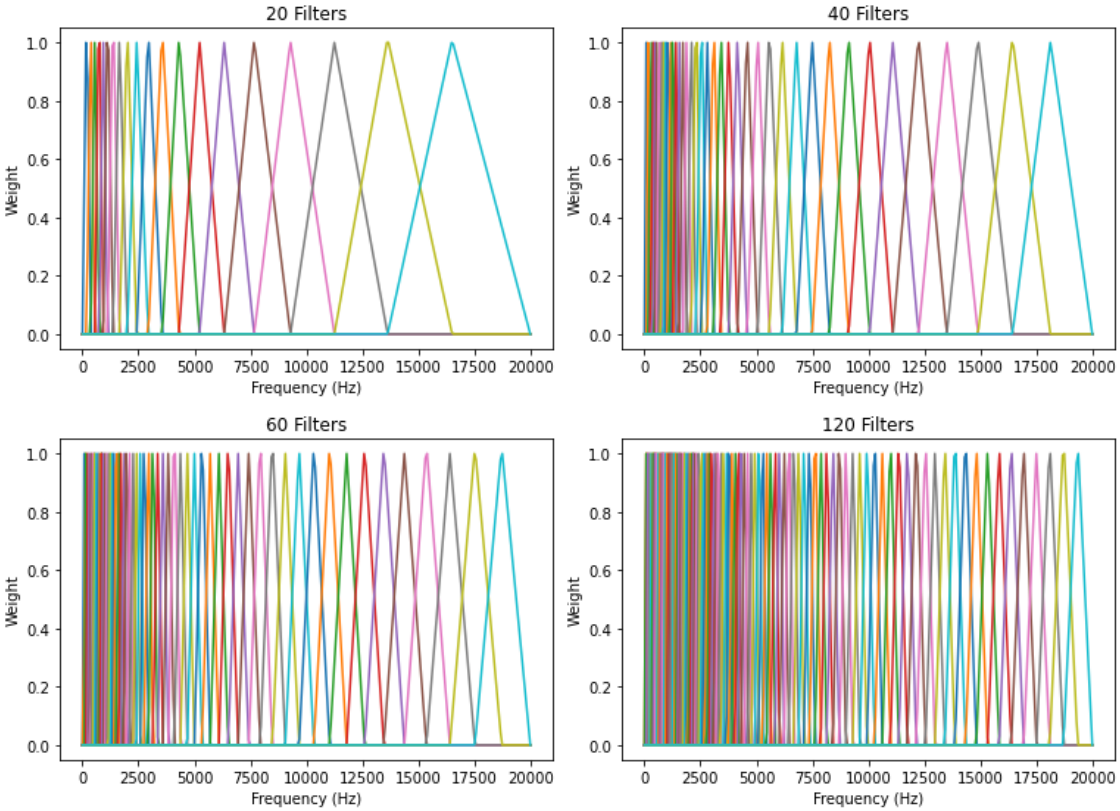



Figure 4.8: Tested mel filter banks in the construction of the melspectrogram used to derive the MFCC. It is possible to observe an increase in filter density on the higher frequencies as the number of filters increases in the filter bank.

will have a direct impact in the correctness of our predictions. The proposed features in this work encompass two distinct approaches to the analysis of the vocalizations which can be seen in Figure 4.9. These features will be obtained from any of the presented time-frequency representations. Due to the possibility of using either STFT or MFCC representations, we will refer to the frequency bins in the spectrogram and the coefficients in the MFCC matrix as frequency components on the following sections. After obtaining each feature subset as our resulting features span different values of magnitude, they are normalized to become equally weighted and then they are subjected to a dimensionality reduction process (Section 4.4.3).

### 4.4.1 Frequency analysis features

The first feature subset ($Fs_1$) culminates in the creation of three distinct **frequency analysis features**, which are the *Frequency component's magnitude sum*, the *Variation coefficient* and the *Magnitude variation*. These features try to capture the predominant frequency components of any call, which can be a good indicator of the frequency distribution and range of a species' vocalizations.

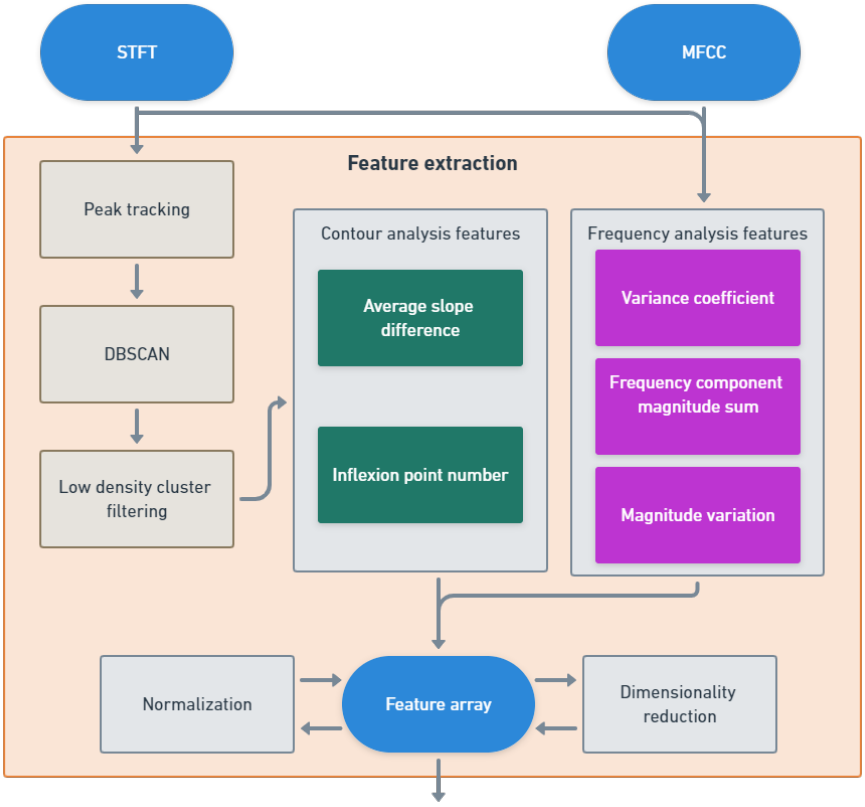



Figure 4.9: Feature extraction process.

#### 4.4.1.1 Frequency component magnitude sum

The first feature in this subset corresponds to each **frequency component's magnitude sum** $MS_{fc}(f)$, where each entry is given by:

$$MS_{fc}(f) = \sum_{t=0}^{\|T\|} m(f,t) \tag{4.1}$$

where the magnitude sum of the frequency component $f$ is given by the cumulative sum of the magnitudes $m$ on every time frame $t$ of the recording sample. $\|T\|$ is the number of time frames $t$. This feature provides the essential information regarding the predominant frequency components where vocalizations may lie, as the magnitude on those instances will contribute greatly to the magnitude sum of its frequency components, and paint an overall picture of the vocalization frequency range.

A distinct approach to calculate the frequency component's magnitude sum was also tested. This approach only took into account time frames in each frequency component which were considered magnitude peaks by the peak tracking technique detailed in Section 4.4.2. In other words, to estimate the feature we only take into consideration instances in the spectral representation which correspond to a vocalization contour. However, this approach as we will see in Section 5.1 hindered the classification power of this

feature and so was discarded.

#### 4.4.1.2 Variation coefficient

The following feature is the **Variation coefficient** of a signal's spectral representation. In order to derive this metric, we firstly need to determine the average frequency component magnitude sum ($\overline{MS_{fc}}$), which is achieved by the following expression:

$$\overline{MS_{fc}} = \frac{\sum_{f \in F} MS_{fc}(f)}{\|F\|} \tag{4.2}$$

with $\|F\|$ corresponding to the number of frequency components $f$. The average frequency component magnitude sum allows to estimate the standard deviation of the overall frequency component magnitude sum of the signal, and consequently the $VarCoef$ feature:

$$VarCoef = \frac{\sqrt{\frac{1}{\|F\|} \cdot [\sum_{f \in F} [MS_{fc}(f) - \overline{MS_{fc}}]^2]}}{\overline{MS_{fc}}} \tag{4.3}$$

This feature assesses the relative variation of the magnitude along different frequency components. This may be of interest to help discriminate different species which may possess different degrees of magnitude along any of the frequency bands. For example, two species, one vocalizing with a strong dynamics of magnitude along different frequency components, and other vocalizing similarly in magnitude for all frequency components, will be distinguished by this feature.

#### 4.4.1.3 Magnitude variation

The final feature in this subset intends to highlight the average difference between two consecutive frequency component maximum magnitudes. This is achieved by the following expression:

$$Mvar = \frac{\sum_{f=0}^{\|F\|-1} |\max_t(m(f,t)) - \max_t(m(f+1,t))|}{\|F\| - 1} \tag{4.4}$$

where $m(f,t)$ corresponds to the magnitude of frequency component $f$ on time slice $t$, with $f$ being a higher frequency component value than $f+1$. $Mvar$ provides an insight on how smooth the progression of the vocalization magnitude is along the frequency axis. This is the case as by performing the difference between the maximums of contiguous frequency component we might identify the beginning or end of a vocalization contour, as they would present substantially different maximum magnitude values. By averaging these differences, a vocalization with a *steady* pattern (top left Figure 4.11) would present a lower $Mvar$ score than one with a more *erratic* behaviour (top right Figure 4.11), which could be helpful to discriminate vocalizations of different species.

### 4.4.2 Contour analysis features

As the vocalizations of most dolphin species boast a wide frequency range, which overlap for many species [68] (Table 4.3 and Figure 4.11), in theory the sole utilization of the frequency analysis feature subset ($Fs_1$) may not be sufficient to properly distinguish vocalizations of distinct dolphin species. To overcome this limitation, we developed two additional features, the **contour analysis features**, which intent to express some of the higher-level details in the contour of the vocalization's spectral representation. The two features which make up this feature subset ($Fs_2$) are: ($a$) the average slope difference ($AvgSlopeDif$), portraying the signals frequency progression over time; and ($b$) the number of inflexion points ($InflexNum$) that occur in a given vocalization's frequency contour.

It must be mentioned that for obtaining this last feature subset, the time-frequency representation is computed only from the STFT as the MFCCs are incapable of maintaining the level of detail in the vocalization pattern that the STFT provides. This can be seen when comparing both spectral representations in Figure 4.10

#### 4.4.2.1 Vocalization contour detection

The first step on the computation of the contour analysis features, is to detect the vocal contours in the magnitude spectrograms. To achieve this several different operations were performed in succession which are shown in Figure 4.12.

This process starts by applying a peak tracking technique to the magnitude spectrogram based on the MQ modeling and PARSHL techniques [69–71]. This process starts by looking for the dominant intensity peaks in each time frame. Due to the characteristics of real-world signals, we cannot simply consider the local maxima in the frame, as too many maxima would be found. Thus, after finding the local maxima, these are filtered out by a peak prominence criterion that compares the height of the peak with the height

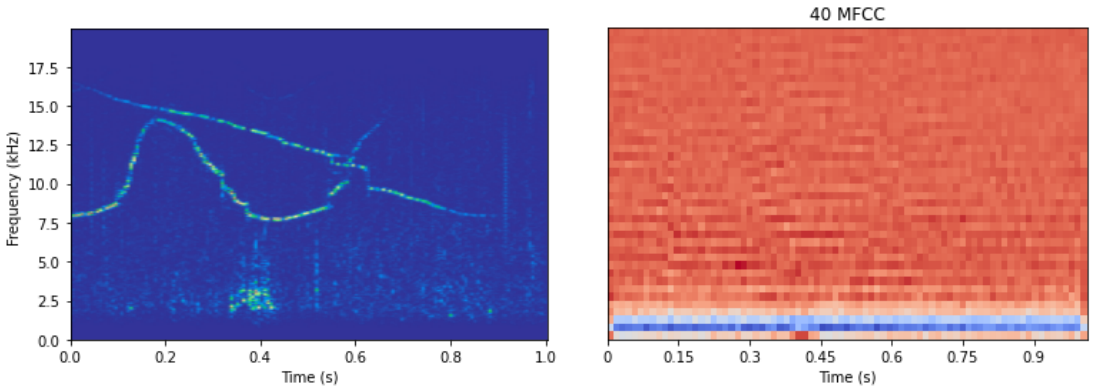



Figure 4.10: Comparison between a STFT time-frequency representation (magnitude spectrogram) and a MFCC time-frequency representation (40 MFCC). Both representations were obtained using frame and window lengths of 512.

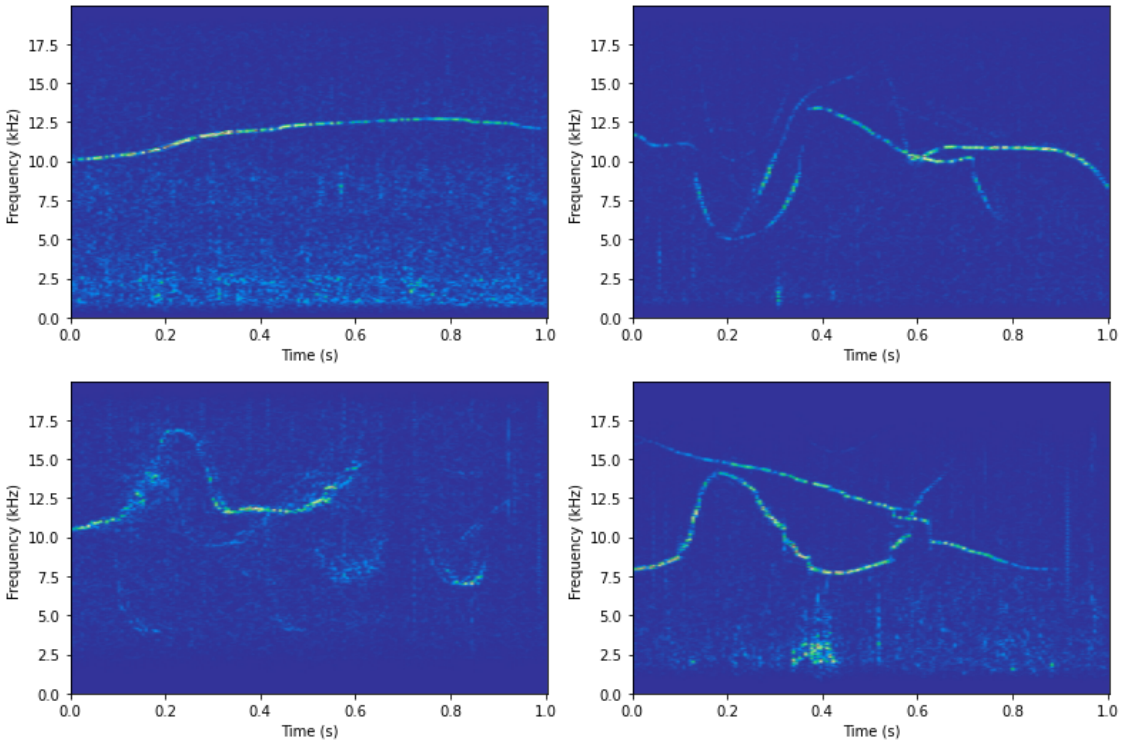



Figure 4.11: Spectrograms of four different dolphin species' vocalizations. From left to right, top to bottom: *delphinus delphis*, *tursiops truncatus*, *stenella frontalis*, *stenella coeruleoalba*.

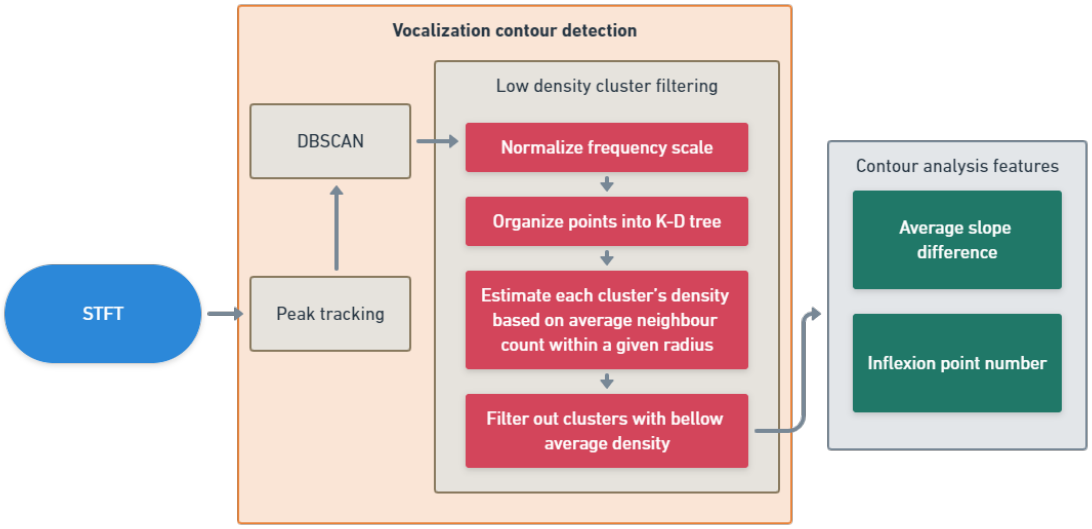



Figure 4.12: Vocalization contour detection pipeline.

of its immediate neighbors, which here are the 5 consecutive frequency bins around the peak (Figure 4.13).

The detection of these local magnitude peaks in successive time frames expresses

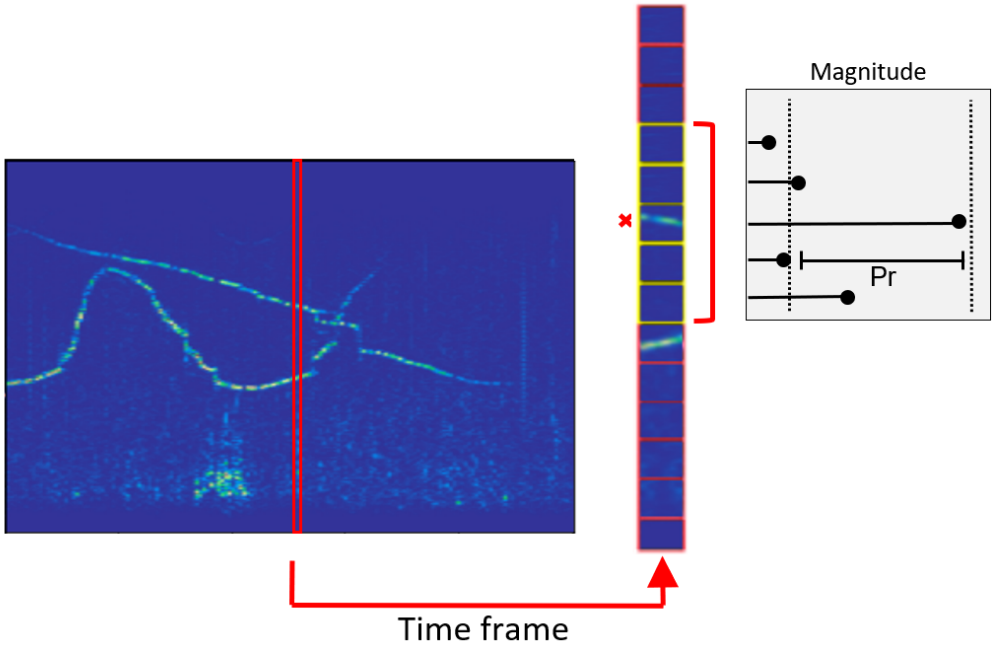



Figure 4.13: Detection of peak in a time frame with a minimum required prominence. The selected peak ($x$) will be valid if the prominence (Pr) to its lowest contour is at least equal to the 95th percentile of the magnitudes present in that time frame.

an unpolished vocalization pattern to which we then apply a clustering algorithm to remove outliers and to obtain clusters of different sections of the vocalization frequency contours. The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [72] (with *eps*=11 and *min_samples*=12) is used for this purpose. However, sometimes this approach may still generate some low density clusters which may correspond to background noise and thus, need to be filtered out (Figure 4.14). To do so, we must determine the density of each cluster and then filter out the ones which have a bellow average cluster density.

This can be achieved by organizing the cluster points into a K-D Tree (Figure 4.15), which allows for an efficient neighbour lookup within a given distance radius [73]. As we are dealing with distances, the coordinates of the points in the cluster must be normalized between 0 and 1 before building the K-D Tree, in order to have both the frequency and time coordinates in the same scale. In this work the used distance radius was 0.01 units in the normalized scale, which was determined by an empirical analysis during the development of this method. By having determined the close neighbours of every point in the cluster it is possible to then estimate the density of the cluster which is given by the average amount of neighbours a cluster has within the mentioned radius. The outcome of this process can be seen Figure 4.14.

These clusters will be the foundation for the estimation of the two contour analysis features.

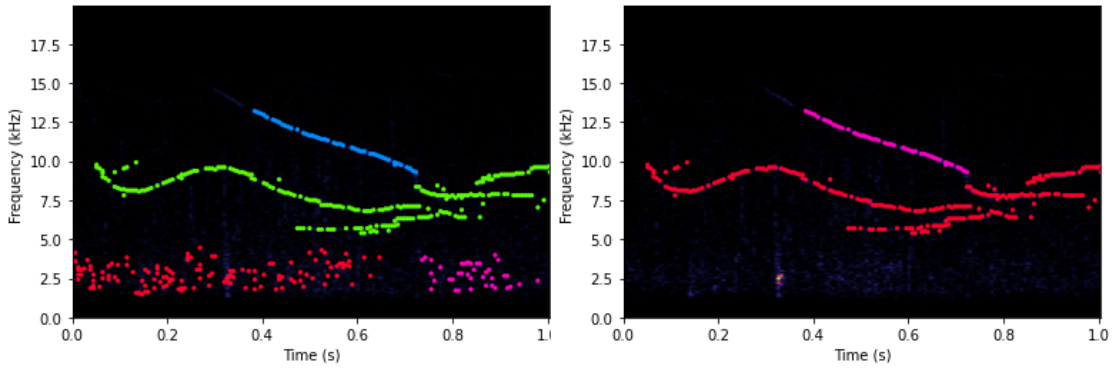



Figure 4.14: Frequency contour clusters of a vocalization of a striped dolphin (*stenella coeruleoalba*) after applying peak tracking and DBSCAN (left) and sucessive density based cluster filtering approach to remove low density clusters (right).

#### 4.4.2.2 Average slope diference

As both metrics in this feature subset rely on intermediate time instance calculations, the spectrogram containing the vocalization clusters is divided into $n$ time segments (which we set to 10 in our tests) and then the slopes of every cluster on each segment are computed.

For each cluster $c$ in time segment $[t_i, t_i + \Delta t^*]$, the first active point of the cluster ($P_{t_i}$) and the last active point ($P_{t_i+\Delta t^*}$) are used to calculate the time-frequency slope $S_{c,t_i}$ of cluster $c$ in that segment:

$$S_{c,t_i} = \frac{F_{P_{t_i+\Delta t^*}} - F_{P_{t_i}}}{\Delta t^*} \tag{4.5}$$

where $\Delta t^*$ is an approximate value of time interval $\Delta t = t_{i+1} - t_i$, since the first and last active points in the cluster may not coincide with those precise time frames. $F_{P_{t_i}}$ is the average frequency value of points within a smaller time window ($\pm\frac{(t_{i+1}-t_i)}{n}$) surrounding the closest point $P$ to time frame $t$, ($t_i$ or $t_{i+1}$). This approach is used as a way to more closely capture the real slope of the cluster, as the true frequency value of the closest point to time frame $t$ could be itself an outlier and misrepresent the true cluster's slope at that time. Figure 4.16 shows an example of how close the obtained cluster slopes overlap with the original vocalization contour.

With this, it is possible to estimate the average cluster slope difference for each cluster with the following expression:

$$ClustSlopeDif(c) = \frac{\sum_{i=1}^{\|T\|-1} S_{c,t_{i+1}} - S_{c,t_i}}{\|T\| - 1} \tag{4.6}$$

where $ClustSlopeDif(c)$ is given by the average of the difference between the slopes $S$ of adjacent time intervals $[t_i, t_{i+1}]$ in cluster $c$. As a cluster only accounts for a segment of

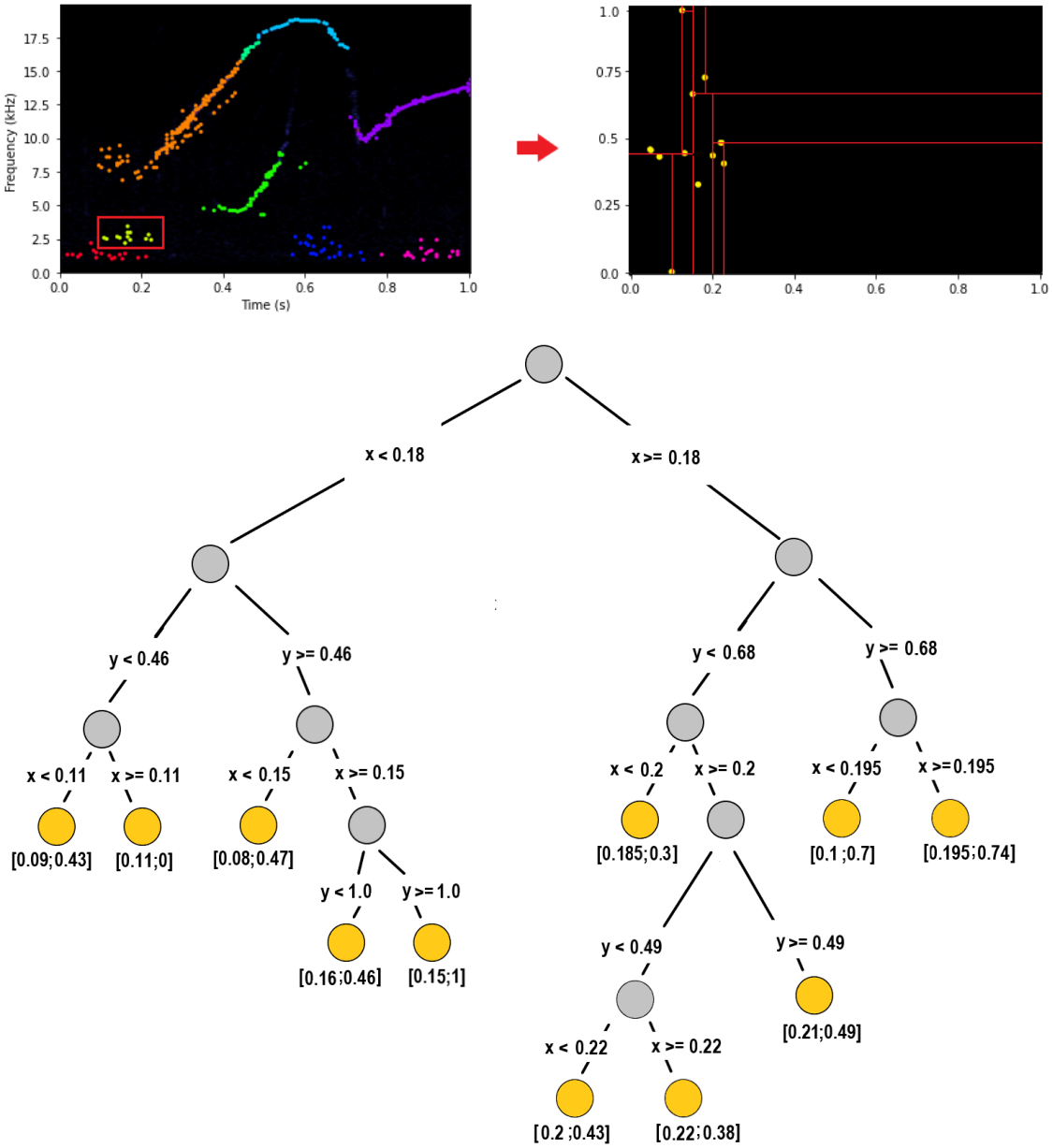



Figure 4.15: Resulting K-D Tree of a small cluster (highlighted in the top left Figure) obtained after applying the peak tracking and DBSCAN algorithms. The top right Figure represents the spatial fragmentation of the normalized spectrum from which the K-D tree was built (each leaf in the tree corresponds to a point on the plane, and the gray nodes correspond to the sections on the plane.).

a given vocalization, the final aimed feature is expressed by the average value for each cluster in the spectrum:

$$AvgSlopeDif = \frac{\sum_{c \in C} ClustSlopeDif(c)}{||C||} \tag{4.7}$$

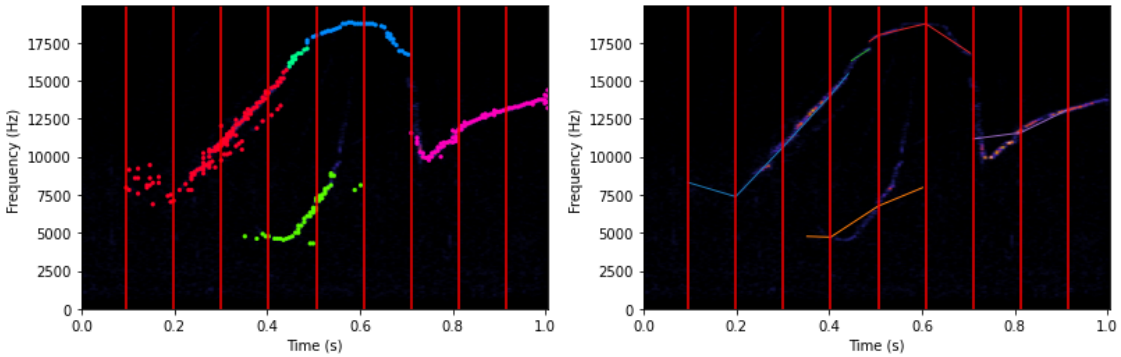



Figure 4.16: Comparison between the final vocalization clusters (left) with the obtained cluster slopes for each time step (right). It is possible to see that in spite not perfect, the resulting slopes make a good approximation of the original vocal contour.

This feature will represent the overall slope of the entire vocalization contour, providing an insight on the direction and how prevalent the dominant slope of the vocalization is. This also enables the identification of weather the vocalization is predominantly an upsweep or downsweep, something given by the sign of the $AvgSlopeDif$ value, whereas the absolute value describes the degree of this progression.

#### 4.4.2.3 Inflexion point number

The last feature in this subset takes into account the number of inflexion points, i.e. shifts between upsweeps and downsweeps, in a given vocalization. This feature might be useful to discriminate vocalizations which are more stable in their progression over time, from vocalizations which are more erratic and which are constantly shifting between upsweeps and downsweeps, which can be a discriminant factor to distinguish some species of dolphins. This feature is given by the following expression:

$$InflexNum = \sum_{c \in C} \sum_{i=1}^{\|T\|-1} 0^{(S_{c,t_i} \times S_{c,t_{i+1}} + |S_{c,t_i} \times S_{c,t_{i+1}}|)} \tag{4.8}$$

By taking advantage of the property that expresses that $0^0 = 1$ and $0^n = 0 \ \forall n \in \mathrm{R}_{>0}$, we multiply slopes $S_{c,t}$ in adjacent time intervals in such a way that if they carry opposite signs the power will be 0 and an inflexion in the pattern would be detected. This operation is applied to every cluster in a signal, resulting in the final number of inflexions in the vocalization.

### 4.4.3 Dimensionality reduction

Even though we presented five features across two distinct sets, one of the features, $MS_{fc}(f)$, can be reflected in as many *sub features* as the number of frequency components in the used spectrogram. Due to this, there is a need to perform some type of dimensionality reduction, as not only to get rid of redundant features (which might translate into

better classification results), but also to improve the execution time of our classification models.

In spite the fact that ICA is not traditionally used with the intent of dimensionality reduction, its capability of obtaining independent components which represent hidden aspects in the data might be useful as a way to improve the discriminative nature of our obtained features. Due to the well established nature of PCA as a dimentionality reduction technique, it was also chosen as an option to be tested. Finally a third alternative method which combined the use of PCA before applying ICA was also taken into consideration [39].

For all these alternatives the extraction of 8 components was tested and its results can be seen in Section 5.2. In order to perform both the PCA and FastICA algorithms the decomposition module of the sklearn phyton library was used.

## 4.5 Classification

After we preprocess our data, having it characterized by features, we can proceed to the learning and classification stage presented in Figure 1.2. The suggested methodology can be seen with greater detail in Figure 4.17.

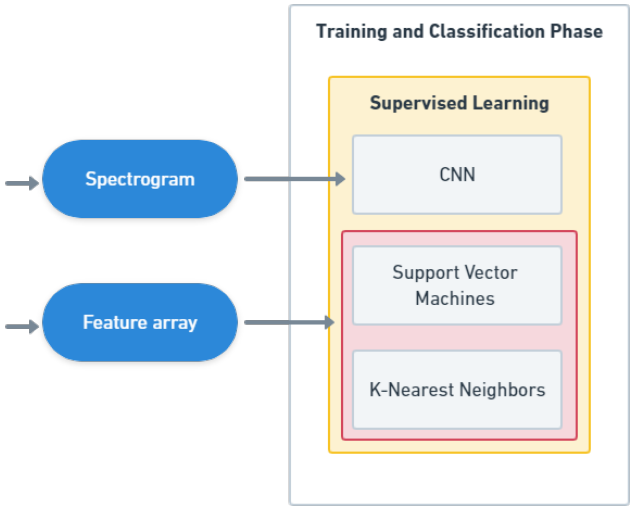



Figure 4.17: Training and Classification stage.

In this stage, we experiment with several different classification models, in a supervised learning fashion. We tested the performance of models such as SVM, K-NN and CNN's. However in this last approach only raw spectrograms were used as input to the CNN model and not the features in our feature set.

### 4.5.1 CNN

For the purpose of this work, the tests using CNN's only receive as input magnitude spectrograms instead of the obtained feature set. This happens as we intend to use the obtained CNN results as a solid comparison term towards the results obtained with the extracted features on the SVM and K-NN classifiers. This decision is based on how prevalent the usage of CNN's has become to solve bioacoustic classification tasks, in part due to its capability to extract usefull features from raw spectrograms while achieving satisfactory results (Section 3.1).

Three different models of CNN's were tested over 50 epochs, while testing two different batch sizes (64 and 100), using a learning rate of $0,0001$, a decay of $0,001$ and a frame and window size of 512. The tested models consist of two custom made models (CM1 and CM2) and the InceptionV3 model [74]. Both developed custom models are mainly composed by the same layer schematics only differing on the size, shape of the convolution kernel and amount of filters on those layers. While CM1 uses even shaped kernels in the convolution layers ([7x7] with stride 4 and [5x5] with stride 2), CM2 uses rectangular kernels which are taller than wider ([10x2] with stride 2 and [20x4] with stride 4). This decision means that while performing the convolutions on the CM2, the kernels will only overlap vertically (i.e. over the frequency axis) as the stride is the same as the width of the kernel. This means that the resulting feature map from the convolution will tend to represent features which account for thinner and more vertical vocalization contours (e.g rapid. shifts in frequency like narrow band high frequency clicks). It must be mentioned that all the used models use same padding during the convolutions, ReLU activation functions in between layers (to solve the vanishing gradient problem [19]) and have as the last layer, a dense layer of size 4 with a softmax activation in order to classify the dolphin classes with probabilistic values between 0 and 1. The schematics of both CM1 and CM2 can be seen in Figures 4.18 and 4.19 respectively.

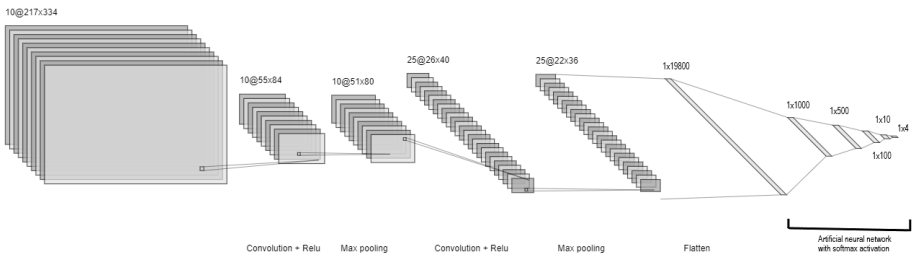



Figure 4.18: Architectural diagram of the CNN costum model 1 (CM1).

### 4.5.2 Training stage

The dataset was split into three distinct sets for the purposed of training (70%), validation (20%) and testing (10%). In order to obtain optimal results with both the SVM and

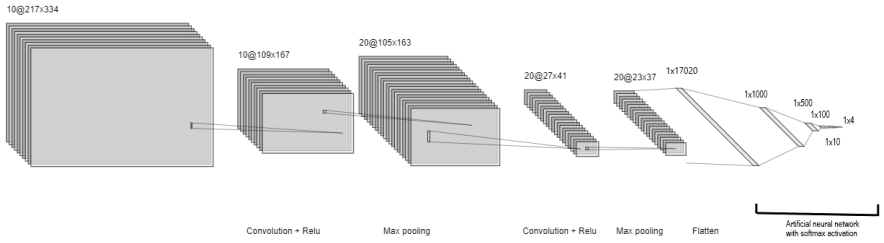



Figure 4.19: Architectural diagram of the CNN costum model 2 (CM2).

KNN classifiers we first need to estimate their optimal hyper-parameters. In the case of the SVM classifier (with RBF kernel) we have two hyper-parameters to estimate, *gamma* and $C$, while for KNN we have one, the number of neighbours, $K$. To optimize these parameters we used a stratified 5 fold cross validation approach, with a grid-search over predefined interval values of each parameter (Table 4.5).

Table 4.5: All of the tested values for each of the classifiers hyperparameters.

| | | **Hyperparameter values tested** |
|---|---|---|
| **SVM** | $C$: | {0.1, 1, 10, 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000} |
| | *Gamma*: | {200, 150, 100, 50, 10, 1, 0.1, 0.01, 0.001, 0.0001} |
| **K-NN** | $K$: | [3, 25] |

This optimization process will go through all combination of hyperparameter values (in our worst case 130 combinations with SVM) and note their training and test performance using stratified 5 fold-cross validation (Figure 4.20). The performance of a given combination of hyperparameters values corresponds to the averaged training and validation accuracy over the 5 folds. The combination with the best performance is then selected to perform the final training of the model.

Following the estimation of the optimal hyper-parameters, the training and validation sets are used to perform a final training of the models. When this training has concluded, the test set is used to generate each models predictions which are then represented by a confusion matrix. From this matrix it is possible to estimate both the final model and individual species accuracies. However, as we run each test 10 times, the results presented in Section 5, were obtained by the cumulative sum of every run's confusion matrix.

Regarding the CNN's training, it was done locally (using the system described in Section 4.5.2.1) and repeated 3 times for each CNN model. The results obtained only account for the entire model accuracy and not individual specie's accuracies.

#### 4.5.2.1 Experimental environment

The experimental environment consisted of a system running windows 10 version 21H1, with a Ryzen 5 3600 6-core CPU @ 3.6GHz, 16 GB of RAM and an Nvidia RTX 3060 graphics card (3584 Cuda cores, 12GB of memory and 112 tensor cores). The Phyton

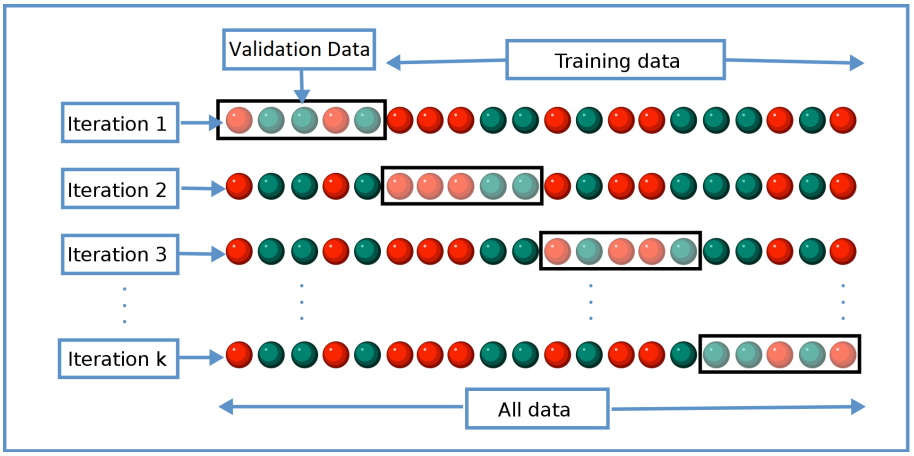



Figure 4.20: Diagram exemplifying $K$ fold cross validation, with the $k$ being the number of sets the data will be segmented in. By using stratified $k$ fold we also ensure that each fold in the data has the same proportion of observations for each class [75].

version used for development of this work was verion 3.8.11. For both the training and implementation of the CNN models, Tensorflow-gpu version 2.5.0, and keras version 2.4.3 were used. Other Phyton modules also used in the implementation of this work are: Librosa (to extract the MFCC), Pandas (for data management) and sklean (to use K-NN, SVM, DBSCAN and grid search with cross validation).

# 5

# Results and Discussion

In this section we present all the obtained results from the tests performed during the course of the realization of this dissertation. These contain the tests carried out to test the best way to calculate the frequency component magnitude sum feature (Section 5.1), the tests needed to determine the best dimensionality reduction approach (Section 5.2), the CNN model classification tests (Section 5.3) and the overall SVM and K-NN model tests over several different combinations of features in the feature set and also different time-frequency representation methods (Section 5.4). As stated in Section 4.5.2 each of the performed tests was run 10 times for each of its tested parameter combinations, with the presented results accounting for the averaged prediction accuracy from those runs. The obtained model accuracy is represented in each table by the $M_{acc}$ column.

## 5.1 Vocalization peak contours $MS_{fc}$ features test

The following test consisted in discovering the best approach to calculate the frequency component magnitude sum features $MS_{fc}$ . The first approach (*Peaks*) consisted in only considering for the feature the frequency components where vocalization contours were detected (by using a peak tracking technique). The second approach ($AllFreq_{comp}$) consisted in using all of the frequency components available in the selected time-frequency representation. Each of these approaches was tested by using them to derivate the $MS_{fc}$ features, obtained from a STFT time-frequency representation. Then, by applying ICA as a dimensionality reduction method over the $MS_{fc}$ features and the remaining features in the entire feature set, 8 components were extracted and used to train both a K-NN and SVM model. The obtained model accuracies can be seen in Table 5.1.

By analysing both subtables it can be seen that using all of the available frequency components in the time-frequency representation provides better model accuracy results. This is clear as in both the K-NN and SVM tests, this approach ($AllFreq_{comp}$) outperforms the *Peaks* on all of the individual species accuracies and all of the model accuracies (84.81% vs 40.96% in the K-NN tests and 85.77% vs 45.85 in the SVM ones). For this reason in the following tests the *Peaks* approach was not considered to derive the $MS_{fc}$ features.

Table 5.1: Comparison between using all the frequency components and using only the ones which match vocalization contours (peak extraction) to derive the frequency component magnitude sum feature (window len=512, frame size=512; 8 components (ICA) extracted from the both feature subsets ($Fs_1$ and $Fs_2$)).

| | KNN | | | | |
|---|---|---|---|---|---|
| | Common dolphin (%) | Bottlenose dolphin (%) | Spotted dolphin (%) | Striped dolphin (%) | $M_{acc}$ (%) |
| *Peaks* | 53.08 | 36.15 | 27.62 | 46.92 | **40.96** |
| *All* $Freq_{Comp}$ | 82.31 | 81.54 | 92.31 | 83.08 | **84.81** |

| | SVM | | | | |
|---|---|---|---|---|---|
| | Common dolphin (%) | Bottlenose dolphin (%) | Spotted dolphin (%) | Striped dolphin (%) | $M_{acc}$ (%) |
| *Peaks* | 49.23 | 30.00 | 57.25 | 46.92 | **45.85** |
| *All* $Freq_{Comp}$ | 80.77 | 88.46 | 90.00 | 83.85 | **85.77** |

## 5.2 Dimensionality reduction tests

The dimensionality reduction tests consisted in testing which of the three considered dimensionality reduction methods (PCA,ICA, PCA+ICA) provided the best model accuracy in an even test setting. This was achieved by testing each of these methods over both feature subsets ($Fs_1$ and $Fs_2$), obtained from both STFT and MFCC time-frequency representations. Then the reduced features (8 extracted components) were used to train both a K-NN and SVM model. The obtained model accuracies can be seen in Tables 5.2 and 5.3.

Table 5.2: Dimensionality reduction test results for KNN. The three different approaches were tested on both time frequency representations (window len=512, frame size=512; 20 MFCCs) while extracting 8 components from both feature subsets ($Fs_1$ and $Fs_2$).

| | | KNN | | | | |
|---|---|---|---|---|---|---|
| | | Common dolphin (%) | Bottlenose dolphin (%) | Spotted dolphin (%) | Striped dolphin (%) | $M_{acc}$ (%) |
| PCA | *STFT* | 51.54 | 28.46 | 23.33 | 36.92 | **35.06** |
| | *MFCC* | 36.15 | 33.85 | 33.33 | 40.00 | **35.83** |
| ICA | *STFT* | 83.85 | 83.08 | 94.62 | 83.85 | **86.35** |
| | *MFCC* | 74.62 | 78.46 | 90.77 | 90.00 | **83.46** |
| PCA+ICA | *STFT* | 82.31 | 81.54 | 92.31 | 83.08 | **84.81** |
| | *MFCC* | 61.54 | 74.62 | 96.15 | 88.46 | **80.19** |

By analysing the obtained results, it is possible to see that, in this problem context the usage of PCA alone as a dimensionality reduction method leads to poor performing models with accuracies ranging from as low as 34.23% (SVM-20 MFCC) to 49.04%(SVM-STFT). On the other hand, the usage of ICA and PCA+ICA performed similarly. While the usage of ICA achieved better results in the K-NN tests, with a model accuracy of 86.35% (K-NN-STFT) against 84.81% with PCA+ICA (K-NN-STFT), PCA+ICA achieved better results in the SVM tests, with a model accuracy of 89.04% (SVM-MFCC) against 85.77% (SVM-MFCC and SVM-STFT). In spite of their similar performances, ICA obtained a better average accuracy around all tests in both classification models (85.34% against

Table 5.3: Dimensionality reduction test results for SVM. The three different approaches were tested on both time frequency representations (window len=512, frame size=512; 20 MFCCs) while extracting 8 components from the entire feature set ($Fs_1$ and $Fs_2$).

| | | SVM | | | | |
|---|---|---|---|---|---|---|
| | | Common dolphin (%) | Bottlenose dolphin (%) | Spotted dolphin (%) | Striped dolphin (%) | $M_{acc}$ (%) |
| PCA | *STFT* | 51.54 | 45.38 | 30.77 | 68.46 | **49.04** |
| | *MFCC* | 29.23 | 29.23 | 26.92 | 51.54 | **34.23** |
| ICA | *STFT* | 80.77 | 88.46 | 90.00 | 83.85 | **85.77** |
| | *MFCC* | 72.31 | 90.77 | 86.92 | 93.08 | **85.77** |
| PCA+ICA | *STFT* | 78.46 | 83.85 | 90.00 | 80.77 | **83.27** |
| | *MFCC* | 83.85 | 87.69 | 90.00 | 94.62 | **89.04** |

84.33% from PCA+ICA). For this better all around performance accuracy, ICA was used as the dimensionality reduction method in the overall tests in Section 5.4.

## 5.3 CNN results

The CNN tests consisted in evaluating the performance of the 3 different CNN models mentioned in Section 4.5.1 (custom models $CM_1$ and $CM_2$ and the InceptionV3 model). Each model was trained over 50 epochs using two distinct batch sizes (64 and 100), a learning rate of 0,0001 and decay of 0,001. As input for the models, spectrograms obtained from a STFT time-frequency representation (window and frame size of 512) were used. The obtained model accuracy results can be seen in Table 5.4.

Table 5.4: Model accuracy of each CNN model architecture using spectrograms obtained using a window lenght and frame size of 512.

| Model | Batch size | $M_{acc}$ (%) |
|---|---|---|
| $CM_1$ | 64 | 81.68 |
| $CM_1$ | 100 | 82.78 |
| $CM_2$ | 64 | 85.89 |
| $CM_2$ | 100 | 75.46 |
| InceptionV3 | 64 | 80.03 |
| InceptionV3 | 100 | 73.26 |

By observing the table it is possible to see that the obtained model accuracies are comprehended between 73.26% (InceptionV3 model using a batch size of 100) and 85.89% ($CM_2$ using a batch size of 64). From the three models $CM_1$ provided the most consistent results among the two batch size tests, achieving 81.68% with a batch size of 64 and 82.78% with a batch size of 100.

## 5.4 Overall performance results

This section highlights the overall performance tests of the developed features in this dissertation, when derived from different time-frequency representations and used with

any of two classification models SVM or K-NN. These tests were designed in order to evaluate the impact that the parameterization of the feature extraction stage has on the final model accuracy and also on the individual species accuracies. This was achieved by testing for each classifier:

- five different time-frequency representation variations (STFT and MFCC with four different amounts of coefficients extracted);
- tree distinct STFT parameter combinations (window and frame size) mentioned in Section 4.3;
- seven different combinations of features within the created feature set (Section 4.4).

All of the obtained features were then reduced using ICA as a dimensionality reduction method before being used in the classifiers. The obtained results span six Tables (Tables 5.5, 5.6, 5.7, 5.8, 5.9 and 5.10), three per classifier (each corresponding to a given STFT parameter combination).

In order to more easily visualize the obtained results, three additional visual representations of the obtained data were created. The first can be seen in Figure 5.1 and presents a box plot of the several model accuracies, obtained using different feature combinations, while using a given classifier with a specific time-frequency representation. The second representation in Figure 5.2 shows a bar chart highlighting the averaged model accuracy of every feature combination tested with a given STFT parameters and classifier, allowing to more easily visualize the impact that the introduction of a given feature has on the performance of the model. Finally, the third visualization in Figure 5.3 which is also a bar chart, highlights the average accuracy obtained from all of the tests which used the same STFT parameters combinations, allowing to detect which one contributed to better all around results.

A superficial analysis of the results obtained by both classifiers suggest that the SVM model slightly outperforms the K-NN model, achieving a maximum model accuracy of 96.15% (Table 5.8, 120 MFCC, using as features $MS_{fc}$+$Mvar$) against 95.58% achieved with the same test configuration with K-NN (Table 5.6).

By performing an overview on the impact which a given time-frequency representation has on a classifier performance, it is possible to observe that in both classifiers the features derived from MFCC tended to perform better than the ones obtained from the STFT. This can be observed in both the K-NN and SVM tests, however it is more evident when using SVM (Tables 5.8,5.9,5.10). In these tests, almost every feature combination test which uses a STFT time-frequency representation is outperformed by all the remaining MFCC representations with the same frame and window size. The only time this does not happen can be seen in Table 5.10, where a STFT representation with a frame size of 1024 and a window length of 512 outperforms 20 MFCC coefficients when using a feature combination of $MS_{fc}$+$Mvar$ (89.23% vs 87.12%). By observing the visualization

Table 5.5: Accuracy results for K-NN when using a frame size of 512 and window length of 512. With $Fs_1$:{$MS_{fc}$;$VarCoef$;$Mvar$} and $Fs_2$:{$AvgSlopeDif$;$InflexNum$}

| | | KNN (frame size=512, window len=512) | | | | |
|---|---|---|---|---|---|---|
| | | Common dolphin (%) | Bottlenose dolphin (%) | Spotted dolphin (%) | Striped dolphin (%) | $M_{acc}$ (%) |
| $STFT$ | $MS_{fc}$ | 87.02 | 94.62 | 96.92 | 84.62 | **90.79** |
| | $MS_{fc}$+$VarCoef$ | 86.15 | 86.92 | 95.38 | 83.08 | **87.88** |
| | $MS_{fc}$+$Mvar$ | 84.62 | 83.08 | 95.38 | 88.46 | **87.88** |
| | $Fs_1$ | 81.54 | 88.46 | 94.62 | 87.69 | **88.08** |
| | $Fs_1$+$AvgSlopeDif$ | 79.23 | 83.08 | 93.08 | 80.00 | **83.85** |
| | $Fs_1$+$InflexNum$ | 82.31 | 88.46 | 98.46 | 86.15 | **88.85** |
| | $Fs_1$+$Fs_2$ | 86.15 | 80.00 | 90.00 | 84.62 | **85.19** |
| $MFCC_{20}$ | $MS_{fc}$ | 82.31 | 91.54 | 99.23 | 90.00 | **90.77** |
| | $MS_{fc}$+$VarCoef$ | 83.08 | 90.00 | 98.46 | 85.38 | **89.23** |
| | $MS_{fc}$+$Mvar$ | 81.54 | 94.62 | 98.46 | 90.00 | **91.15** |
| | $Fs_1$ | 85.38 | 88.46 | 96.15 | 90.00 | **90.00** |
| | $Fs_1$+$AvgSlopeDif$ | 70.77 | 80.00 | 96.92 | 93.08 | **85.19** |
| | $Fs_1$+$InflexNum$ | 79.23 | 87.69 | 96.15 | 93.08 | **89.04** |
| | $Fs_1$+$Fs_2$ | 71.54 | 83.08 | 93.85 | 89.23 | **84.42** |
| $MFCC_{40}$ | $MS_{fc}$ | 85.38 | 91.54 | 97.69 | 96.92 | **92.88** |
| | $MS_{fc}$+$VarCoef$ | 84.62 | 89.23 | 98.46 | 91.54 | **90.96** |
| | $MS_{fc}$+$Mvar$ | 86.15 | 90.77 | 96.92 | 93.08 | **91.73** |
| | $Fs_1$ | 91.54 | 91.54 | 97.69 | 90.77 | **92.88** |
| | $Fs_1$+$AvgSlopeDif$ | 78.46 | 86.15 | 92.31 | 93.08 | **87.50** |
| | $Fs_1$+$InflexNum$ | 83.08 | 90.00 | 95.38 | 96.92 | **91.35** |
| | $Fs_1$+$Fs_2$ | 86.15 | 81.54 | 82.31 | 93.85 | **85.96** |
| $MFCC_{60}$ | $MS_{fc}$ | 91.54 | 94.62 | 96.15 | 94.62 | **94.23** |
| | $MS_{fc}$+$VarCoef$ | 91.54 | 96.92 | 96.15 | 96.15 | **95.19** |
| | $MS_{fc}$+$Mvar$ | 84.62 | 93.85 | 97.69 | 96.15 | **93.08** |
| | $Fs_1$ | 85.38 | 92.31 | 98.46 | 96.15 | **93.08** |
| | $Fs_1$+$AvgSlopeDif$ | 80.77 | 84.62 | 83.85 | 95.38 | **86.15** |
| | $Fs_1$+$InflexNum$ | 81.54 | 88.46 | 92.31 | 94.62 | **89.23** |
| | $Fs_1$+$Fs_2$ | 76.15 | 83.08 | 82.31 | 95.38 | **84.23** |
| $MFCC_{120}$ | $MS_{fc}$ | 86.15 | 92.31 | 96.92 | 99.23 | **93.65** |
| | $MS_{fc}$+$VarCoef$ | 83.85 | 92.31 | 99.23 | 93.85 | **92.31** |
| | $MS_{fc}$+$Mvar$ | 81.54 | 93.85 | 98.46 | 94.62 | **92.12** |
| | $Fs_1$ | 84.62 | 93.08 | 99.23 | 96.15 | **93.27** |
| | $Fs_1$+$AvgSlopeDif$ | 76.15 | 93.85 | 90.00 | 100.00 | **90.00** |
| | $Fs_1$+$InflexNum$ | 75.38 | 86.92 | 93.08 | 98.46 | **88.46** |
| | $Fs_1$+$Fs_2$ | 73.85 | 80.77 | 85.38 | 90.77 | **82.69** |

in Figure 5.1, this comparative lack of performance of STFT time-representations against MFCC representations is again corroborated by comparing the placement of the resulting box plots of each test. Due to the poor performance of the STFT representations, seen by its lower median test values (represented by the solid horizontal gray line within the box plot), they will tend to be placed bellow any other boxplot within the same test parameters (same classifier, frame and window sizes). This visualization also highlights an upwards trend in model accuracy as we increase the number of coefficients in the MFCC representations (shown by the positioning of the box plot top whiskers and growing median accuracy of the tests). However, this trend in some instances reverts itself past the 40 coefficients mark, like it was the case with the K-NN tests using a frame size of 1024 and a window length of 512.

The impact that each feature combination has on the general model accuracy can be assessed by observing the results on the ables and Figure 5.2. These highlight a slight decline in overall accuracy when taking more features into account, especially

Table 5.6: Accuracy results for K-NN when using a frame size of 512 and window length of 256. With $Fs_1$:{$MS_{fc}$;$VarCoef$;$Mvar$} and $Fs_2$:{$AvgSlopeDif$;$InflexNum$}

| | | KNN (frame size=512, window len=256) | | | | |
|---|---|---|---|---|---|---|
| | | Common dolphin (%) | Bottlenose dolphin (%) | Spotted dolphin (%) | Striped dolphin (%) | $M_{acc}$ (%) |
| $STFT$ | $MS_{fc}$ | 80.00 | 86.15 | 98.46 | 82.31 | **86.73** |
| | $MS_{fc}$+$VarCoef$ | 72.31 | 88.46 | 93.85 | 78.46 | **83.27** |
| | $MS_{fc}$+$Mvar$ | 77.69 | 87.69 | 93.08 | 80.77 | **84.81** |
| | $Fs_1$ | 74.62 | 90.77 | 95.38 | 81.54 | **85.58** |
| | $Fs_1$+$AvgSlopeDif$ | 76.15 | 86.92 | 93.08 | 77.69 | **83.46** |
| | $Fs_1$+$InflexNum$ | 76.92 | 83.08 | 93.08 | 84.62 | **84.42** |
| | $Fs_1$+$Fs_2$ | 70.77 | 86.15 | 93.85 | 90.77 | **85.38** |
| $MFCC_{20}$ | $MS_{fc}$ | 85.38 | 90.77 | 100.00 | 93.08 | **92.31** |
| | $MS_{fc}$+$VarCoef$ | 83.08 | 89.23 | 95.38 | 91.54 | **89.81** |
| | $MS_{fc}$+$Mvar$ | 81.54 | 90.00 | 98.46 | 90.00 | **90.00** |
| | $Fs_1$ | 82.31 | 92.31 | 100.00 | 93.08 | **91.92** |
| | $Fs_1$+$AvgSlopeDif$ | 74.62 | 83.08 | 93.08 | 97.69 | **87.12** |
| | $Fs_1$+$InflexNum$ | 66.92 | 93.08 | 96.92 | 92.31 | **87.31** |
| | $Fs_1$+$Fs_2$ | 73.85 | 87.69 | 92.31 | 94.62 | **87.12** |
| $MFCC_{40}$ | $MS_{fc}$ | 83.08 | 93.85 | 97.69 | 93.08 | **91.92** |
| | $MS_{fc}$+$VarCoef$ | 91.54 | 90.00 | 94.62 | 95.38 | **92.88** |
| | $MS_{fc}$+$Mvar$ | 83.85 | 90.77 | 99.23 | 93.08 | **91.73** |
| | $Fs_1$ | 83.08 | 89.31 | 96.15 | 94.62 | **90.79** |
| | $Fs_1$+$AvgSlopeDif$ | 78.46 | 86.15 | 91.54 | 93.08 | **87.31** |
| | $Fs_1$+$InflexNum$ | 79.23 | 93.18 | 93.85 | 96.92 | **90.80** |
| | $Fs_1$+$Fs_2$ | 65.38 | 90.77 | 85.38 | 96.15 | **84.42** |
| $MFCC_{60}$ | $MS_{fc}$ | 83.08 | 93.85 | 94.62 | 96.15 | **91.92** |
| | $MS_{fc}$+$VarCoef$ | 89.23 | 90.00 | 98.46 | 92.31 | **92.50** |
| | $MS_{fc}$+$Mvar$ | 90.77 | 89.23 | 97.69 | 95.38 | **93.27** |
| | $Fs_1$ | 81.54 | 90.00 | 96.92 | 94.62 | **90.77** |
| | $Fs_1$+$AvgSlopeDif$ | 83.08 | 87.69 | 90.77 | 93.85 | **88.85** |
| | $Fs_1$+$InflexNum$ | 73.08 | 90.00 | 90.00 | 96.92 | **87.50** |
| | $Fs_1$+$Fs_2$ | 74.62 | 84.62 | 74.62 | 97.69 | **82.88** |
| $MFCC_{120}$ | $MS_{fc}$ | 86.92 | 89.23 | 99.23 | 96.92 | **93.08** |
| | $MS_{fc}$+$VarCoef$ | 86.15 | 93.85 | 97.69 | 96.15 | **93.46** |
| | $MS_{fc}$+$Mvar$ | 93.08 | 93.08 | 98.46 | 97.69 | **95.58** |
| | $Fs_1$ | 83.85 | 94.62 | 97.69 | 96.92 | **93.27** |
| | $Fs_1$+$AvgSlopeDif$ | 78.46 | 86.15 | 93.08 | 98.46 | **89.04** |
| | $Fs_1$+$InflexNum$ | 77.69 | 83.08 | 92.31 | 96.15 | **87.31** |
| | $Fs_1$+$Fs_2$ | 68.46 | 82.31 | 77.69 | 97.69 | **81.54** |

when the introduced features are one of the two frequency contour analysis features belonging to $Fs_2$ (Section 4.4.2). This is clear as we observe a sharp decline in the averaged model accuracy when either the $AvgSlopeDif$ or $InflexNum$ are introduced (apart from testing $Fs_1$+$InflexNum$ with a frame and window length of 512 on a SVM, where a small performance uplift occurs). In spite of this, by observing the individual species accuracies on the result tables we can still make a case for these features, as their introduction has been proven helpful for some species in some scenarios. An example of this is the usage of the $InflexNum$ feature to boost the individual accuracy of the bottlenose dolphin in the SVM tests in Table 5.8 (with a frame and window lenght of 512), where for several different time-frequency representations it achieved a maximum individual accuracy when comparing with other feature combinations (93.08% when using STFT; 94.62% with 60 MFCC and 96.92% with 120 MFCC). Another species which seemed to benefit from either the $AvgSlopeDif$ or the $InflexNum$ features on some instances was the stripped dolphin, which managed to achieve maximum accuracies with any of those features in

Table 5.7: Accuracy results for K-NN when using a frame size of 1024 and window length of 512. With $Fs_1$:{$MS_{fc}$;$VarCoef$;$Mvar$} and $Fs_2$:{$AvgSlopeDif$;$InflexNum$}

| | | KNN (frame size=1024, window len=512) | | | | |
|---|---|---|---|---|---|---|
| | | Common dolphin (%) | Bottlenose dolphin (%) | Spotted dolphin (%) | Striped dolphin (%) | $M_{acc}$ (%) |
| $STFT$ | $MS_{fc}$ | 85.38 | 87.69 | 96.92 | 88.46 | **89.62** |
| | $MS_{fc}$+$VarCoef$ | 83.08 | 90.00 | 96.92 | 88.46 | **89.62** |
| | $MS_{fc}$+$Mvar$ | 88.46 | 83.85 | 94.62 | 88.46 | **88.85** |
| | $Fs_1$ | 82.31 | 91.54 | 96.92 | 85.38 | **89.04** |
| | $Fs_1$+$AvgSlopeDif$ | 79.23 | 80.77 | 95.38 | 80.00 | **83.85** |
| | $Fs_1$+$InflexNum$ | 86.92 | 88.46 | 97.69 | 83.08 | **89.04** |
| | $Fs_1$+$Fs_2$ | 80.77 | 88.46 | 94.62 | 81.54 | **86.35** |
| $MFCC_{20}$ | $MS_{fc}$ | 80.00 | 90.77 | 96.92 | 89.23 | **89.23** |
| | $MS_{fc}$+$VarCoef$ | 87.69 | 85.38 | 98.46 | 91.54 | **90.77** |
| | $MS_{fc}$+$Mvar$ | 76.92 | 83.85 | 99.23 | 91.54 | **87.88** |
| | $Fs_1$ | 83.85 | 88.46 | 97.69 | 89.23 | **89.81** |
| | $Fs_1$+$AvgSlopeDif$ | 76.15 | 83.08 | 93.85 | 96.15 | **87.31** |
| | $Fs_1$+$InflexNum$ | 74.62 | 84.62 | 97.69 | 96.92 | **88.46** |
| | $Fs_1$+$Fs_2$ | 72.31 | 77.69 | 92.31 | 96.15 | **84.62** |
| $MFCC_{40}$ | $MS_{fc}$ | 91.54 | 91.54 | 97.69 | 96.15 | **94.23** |
| | $MS_{fc}$+$VarCoef$ | 86.92 | 87.69 | 99.23 | 95.38 | **92.31** |
| | $MS_{fc}$+$Mvar$ | 87.69 | 90.77 | 100.00 | 93.85 | **93.08** |
| | $Fs_1$ | 89.23 | 90.00 | 96.92 | 96.15 | **93.08** |
| | $Fs_1$+$AvgSlopeDif$ | 79.23 | 81.54 | 92.31 | 88.46 | **85.38** |
| | $Fs_1$+$InflexNum$ | 82.31 | 86.92 | 98.46 | 92.31 | **90.00** |
| | $Fs_1$+$Fs_2$ | 79.23 | 83.85 | 90.77 | 92.31 | **86.54** |
| $MFCC_{60}$ | $MS_{fc}$ | 86.92 | 93.08 | 99.23 | 93.85 | **93.27** |
| | $MS_{fc}$+$VarCoef$ | 89.23 | 91.54 | 99.23 | 91.54 | **92.88** |
| | $MS_{fc}$+$Mvar$ | 84.62 | 92.31 | 96.92 | 93.85 | **91.92** |
| | $Fs_1$ | 83.08 | 94.62 | 95.38 | 90.77 | **90.96** |
| | $Fs_1$+$AvgSlopeDif$ | 81.54 | 86.92 | 93.08 | 95.35 | **89.22** |
| | $Fs_1$+$InflexNum$ | 70.77 | 94.62 | 96.92 | 94.62 | **89.23** |
| | $Fs_1$+$Fs_2$ | 72.31 | 83.85 | 89.23 | 95.38 | **85.19** |
| $MFCC_{120}$ | $MS_{fc}$ | 86.92 | 90.77 | 98.46 | 90.77 | **91.73** |
| | $MS_{fc}$+$VarCoef$ | 87.69 | 89.23 | 97.69 | 93.85 | **92.12** |
| | $MS_{fc}$+$Mvar$ | 85.38 | 93.08 | 97.69 | 95.38 | **92.88** |
| | $Fs_1$ | 86.15 | 86.15 | 96.92 | 92.31 | **90.38** |
| | $Fs_1$+$AvgSlopeDif$ | 86.15 | 90.77 | 95.38 | 90.77 | **90.77** |
| | $Fs_1$+$InflexNum$ | 77.69 | 84.62 | 95.38 | 94.62 | **88.08** |
| | $Fs_1$+$Fs_2$ | 70.77 | 83.85 | 86.15 | 95.38 | **84.04** |

several test parameterizations (Table 5.10: 94.62% when using $Fs_1$+$AvgSlopeDif$ on a SVM with 120 MFCC and a frame size of 1024 and window lenght of 512).

Finally, in order to evaluate how the selected STFT parameter combinations affected a given model's accuracy we can observe Figure 5.3. By analysing the results it is possible to see that the usage of a frame and window size of 512 yielded the better average results in both classifiers (K-NN: 89.522% and SVM: 90.801%), in spite of both of the maximum model accuracies obtained by both classification models were achieved by using a smaller window length of 256.

Table 5.8: Accuracy results for SVM when using a frame size of 512 and window length of 512. With $Fs_1$:{$MS_{fc}$;$VarCoef$;$Mvar$} and $Fs_2$:{$AvgSlopeDif$;$InflexNum$}

| | | **SVM** (frame size=512, window len=512) | | | | |
|---|---|---|---|---|---|---|
| | | Common dolphin (%) | Bottlenose dolphin (%) | Spotted dolphin (%) | Striped dolphin (%) | $M_{acc}$ (%) |
| $STFT$ | $MS_{fc}$ | 73.85 | 86.92 | 93.08 | 84.62 | **84.62** |
| | $MS_{fc}$+$VarCoef$ | 79.23 | 82.31 | 92.31 | 90.77 | **86.15** |
| | $MS_{fc}$+$Mvar$ | 83.85 | 85.38 | 93.85 | 88.46 | **87.88** |
| | $Fs_1$ | 83.08 | 90.77 | 96.92 | 83.08 | **88.46** |
| | $Fs_1$+$AvgSlopeDif$ | 73.08 | 80.00 | 92.31 | 90.77 | **84.04** |
| | $Fs_1$+$InflexNum$ | 80.77 | 93.08 | 95.38 | 81.54 | **87.69** |
| | $Fs_1$+$Fs_2$ | 81.54 | 79.23 | 96.15 | 84.62 | **85.38** |
| $MFCC_{20}$ | $MS_{fc}$ | 83.08 | 92.31 | 96.92 | 93.08 | **91.35** |
| | $MS_{fc}$+$VarCoef$ | 89.23 | 93.08 | 95.38 | 97.69 | **93.85** |
| | $MS_{fc}$+$Mvar$ | 85.38 | 90.77 | 93.08 | 94.62 | **90.96** |
| | $Fs_1$ | 85.38 | 91.54 | 93.08 | 94.62 | **91.15** |
| | $Fs_1$+$AvgSlopeDif$ | 84.62 | 90.77 | 86.92 | 93.08 | **88.85** |
| | $Fs_1$+$InflexNum$ | 85.38 | 86.92 | 96.92 | 96.15 | **91.35** |
| | $Fs_1$+$Fs_2$ | 80.77 | 88.46 | 96.15 | 89.23 | **88.65** |
| $MFCC_{40}$ | $MS_{fc}$ | 93.85 | 94.62 | 93.85 | 92.31 | **93.65** |
| | $MS_{fc}$+$VarCoef$ | 90.77 | 94.62 | 89.23 | 96.15 | **92.69** |
| | $MS_{fc}$+$Mvar$ | 94.62 | 94.62 | 95.38 | 96.92 | **95.38** |
| | $Fs_1$ | 86.92 | 97.69 | 93.08 | 95.38 | **93.27** |
| | $Fs_1$+$AvgSlopeDif$ | 85.38 | 90.00 | 93.08 | 93.85 | **90.58** |
| | $Fs_1$+$InflexNum$ | 88.46 | 94.62 | 95.38 | 96.92 | **93.85** |
| | $Fs_1$+$Fs_2$ | 86.92 | 86.92 | 89.23 | 93.85 | **89.23** |
| $MFCC_{60}$ | $MS_{fc}$ | 89.23 | 94.62 | 91.54 | 96.15 | **92.88** |
| | $MS_{fc}$+$VarCoef$ | 93.08 | 92.31 | 93.08 | 95.38 | **93.46** |
| | $MS_{fc}$+$Mvar$ | 90.00 | 89.23 | 90.70 | 93.08 | **90.75** |
| | $Fs_1$ | 85.38 | 92.31 | 95.38 | 95.38 | **92.12** |
| | $Fs_1$+$AvgSlopeDif$ | 86.15 | 91.54 | 88.46 | 90.00 | **89.04** |
| | $Fs_1$+$InflexNum$ | 91.54 | 94.62 | 90.77 | 93.08 | **92.50** |
| | $Fs_1$+$Fs_2$ | 81.54 | 89.23 | 86.15 | 88.46 | **86.35** |
| $MFCC_{120}$ | $MS_{fc}$ | 90.00 | 95.38 | 95.38 | 94.62 | **93.85** |
| | $MS_{fc}$+$VarCoef$ | 85.38 | 94.62 | 96.15 | 98.46 | **93.65** |
| | $MS_{fc}$+$Mvar$ | 90.00 | 93.85 | 97.69 | 95.38 | **94.23** |
| | $Fs_1$ | 83.85 | 94.62 | 100.00 | 95.38 | **93.46** |
| | $Fs_1$+$AvgSlopeDif$ | 93.85 | 90.77 | 94.62 | 93.08 | **93.08** |
| | $Fs_1$+$InflexNum$ | 87.69 | 96.92 | 93.85 | 96.90 | **93.84** |
| | $Fs_1$+$Fs_2$ | 84.62 | 87.69 | 92.31 | 94.62 | **89.81** |

Table 5.9: Accuracy results for SVM when using a frame size of 512 and window length of 256. With $Fs_1$:{$MS_{fc}$;$VarCoef$;$Mvar$} and $Fs_2$:{$AvgSlopeDif$;$InflexNum$}

| | | **SVM** (frame size=512, window len=256) | | | | |
|---|---|---|---|---|---|---|
| | | Common dolphin (%) | Bottlenose dolphin (%) | Spotted dolphin (%) | Striped dolphin (%) | $M_{acc}$ (%) |
| $STFT$ | $MS_{fc}$ | 85.38 | 83.08 | 93.85 | 86.92 | **87.31** |
| | $MS_{fc}$+$VarCoef$ | 82.31 | 91.54 | 94.62 | 83.85 | **88.08** |
| | $MS_{fc}$+$Mvar$ | 76.15 | 89.23 | 95.38 | 87.69 | **87.12** |
| | $Fs_1$ | 70.00 | 86.92 | 95.38 | 84.62 | **84.23** |
| | $Fs_1$+$AvgSlopeDif$ | 77.69 | 74.62 | 95.38 | 83.08 | **82.69** |
| | $Fs_1$+$InflexNum$ | 77.69 | 81.54 | 96.15 | 86.92 | **85.58** |
| | $Fs_1$+$Fs_2$ | 73.85 | 83.08 | 86.92 | 81.54 | **81.35** |
| $MFCC_{20}$ | $MS_{fc}$ | 80.77 | 92.31 | 93.85 | 94.62 | **90.38** |
| | $MS_{fc}$+$VarCoef$ | 87.69 | 91.54 | 96.15 | 92.31 | **91.92** |
| | $MS_{fc}$+$Mvar$ | 86.92 | 93.08 | 89.23 | 96.92 | **91.54** |
| | $Fs_1$ | 85.38 | 91.54 | 91.54 | 93.85 | **90.58** |
| | $Fs_1$+$AvgSlopeDif$ | 80.77 | 86.15 | 90.77 | 94.62 | **88.08** |
| | $Fs_1$+$InflexNum$ | 83.85 | 82.31 | 91.54 | 95.38 | **88.27** |
| | $Fs_1$+$Fs_2$ | 86.15 | 86.92 | 93.85 | 93.85 | **90.19** |
| $MFCC_{40}$ | $MS_{fc}$ | 90.77 | 93.85 | 92.31 | 96.92 | **93.46** |
| | $MS_{fc}$+$VarCoef$ | 90.77 | 90.77 | 91.54 | 96.92 | **92.50** |
| | $MS_{fc}$+$Mvar$ | 92.31 | 94.62 | 98.46 | 93.08 | **94.62** |
| | $Fs_1$ | 87.69 | 92.31 | 94.62 | 96.92 | **92.88** |
| | $Fs_1$+$AvgSlopeDif$ | 86.15 | 86.92 | 87.69 | 96.92 | **89.42** |
| | $Fs_1$+$InflexNum$ | 84.62 | 90.00 | 90.77 | 96.15 | **90.38** |
| | $Fs_1$+$Fs_2$ | 86.92 | 89.23 | 93.08 | 96.15 | **91.35** |
| $MFCC_{60}$ | $MS_{fc}$ | 93.85 | 95.38 | 96.92 | 92.31 | **94.62** |
| | $MS_{fc}$+$VarCoef$ | 90.00 | 93.85 | 96.92 | 96.15 | **94.23** |
| | $MS_{fc}$+$Mvar$ | 86.92 | 90.00 | 96.15 | 90.77 | **90.96** |
| | $Fs_1$ | 92.31 | 90.77 | 93.85 | 93.85 | **92.69** |
| | $Fs_1$+$AvgSlopeDif$ | 90.00 | 93.08 | 87.69 | 96.92 | **91.92** |
| | $Fs_1$+$InflexNum$ | 89.23 | 91.54 | 83.85 | 97.69 | **90.58** |
| | $Fs_1$+$Fs_2$ | 86.15 | 89.23 | 91.54 | 90.77 | **89.42** |
| $MFCC_{120}$ | $MS_{fc}$ | 90.00 | 93.08 | 96.15 | 96.15 | **93.85** |
| | $MS_{fc}$+$VarCoef$ | 90.00 | 93.85 | 95.38 | 97.69 | **94.23** |
| | $MS_{fc}$+$Mvar$ | 94.62 | 93.85 | 98.46 | 97.69 | **96.15** |
| | $Fs_1$ | 89.23 | 94.62 | 96.92 | 96.92 | **94.42** |
| | $Fs_1$+$AvgSlopeDif$ | 91.54 | 87.69 | 92.31 | 96.92 | **92.12** |
| | $Fs_1$+$InflexNum$ | 83.08 | 96.92 | 90.77 | 96.15 | **91.73** |
| | $Fs_1$+$Fs_2$ | 83.08 | 93.85 | 88.46 | 98.46 | **90.96** |

Table 5.10: Accuracy results for SVM when using a frame size of 1024 and window length of 512. With $Fs_1$:{$MS_{fc}$;$VarCoef$;$Mvar$} and $Fs_2$:{$AvgSlopeDif$;$InflexNum$}

| | | **SVM** (frame size=1024, window len=512) | | | | |
|---|---|---|---|---|---|---|
| | | Common dolphin (%) | Bottlenose dolphin (%) | Spotted dolphin (%) | Striped dolphin (%) | $M_{acc}$ (%) |
| $STFT$ | $MS_{fc}$ | 75.38 | 89.23 | 94.62 | 87.69 | **86.73** |
| | $MS_{fc}$+$VarCoef$ | 80.77 | 86.92 | 98.46 | 89.23 | **88.85** |
| | $MS_{fc}$+$Mvar$ | 83.85 | 90.00 | 95.38 | 87.69 | **89.23** |
| | $Fs_1$ | 83.08 | 90.00 | 99.23 | 89.23 | **90.38** |
| | $Fs_1$+$AvgSlopeDif$ | 76.92 | 81.54 | 94.62 | 86.92 | **85.00** |
| | $Fs_1$+$InflexNum$ | 79.23 | 86.92 | 90.00 | 86.92 | **85.77** |
| | $Fs_1$+$Fs_2$ | 81.54 | 80.77 | 91.54 | 85.38 | **84.81** |
| $MFCC_{20}$ | $MS_{fc}$ | 90.00 | 86.15 | 93.85 | 96.15 | **91.54** |
| | $MS_{fc}$+$VarCoef$ | 86.15 | 86.15 | 93.08 | 96.92 | **90.58** |
| | $MS_{fc}$+$Mvar$ | 75.38 | 86.15 | 91.54 | 95.38 | **87.12** |
| | $Fs_1$ | 83.85 | 90.00 | 97.69 | 94.62 | **91.54** |
| | $Fs_1$+$AvgSlopeDif$ | 81.54 | 88.46 | 94.62 | 96.15 | **90.19** |
| | $Fs_1$+$InflexNum$ | 89.23 | 89.23 | 96.92 | 99.23 | **93.65** |
| | $Fs_1$+$Fs_2$ | 82.31 | 90.00 | 95.38 | 93.85 | **90.38** |
| $MFCC_{40}$ | $MS_{fc}$ | 90.77 | 95.38 | 96.92 | 96.15 | **94.81** |
| | $MS_{fc}$+$VarCoef$ | 88.46 | 96.15 | 95.38 | 93.85 | **93.46** |
| | $MS_{fc}$+$Mvar$ | 85.38 | 93.85 | 90.77 | 94.62 | **91.15** |
| | $Fs_1$ | 91.54 | 92.31 | 87.69 | 95.38 | **91.73** |
| | $Fs_1$+$AvgSlopeDif$ | 85.38 | 85.38 | 93.85 | 94.62 | **89.81** |
| | $Fs_1$+$InflexNum$ | 91.54 | 90.00 | 97.69 | 92.31 | **92.88** |
| | $Fs_1$+$Fs_2$ | 86.92 | 82.31 | 86.15 | 92.31 | **86.92** |
| $MFCC_{60}$ | $MS_{fc}$ | 92.31 | 91.54 | 93.82 | 98.46 | **94.04** |
| | $MS_{fc}$+$VarCoef$ | 94.62 | 93.85 | 88.46 | 92.31 | **92.31** |
| | $MS_{fc}$+$Mvar$ | 91.54 | 93.08 | 94.62 | 94.62 | **93.46** |
| | $Fs_1$ | 95.38 | 92.31 | 92.31 | 93.85 | **93.46** |
| | $Fs_1$+$AvgSlopeDif$ | 88.46 | 89.23 | 93.08 | 94.62 | **91.35** |
| | $Fs_1$+$InflexNum$ | 87.69 | 94.62 | 94.62 | 95.38 | **93.08** |
| | $Fs_1$+$Fs_2$ | 86.15 | 85.38 | 84.62 | 96.15 | **88.08** |
| $MFCC_{120}$ | $MS_{fc}$ | 90.00 | 93.08 | 95.38 | 92.31 | **92.69** |
| | $MS_{fc}$+$VarCoef$ | 86.92 | 95.38 | 92.31 | 90.00 | **91.15** |
| | $MS_{fc}$+$Mvar$ | 86.15 | 93.08 | 94.62 | 92.31 | **91.54** |
| | $Fs_1$ | 91.54 | 93.08 | 94.62 | 86.92 | **91.54** |
| | $Fs_1$+$AvgSlopeDif$ | 89.23 | 85.38 | 92.31 | 94.62 | **90.38** |
| | $Fs_1$+$InflexNum$ | 82.31 | 91.54 | 90.77 | 91.54 | **89.04** |
| | $Fs_1$+$Fs_2$ | 86.15 | 91.54 | 85.38 | 90.77 | **88.46** |

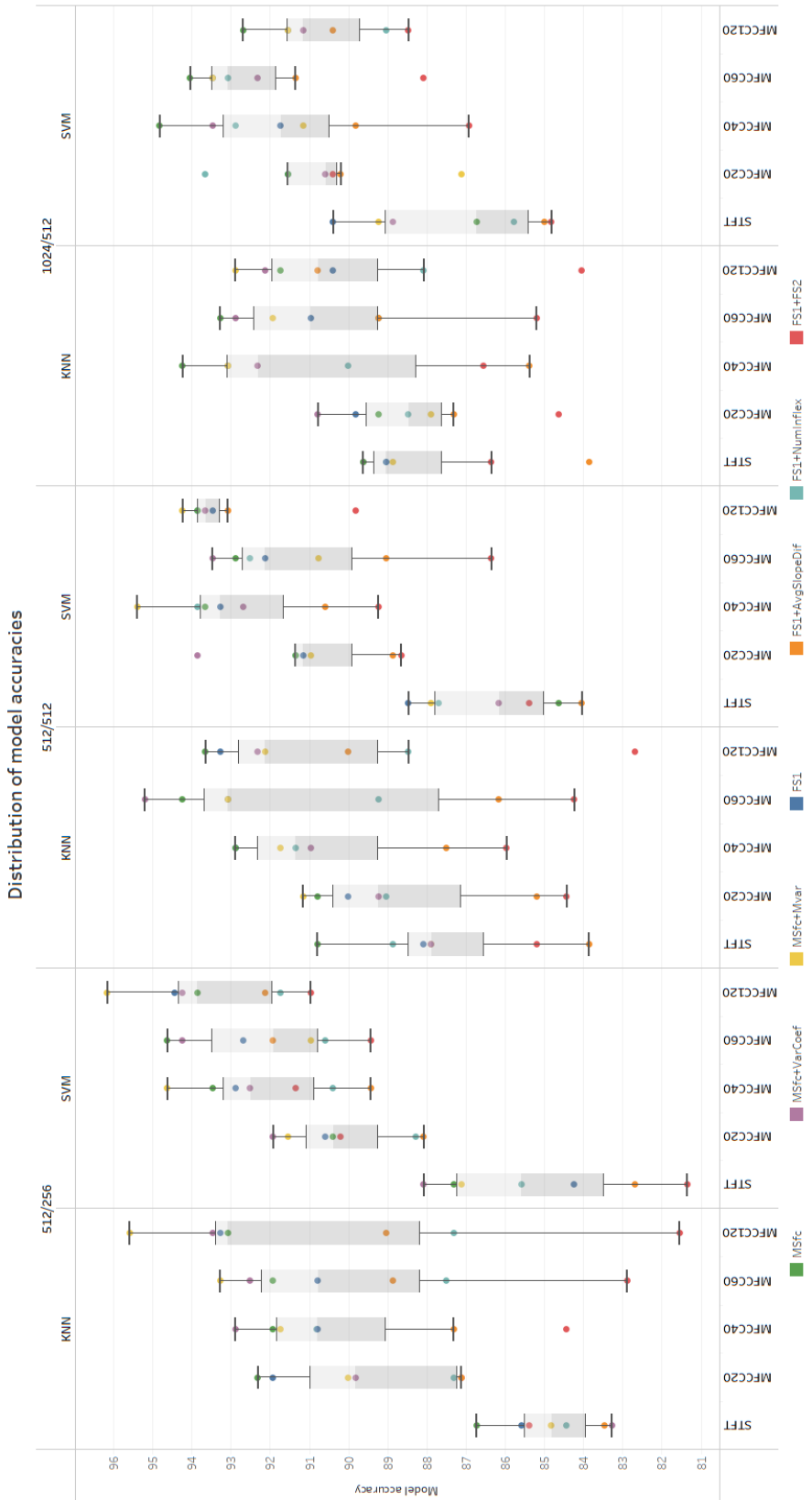



Figure 5.1: Distribution of the obtained model accuracy regarding the classifier and time-frequency representation used for each feature combination test (colored dots).

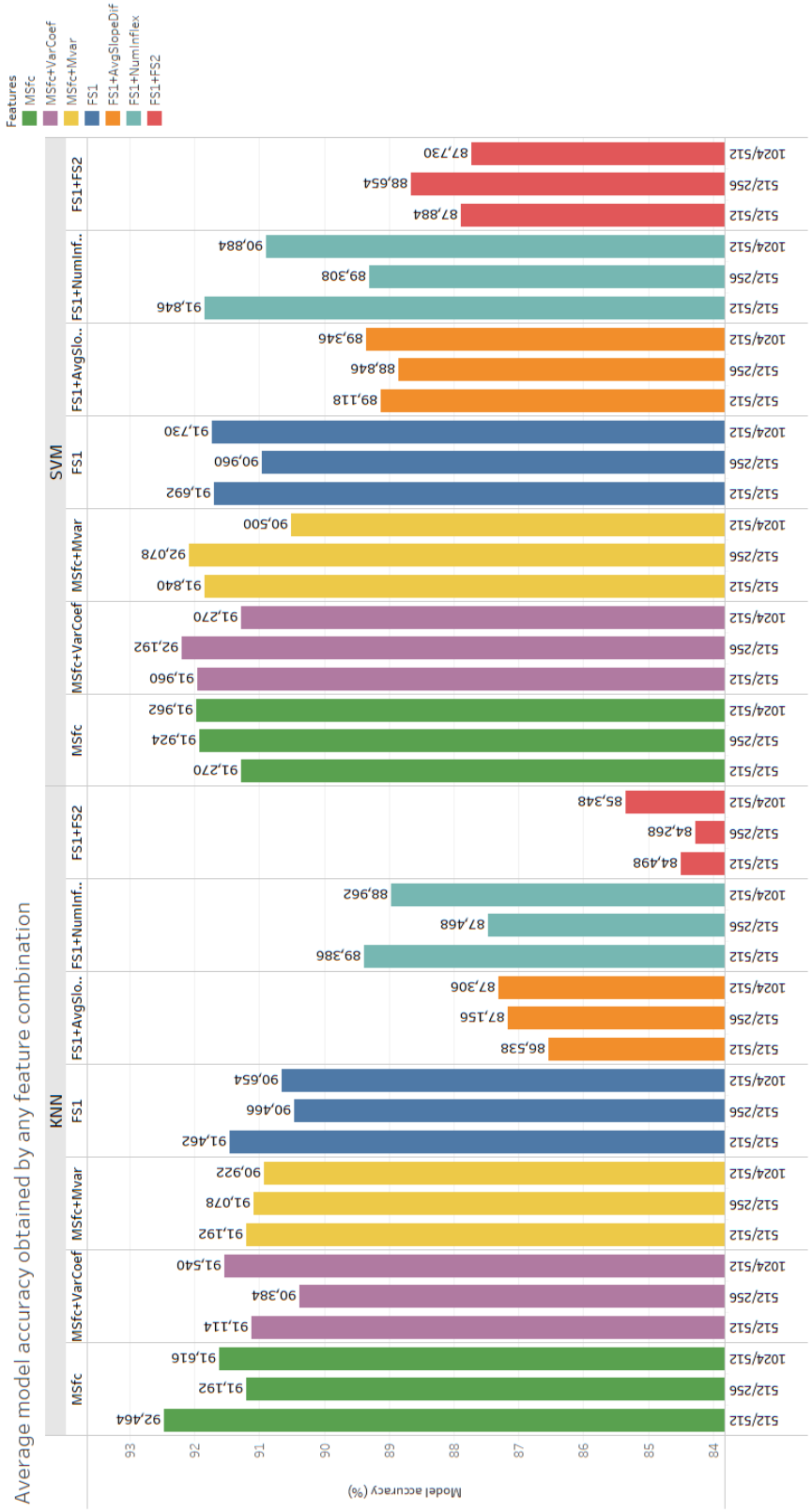



Figure 5.2: Averaged model accuracy results obtained by all feature combinations for different STFT parameters and classifier.

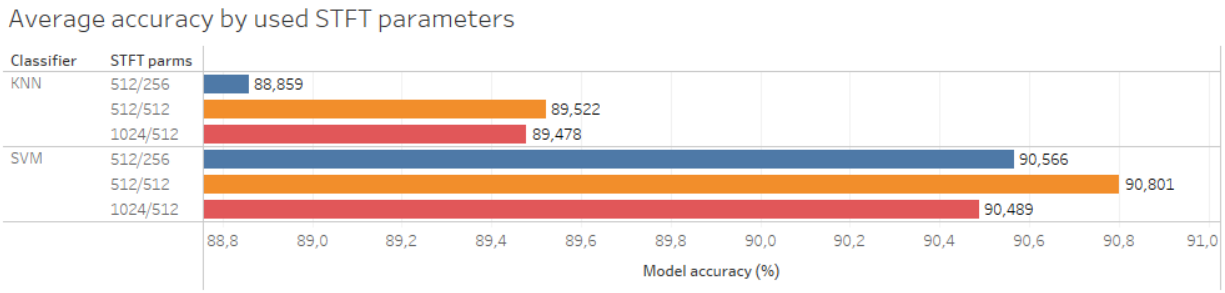



Figure 5.3: Averaged model accuracy of all tests which used the same STFT parameter combination.

# 6

# Conclusion

In this dissertation, we propose the development of a feature extraction and classification method which is able to obtain several different features capable of distinguishing vocalizations of small species of dolphins indigenous to the Madeira archipelago. The development of these tailor made features and their underlying classifier presents itself as an useful and efficient way to support marine biologists with the conservation efforts of these species. These will facilitate the identification of species by their vocalizations in any obtained PAM recordings, something which currently relies on visual confirmation at the time of the recording or is a laborious manual task to perform. This chapter will provide an overview of the contributions which resulted from this work and some possible work to be developed in the future.

In order to obtain the most accurate classification model for the intended task, we performed a comparative study evaluating different time-frequency representations (with different parameterizations) from where to derive the features, as well as different supervised classification approaches. As time-frequency representations, the use of MFCC (proposed in [49, 55]) and STFT (proposed in [9]) were considered. As far as the classification models taken into consideration, SVM and K-NN were used. However, the use of several CNN models was still tested (in an end-to-end way) on vocalization spectrograms, to serve as a control comparison group due to their proven capability of extracting useful low-level features which make up vocalization contours [15, 48].

In order to obtain a more complete and precise way to distinguish the intended species vocalizations, we propose a novel set of features that captures different properties of the vocalizations, mostly their predominant frequency components (Section 4.4.1) and vocalization contours (Section 4.4.2). As some of the resulting features might be considered as uninformative or describe some information already covered by other features in the feature space, we propose the usage of a dimensionality reduction method under the form of ICA to mitigate this.

The derived features were validated by testing them in several different combination and in diverse test settings (i.e. using different time-frequency representations and classifiers). The obtained results (Section 5), highlight the viability of the proposed model

and features, managing to achieve a maximum classification accuracy of 96.15% (Common dolphin: 94.62%; Bottlenose dolphin: 93.85%; Spotted dolphin: 98.46%; Striped dolphin: 97.69%) while using 120 MFCC's (frame size=512, window size=256) on a SVM classifier using $Ms_{fc}$ (Section 4.4.1.1) and $Mvar$ (Section 4.4.1.3) as features. This general accuracy surpassed the results of previous studies on the task of dolphin classification for the best of our knowledge. These results also outperform most of the best individual species accuracies achieved in previous works, with them being: Common dolphin: 90.05% [57]; Bottlenose dolphin: 100.0% [55]; Spotted dolphin: 95.00% [55]; Striped dolphin: 34.1% [52]. However, it must be mentioned that some of these previous works performed classification on multiple cetacean species (not only on dolphins), thus making a direct comparison with our obtained results should be made with caution.

Even though, the best model accuracy achieved did not use the entire feature set and there was a slight decline in overall accuracy when using more features (Figure 5.2), at some point, the introduction of any of the developed features managed to contribute to an improvement of the individual accuracy of some species (e.g. introduction of the $InflexNum$ feature on the SVM tests, in Table 5.8 leads to a maximum individual accuracy for the Bottlenose dolphin when using STFT, 60 MFCC and 120 STFT as time-frequency representations). Due to this, one can still make a case for the usage of any of the proposed features.

Regarding the other parameters tested in the proposed comparative study, the obtained results suggest that for future works, the usage of MFCC as a time-frequency representation should be prioritized over the usage of STFT, as it leads to better accuracy results (Figure 5.1). The usage of an increasing number of MFCC can also contribute to better results. As for the classifiers, SVM outperformed the K-NN classifier by achieving a better average model accuracy of 90.801% against 89.522% (seen on Figure 5.3). This better performance can also be attributed to smaller fluctuations of the model's accuracy when using more features, something made apparent by comparing the results of both models in Figure 5.2. Also, both SVM and K-NN outperformed all of the obtained results from the tested CNN models, which achieved a maximum model accuracy of 85.89% ($CM_2$ trained with a batch size of 64). Finally, the impact the used time-frequency representation parameters (frame and window size) had on the model's accuracy could not be totally defined, mostly due to the average performance proximity of all the tested combinations (Figure 5.3).

This work also culminated in the submission of a conference paper "Extracting Vocalization Features To Recognize Small Dolphin Species", explaining the technical approach used for the estimation of the proposed features. The paper can be seen in Annex I.

## 6.1 Future work

The work developed in this dissertation, might serve as a solid foundation for the continuous study of bioacoustic classification of several different cetacean species. Having

this work established several different features, which succeeded in distinguishing four different small dolphin species, further feature sets can be introduced with the aim of characterizing other different cetacean species vocalizations. With the presented work be essentially a proof of concept for the task of dolphin vocalization distinction, these employed methods can be integrated into a custom piece of software in the future, which could allow for real-time or close to real-time vocalization detection. For this purpose, a real-time vocalization instance detection system should also be developed, which would allow for real-time vocalization detection and segmentation. This approach would replace the preprocessing stage in our current model, which dealt with the segmentation of the raw vocalization signals empirically (discarding the generated segments which did not contained vocalizations). Also, the employment of additional classification methods, such as the use of ANN, might be of interest in the continuous development of this classification model.

# Bibliography


[1] J. G. Mead. "Cetaceans". In: *Brittanica Encyclopedia* (May 2020). URL: https://www.britannica.com/animal/cetacean. (accessed: 2.02.2021).

[2] L. Sayigh. "Cetacean Acoustic Communication". In: *Biocommunication of Animals* (Nov. 2013), pp. 275–297. DOI: 10.1007/978-94-007-7414-8_16.

[3] C. Bergler et al. "ORCA-SPOT: An Automatic Killer Whale Sound Detection Toolkit Using Deep Learning". In: *Scientific Reports* 9 (July 2019). DOI: 10.1038/s41598-019-47335-w.

[4] *Social Influences on Vocal Development*. Cambridge University Press, 1997. DOI: 10.1017/CBO9780511758843.

[5] L. Kyhn et al. "Feeding at a high pitch: Source parameters of narrow band, high-frequency clicks from echolocating off-shore hourglass dolphins and coastal Hector's dolphins". In: *The Journal of the Acoustical Society of America* 125 (Apr. 2009), pp. 1783–91. DOI: 10.1121/1.3075600.

[6] M. Simon et al. "Singing behavior of fin whales in the Davis Strait with implications for mating, migration and foraging". In: *The Journal of the Acoustical Society of America* 128.5 (2010), pp. 3200–3210. DOI: 10.1121/1.3495946. eprint: https://doi.org/10.1121/1.3495946. URL: https://doi.org/10.1121/1.3495946.

[7] V. Janik. "Cetacean vocal learning and communication". In: *Current Opinion in Neurobiology* 28 (Oct. 2014), pp. 60–65. DOI: 10.1016/j.conb.2014.06.010.

[8] L. Shamir et al. "Classification of large acoustic datasets using machine learning and crowdsourcing: Application to whale calls". In: *The Journal of the Acoustical Society of America* 135 (Feb. 2014), pp. 953–962. DOI: 10.1121/1.4861348.

[9] M. Bahoura and Y. Simard. "Blue whale calls classification using short-time Fourier and wavelet packet transforms and artificial neural network". In: *Digital Signal Processing* 20 (July 2010), pp. 1256–1263. DOI: 10.1016/j.dsp.2009.10.024.

[10] H. Schröter et al. "Segmentation, Classification, and Visualization of Orca Calls Using Deep Learning". In: May 2019, pp. 8231–8235. DOI: 10.1109/ICASSP.2019.8683785.

[11] D. Schiller et al. "Relevance-Based Feature Masking: Improving Neural Network Based Whale Classification Through Explainable Artificial Intelligence". In: Sept. 2019, pp. 2423–2427. DOI: 10.21437/Interspeech.2019-2707.

[12] C. Bergler et al. "ORCA-CLEAN: A Deep Denoising Toolkit for Killer Whale Communication". In: *Proc. Interspeech 2020*. 2020, pp. 1136–1140. DOI: 10.21437/Interspeech.2020-1316. URL: http://dx.doi.org/10.21437/Interspeech.2020-1316.

[13] L. Tacioli, L. Toledo, and C. Medeiros. "An Architecture for Animal Sound Identification based on Multiple Feature Extraction and Classification Algorithms". In: Feb. 2020, pp. 29–36. DOI: 10.5753/bresci.2017.9919.

[14] N. Fisheries. *Passive Acoustic Research in the Atlantic Ocean*. Oct. 2020. URL: https://www.fisheries.noaa.gov/new-england-mid-atlantic/endangered-species-conservation/passive-acoustic-research-atlantic-ocean. (accessed: 2.02.2021).

[15] P. Bermant et al. "Deep Machine Learning Techniques for the Detection and Classification of Sperm Whale Bioacoustics". In: *Scientific Reports* 9 (Aug. 2019), pp. 1–10. DOI: 10.1038/s41598-019-48909-4.

[16] L. Freitas et al. *Cetáceos no Arquipélago da Madeira*. Multiponto S.A., 2004.

[17] W. A. Yost. *Fundamentals of hearing: an instructor's handbook*. 4th edition. Academic Press, 2000.

[18] L. Krippahl. *Aprendizagem Automática (Machine Learning) - Lecture Notes*. 2018.

[19] L. Krippahl. *Aprendizagem com Dados Não Estruturados - Lecture Notes*. 2019.

[20] M. Ariño. *Sound Basics: Propagation, Amplitude, Frequency and Timbre*. 2014. URL: https://pt.slideshare.net/menabellica/sound-basics-propagation-amplitude-frequency-and-timbre/10. (accessed: 15.01.2021).

[21] M. René. *The emission, propagation and perception of sound*. Feb. 2019. URL: https://www.encyclopedie-environnement.org/en/physics/emission-propagation-and-perception-of-sound/#6_Sound_propagation_in_water_and_solids. (accessed: 16.01.2021).

[22] Libretexts. *17.3: Speed of Sound*. Libretexts, Nov. 2020. URL: https://phys.libretexts.org/Bookshelves/University_Physics/Book%5C%3A_University_Physics_(OpenStax)/Map%5C%3A_University_Physics_I_-_Mechanics_Sound_Oscillations_and_Waves_(OpenStax)/17%5C%3A_Sound/17.03%5C%3A_Speed_of_Sound. (accessed: 16.01.2021).

[23] *The Underwater Propagation of Sound and its Applications*. 2012. URL: https://sites.dartmouth.edu/dujs/2012/03/11/the-underwater-propagation-of-sound-and-its-applications/. (accessed: 16.01.2021).

[24] *Sound absorption*. 2002. URL: https://encyclopedia2.thefreedictionary.com/sound+absorption. (accessed: 16.01.2021).

[25] A. alearningaday. Dec. 2019. URL: https://alearningaday.blog/2019/12/06/internalizing-the-sine-wave/. (accessed: 15.01.2021).

[26] Des et al. *Sinusoidal and Random Vibration Testing Primer*. Oct. 2017. URL: https://www.desolutions.com/blog/2013/04/sinusoidal-and-random-vibration-testing-primer/. (accessed: 16.01.2021).

[27] A. Carter. *Learn About the AC Phase Difference*. Mar. 2018. URL: https://www.eeweb.com/learn-about-the-ac-phase-difference/. (accessed: 16.01.2021).

[28] K. Chaudhary. *Understanding Audio data, Fourier Transform, FFT, Spectrogram and Speech Recognition*. July 2020. URL: https://towardsdatascience.com/understanding-audio-data-fourier-transform-fft-spectrogram-and-speech-recognition-a4072d228520. (accessed: 18.01.2021).

[29] *Seeing sound: What is a spectrogram?* Sept. 2018. URL: https://blogs.bl.uk/sound-and-vision/2018/09/seeing-sound-what-is-a-spectrogram.html. (accessed: 16.01.2021).

[30] *PyFilterbank documentation*. 2014. URL: http://siggigue.github.io/pyfilterbank/melbank.html. (accessed: 24.02.2021).

[31] W. Wang et al. "Feature extraction of underwater target in auditory sensation area based on MFCC". In: *2016 IEEE/OES China Ocean Acoustics (COA)*. 2016, pp. 1–6. DOI: 10.1109/COA.2016.7535736.

[32] *Lowpass filter - MathWorks*. URL: https://www.mathworks.com/help/signal/ref/lowpass.html?lang=en. (accessed: 19.01.2021).

[33] R. Berwick. In: *An Idiot's Guide to Support Vector Machines (SVMs)* (2003). URL: http://web.mit.edu/6.034/wwwbob/svm-notes-long-08.pdf. (accessed: 21.01.2021).

[34] D. Tao et al. "Asymmetric Bagging and Random Subspace for Support Vector Machines-Based Relevance Feedback in Image Retrieval". In: *IEEE transactions on pattern analysis and machine intelligence* 28 (Aug. 2006), pp. 1088–99. DOI: 10.1109/TPAMI.2006.134. (accessed: 21.01.2021).

[35] *Support Vector Machines with the mlr package*. Oct. 2019. URL: https://www.r-bloggers.com/2019/10/support-vector-machines-with-the-mlr-package/. (accessed: 22.01.2021).

[36] G. Seif. "The 5 Clustering Algorithms Data Scientists Need to Know". In: *Medium* (Dec. 2020). URL: https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68. (accessed: 23.01.2021).

[37] G. Seif. *The 5 Clustering Algorithms Data Scientists Need to Know*. Jan. 2021. URL: https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68. (accessed: 25.01.2021).

[38] M. Lopes. *Dimensionality Reduction - Does PCA really improve classification outcome?* Jan. 2017. URL: https://towardsdatascience.com/dimensionality-reduction-does-pca-really-improve-classification-outcome-6e9ba21f0a32. (accessed: 24.01.2021).

[39] A. Hyvärinen and E. Oja. "Independent component analysis: algorithms and applications". In: *Neural Networks* 13.4-5 (June 2000), pp. 411–430. DOI: 10.1016/s0893-6080(00)00026-5. URL: https://doi.org/10.1016/s0893-6080(00)00026-5.

[40] A. Tharwat. "Independent component analysis: An introduction". In: 17.2 (Aug. 2020), pp. 222–249. DOI: 10.1016/j.aci.2018.08.006. URL: https://doi.org/10.1016/j.aci.2018.08.006.

[41] P. C. Bermant et al. "Deep Machine Learning Techniques for the Detection and Classification of Sperm Whale Bioacoustics". In: *Scientific Reports* 9.1 (Aug. 2019), p. 12588. ISSN: 2045-2322. DOI: 10.1038/s41598-019-48909-4. URL: https://doi.org/10.1038/s41598-019-48909-4.

[42] A. Oppermann. *What is Deep Learning and How does it work?* Aug. 2020. URL: https://towardsdatascience.com/what-is-deep-learning-and-how-does-it-work-2ce44bb692ac. (accessed: 23.01.2021).

[43] C. Dorian et al. "Bi-class classification of humpback whale sound units against complex background noise with Deep Convolution Neural Network". In: (Mar. 2017).

[44] S. Saha. *A Comprehensive Guide to Convolutional Neural Networks-the ELI5 way*. Dec. 2018. URL: https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53. (accessed: 24.01.2021).

[45] J. Brownlee. *How Do Convolutional Layers Work in Deep Learning Neural Networks?* Apr. 2020. URL: https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/. (accessed: 24.01.2021).

[46] Prabhu. *Understanding of Convolutional Neural Network (CNN) - Deep Learning*. Nov. 2019. URL: https://medium.com/@RaghavPrabhu/understanding-of-convolutional-neural-network-cnn-deep-learning-99760835f148. (accessed: 24.01.2021).

[47] R. Miralles Ricós et al. "Automatic detection and classification of beluga whale vocalizations". In: *Advances in Applied Acoustics*. Vol. 2. Science and Engineering Publishing Company, 2013, pp. 61–70.

[48] L. Zhang et al. "Large-Scale Whale-Call Classification by Transfer Learning on Multi-Scale Waveforms and Time-Frequency Features". In: *Applied Sciences* 9 (Mar. 2019), p. 1020. DOI: 10.3390/app9051020.

[49] J. Noda, D. Sanchez-Rodriguez, and C. Travieso. "A Methodology Based on Bioacoustic Information for Automatic Identification of Reptiles and Anurans". In: July 2018. ISBN: 978-1-78923-400-8. DOI: 10.5772/intechopen.74333.

[50] A. Pedroza et al. "A comparative between Mel Frequency Cepstral Coefficients (MFCC) and Inverse Mel Frequency Cepstral Coefficients (IMFCC) features for an Automatic Bird Species Recognition System". In: Nov. 2018, pp. 1–4. DOI: 10.1109/LA-CCI.2018.8625230.

[51] S. Baumann-Pickering et al. "Discriminating features of echolocation clicks of melon-headed whales (Peponocephala electra), bottlenose dolphins (Tursiops truncatus), and Gray's spinner dolphins (Stenella longirostris longirostris)". In: *The Journal of the Acoustical Society of America* 128.4 (Oct. 2010), pp. 2212–2224. DOI: 10.1121/1.3479549. URL: https://doi.org/10.1121/1.3479549.

[52] D. Gillespie et al. "Automatic detection and classification of odontocete whistles". In: *The Journal of the Acoustical Society of America* 134.3 (Sept. 2013), pp. 2427–2437. DOI: 10.1121/1.4816555. URL: https://doi.org/10.1121/1.4816555.

[53] M. Azzolin et al. "Combining whistle acoustic parameters to discriminate Mediterranean odontocetes during passive acoustic monitoring". In: *The Journal of the Acoustical Society of America* 135.1 (Jan. 2014), pp. 502–512. DOI: 10.1121/1.4845275. URL: https://doi.org/10.1121/1.4845275.

[54] T.-H. Lin and L.-S. Chou. "Automatic classification of delphinids based on the representative frequencies of whistles". In: *The Journal of the Acoustical Society of America* 138.2 (Aug. 2015), pp. 1003–1011. DOI: 10.1121/1.4927695. URL: https://doi.org/10.1121/1.4927695.

[55] M. Nadir et al. "Marine Mammals Classification using Acoustic Binary Patterns". In: (2020). DOI: 10.24425/AOA.2020.135278. URL: https://journals.pan.pl/dlibra/publication/135278/edition/118263/content.

[56] F. Erbs, S. H. Elwen, and T. Gridley. "Automatic classification of whistles from coastal dolphins of the southern African subregion". In: *The Journal of the Acoustical Society of America* 141.4 (Apr. 2017), pp. 2489–2500. DOI: 10.1121/1.4978000. URL: https://doi.org/10.1121/1.4978000.

[57] T. O. S. Amorim et al. "Integrative bioacoustics discrimination of eight delphinid species in the western South Atlantic Ocean". In: *Plos One* 14.6 (2019). DOI: 10.1371/journal.pone.0217977.

[58] *Pamguard: Open source software for passive acoustic monitoring*. URL: https://www.pamguard.org/,%20year=%7B2012%7D.

[59] A. Kawade, R. Shastri, and S. Vidhya. "Denoising Techniques for Underwater Ambient Noise". In: 2 (Feb. 2016), pp. 2349–784.

[60] D. B. V. "Denoising Methods for Underwater Acoustic Signal". In: Nov. 2017. ISBN: 978-953-51-3609-5. DOI: 10.5772/intechopen.69027.

[61] N. Priyadarshani et al. "Birdsong Denoising Using Wavelets". In: *PLOS ONE* 11 (Jan. 2016), e0146790. DOI: 10.1371/journal.pone.0146790.

[62] *Watkins Marine Mammal Sound Database*. URL: https://whoicf2.whoi.edu/science/B/whalesounds/index.cfm. (accessed: 14.02.2021).

[63] R. Oshana. "4 - Overview of Digital Signal Processing Algorithms". In: *DSP Software Development Techniques for Embedded and Real-Time Systems*. Ed. by R. Oshana. Embedded Technology. Burlington: Newnes, 2006, p. 66. ISBN: 978-0-7506-7759-2. DOI: https://doi.org/10.1016/B978-075067759-2/50006-5.

[64] I. C. Ansmann. "The whistle repertoire and acoustic behaviour of short-beaked common dolphins, Delphinus delphis, around the British Isles, with applications for acoustic surveying". PhD thesis. University of Wales Bangor, 2005.

[65] E. Papale et al. "Dolphins Adjust Species-Specific Frequency Parameters to Compensate for Increasing Background Noise". In: 10.4 (Apr. 2015). Ed. by E. J. Warrant, e0121711. DOI: 10.1371/journal.pone.0121711. URL: https://doi.org/10.1371/journal.pone.0121711.

[66] A. F. Azevedo et al. "Whistles emitted by Atlantic spotted dolphins (Stenella frontalis) in southeastern Brazil". In: 127.4 (Apr. 2010), pp. 2646–2651. DOI: 10.1121/1.3308469. URL: https://doi.org/10.1121/1.3308469.

[67] *All about bottlenose dolphin communication*. URL: https://seaworld.org/animals/all-about/bottlenose-dolphin/communication/#:~:text=The%5C%20frequency%5C%20of%5C%20the%5C%20sounds,frequencies%5C%20less%5C%20than%5C%2040%5C%20kHz.. (accessed: 14.07.2021).

[68] W. W. L. A. (auth.) "The Sonar of Dolphins". In: 1st ed. Springer-Verlag New York, 1993. Chap. 7 - Characteristics of Dolphin Sonar Signals, p. 134.

[69] R. J. McAulay and T. Quatieri. "Sinusoidal Coding". In: *Speech Coding and Synthesis*. Ed. by W. Kleijn and K. Paliwal. Elsevier Science B.V., 1995. Chap. 4, pp. 121–173.

[70] R. J. McAulay and T. Quatieri. "Speech analysis/synthesis based on a sinusoidal representation". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* ASSP-34.4 (Aug. 1986), pp. 744–754.

[71] J. Smith and X. Serra. "PARSHL: an analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation". In: *Proceedings of the International Computer Music Conference*. 1987, pp. 290–297.

[72] M. Ester et al. "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. KDD'96. Portland, Oregon: AAAI Press, 1996, pp. 226–231.

[73] S. Maneewongvatana and D. M. Mount. "It's Okay to Be Skinny, If Your Friends Are Fat". In: *Center for Geometric Computing 4th Annual Workshop on Computational Geometry*. 1999.

[74] C. Szegedy et al. "Rethinking the Inception Architecture for Computer Vision". In: *CoRR* abs/1512.00567 (2015). arXiv: 1512.00567. URL: http://arxiv.org/abs/1512.00567.

[75] W. Commons. *K-fold cross validation EN*. 2019. URL: https://commons.wikimedia.org/wiki/File:K-fold_cross_validation_EN.svg.

[76] M. Mahdianpari et al. "Very Deep Convolutional Neural Networks for Complex Land Cover Mapping Using Multispectral Remote Sensing Imagery". In: *Remote Sensing* 10.7 (July 2018), p. 1119. DOI: 10.3390/rs10071119. URL: https://doi.org/10.3390/rs10071119.

# Annex: Submitted conference paper

## EXTRACTING VOCALIZATION FEATURES TO RECOGNIZE SMALL DOLPHIN SPECIES

*Luís Rosário**, *Sofia Cavaco**, *Luís Freitas*†, *Philippe Verborgh*†, *Joaquim Silva**

* NOVA LINCS, Departamento de Informática Faculdade de Ciências e Tecnologia, Universidade NOVA de Lisboa, 2829-516 Caparica, Portugal
† Madeira Whale Museum, 9200-031 Caniçal, Madeira, Portugal

**ABSTRACT**

The identification of small dolphin species by their vocalizations, still remains a challenging task due to their underlying similar vocal signatures and frequency modulation patterns, which difficult their distinction. To overcome this, a new feature set is presented which focuses in capturing both the vocalization's predominant frequency range and other higher level details in the spectral contour, which are useful to distinguish some species. The proposed features are computed from two distinct time-frequency representations of the vocalizations: the short time Fourier transform and Mel frequency cepstral coefficients. Using these features in two popular classifiers (K-nearest neighbors and support vector machines) we obtained a model accuracy of 93.85% which shows improvement over previous studies.

***Index Terms***— Bioacoustic Classification, Cetaceans, Bioacoustic Signal Processing, Supervised Classification

### 1. INTRODUCTION AND RELATED WORK

Passive acoustic monitoring (PAM) is a cost-effective method to detect the presence of cetaceans. Long-term deployments in specific areas can inform on the presence of certain species and their activity. However, the inability to identify with confidence many cetacean species, based on their vocalizations, has limited its contribution to the research, conservation and management of impacts on cetaceans. The development of automated tools to recognize these cetacean species from acoustic recordings would bring considerable biological value to the data collected by PAM and allow the efficient processing of large amounts of data produced by this method.

Signal processing techniques that analyse cetacean vocalizations have been used in tasks ranging from the classification of different calls from a single species [1, 2, 3], to the distinction between different cetacean species which include whales and dolphins [4, 5, 6, 7]. These works suggest using a variety of distinct features such as several statistical acoustic features [5, 6] or Mel frequency cepstral coefficients (MFCCs) [3, 8]. However, when we focus on dolphin vocalizations alone, these studies miss to achieve an effective and accurate way to properly distinguish the species, as shown by their results, with accuracies that range from 37.3% to 93% [7], 43.1% to 69.7% [6], 34.1% to 68.4% [5] and 54% to 75% [4]. Recent works present models with better accuracy values (up to 90.4%), although still with some variance among species, based on a limited data set or centered on a single recording location or largely based on an existing software [9, 10, 8].

This paper proposes a new set of features derived from spectral representations of dolphin vocalizations that is capable of distinguishing four small dolphin species while using recording locations ranging from North America to Southern Europe. This set of features facilitates the distinction of these species, in spite of their underlying similar vocal signatures. This claim is validated by tests made with popular classifiers such as K-nearest neighbors (KNN) and support vector machines (SVM).

The proposed features are derived from the time-frequency representation of the vocalizations obtained from two different approaches: the first uses the magnitude spectrogram of the short time Fourier transform (STFT), while the second uses MFCCs, which have been shown as a viable approach in the domain of bioacoustic classification for several species [11, 12], including cetaceans [13, 14].

Experimental results show the viability of the proposed feature set in distinguishing small dolphin species, while reaching a global model accuracy of 93.85%. The individual species accuracy ranges from 88.46% to 96.92%, which shows improvement over previous studies.

### 2. FEATURES

The proposed feature set is composed of five different features. Their extraction process is performed over the time-frequency representation of the vocalizations. This may either be a magnitude spectrogram of the STFT (computed with a Hanning window of length 512 and an overlap of 256), or the MFCCs matrix (computed with the same parameters as those used for the STFT).

The proposed features encompass two distinct approaches to the analysis of the vocalizations. The first approach, which culminates in the creation of three **frequency analysis features**, tries to capture the predominant frequency components. This can be a good indicator of the frequency distribution and range of a species' vocalizations.

However, as the vocalizations of most dolphin species boast a wide frequency range, which overlap for many species [15] (fig.1), in theory these three features may not be sufficient to properly distinguish vocalizations of distinct dolphin species. To overcome this limitation, we developed two additional features, the **contour analysis features**, which intent to express some of the higher-level details in the pattern of the vocalization's spectral representation. It must be mentioned that for obtaining this feature subset, the time-frequency representation is computed only from the STFT as the MFCCs are incapable of maintaining the level of detail in the vocalization pattern that the STFT offers. For simplicity, we refer to the frequency bins in the spectrogram and the coefficients in the MFCC matrix as frequency components.

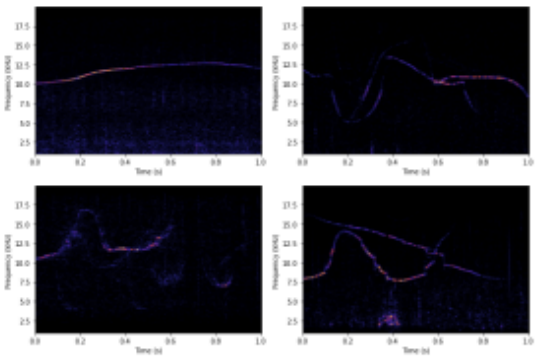


**Fig. 1**. Spectrograms of four different dolphin species' vocalizations. From left to right, top to bottom: *Delphinus delphis*, *Tursiops truncatus*, *Stenella frontalis*, *Stenella coeruleoalba*.

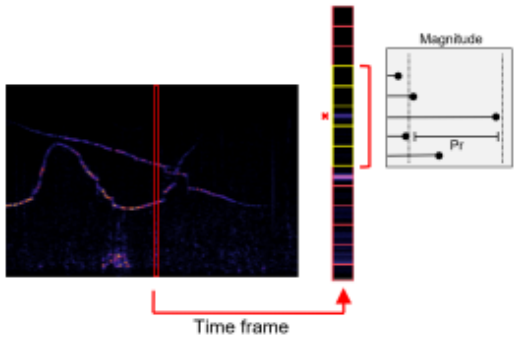



**Fig. 2**. Detection of peak in a time frame with a minimum required prominence. The selected peak ($x$) will be valid if the prominence (Pr) to its lowest contour is at least the 95th percentile of the magnitudes present in that time frame.

### 2.1. Frequency analysis features

This feature subset ($Fs_1$) is composed of three distinct features. The first feature in this subset corresponds to each frequency component's magnitude sum $MS_{fc}(f)$, where each entry is given by:

$$MS_{fc}(f) = \sum_{t=0}^{\|T\|} m(f,t) \tag{1}$$

where the magnitude sum of the frequency component $f$ is given by the cumulative sum of the magnitudes $m$ on that frequency component for a time duration $T$. This feature in conjunction with the following ones in this subset, will provide the essential information regarding the predominant frequency components where vocalizations may lie, as the magnitude of the instances where a vocalization occurs will contributes greatly to the magnitude sum of those components

The following feature is the *variance coefficient* of a signal's spectral representation. In order to derive this metric, we firstly need to determine the average frequency component magnitude sum ($\overline{MS_{fc}}$), which is achieved by the following expression:

$$\overline{MS_{fc}} = \frac{\sum_{f \in F} MS_{fc}(f)}{\|F\|} \tag{2}$$

The average frequency component magnitude sum allows to estimate the standard deviation of the overall frequency component magnitude sum of the signal, and consequently the $VarCoef$ feature:

$$VarCoef = \frac{\sqrt{\frac{1}{\|F\|} \cdot [\sum_{f \in F} [MS_{fc}(f) - \overline{MS_{fc}}]^2]}}{\overline{MS_{fc}}} \tag{3}$$

This feature assesses the relative variation of the magnitude along different frequency components. This may be of interest to help discriminate different species which may posses different degrees of magnitude variation along any of the frequency bands.

The final feature in this subset intends to highlight the average difference between two consecutive frequency component maximum magnitudes. This is achieved by the following expression:

$$Mvar = \frac{\sum_{f=0}^{\|F\|-1} |\max_t(m(f,t)) - \max_t(m(f+1,t))|}{\|F\| - 1} \tag{4}$$

where $m(f,t)$ corresponds to the magnitude of frequency component $f$ on time slice $t$, with $f$ being a higher frequency component value than $f+1$. $Mvar$ provides an insight on how smooth the progression of the vocalization magnitude is along the frequency axis. This is the case as by performing the difference between the maximums of contiguous frequency component we might identify the beginning or end of a vocalization contour, as they would present substantially different maximum magnitude values. By averaging these differences, a vocalization with a *steady* pattern (top left fig. 1) would score a lower $Mvar$ than a one with a more *erratic* behaviour (top right fig. 1), which can help to discriminate vocalizations of different species.

### 2.2. Contour analysis features

As discussed above, the analysis of the time-frequency representation of dolphin vocalizations with a frequency based approach alone may not be sufficient to properly distinguish the species. Therefore, we developed two additional features ($Fs_2$) that focus on describing the vocalization's spectral contour: ($a$) the average slope difference ($AvgSlopeDif$), which portrays the signals frequency progression over time; and ($b$) the number of inflexion points ($InflexNum$) that occur in the frequency contours.

The first step on the computation of the contour analysis features, is to detect the vocal contours in the magnitude spectrograms. For this, the feature computation algorithm starts by applying a peak tracking technique to the magnitude spectrogram based on the MQ modeling and PARSHL techniques [16, 17, 18]. This process starts by looking for the dominant intensity peaks in each time frame. Due to the characteristics of real-world signals, we cannot simply consider the local maxima in the frame, as too many maxima would be found. Thus, after finding the local maxima, these are filtered out by a peak prominence criterion that compares the height of the peak with the height of its immediate neighbors, which here are the 5 consecutive frequency bins around the peak (fig. 2).

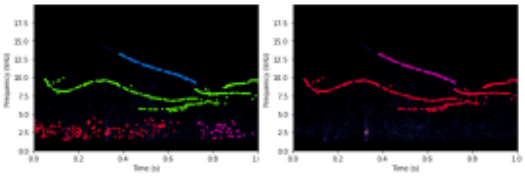


**Fig. 3**. Frequency contour clusters of a vocalization of *Stenella coeruleoalba* after applying peak tracking and DBSCAN (left) and sucessive density based cluster filtering approach to remove low density clusters (right).

The detection of these local magnitude peaks in successive time frames expresses an unpolished vocalization pattern to which we then apply a clustering algorithm to remove outliers and to obtain clusters of different sections of the vocalization frequency contours. The Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm [19] (with *eps*=11 and *min_samples*=12) is used for this purpose. However, sometimes this approach may still generate some low density clusters which may correspond to background noise and thus, need to be filtered out. To do so, we determine each of the clusters' density by estimating the average number of neighbors within a given radius to each point in the cluster, and then filtering out the clusters which have a bellow average cluster density. Fig. 3 shows an example of the outcome of this process.

These clusters will be the foundation for the estimation of the two contour analysis features. As both these metrics rely on intermediate time instance calculations, the spectrogram is divided into $n$ time segments (which we set to 10 in our tests) and we compute the slope of each cluster in the segments. For each cluster $c$ in time segment $[t_i, t_i + \Delta t^*]$, the first active point of the cluster ($P_{t_i}$) and the last active point ($P_{t_i+\Delta t^*}$) are used to calculate the slope $S_{c,t_i}$ of $c$ in that segment:

$$S_{c,t_i} = \frac{F_{P_{t_i+\Delta t^*}} - F_{P_{t_i}}}{\Delta t^*} \tag{5}$$

where $\Delta t^*$ is an approximate value of time interval $\Delta t = t_{i+1} - t_i$, since the first and last active points in the cluster may not coincide with those precise time instances. $F_{P_{t_i}}$ is the average frequency value of points within a smaller time window ($\pm\frac{(t_{i+1}-t_i)}{n}$) surrounding the closest point $P$ to time instance $t$, ($t_i$ or $t_{i+1}$). This approach is used as a way to more closely capture the real slope of the cluster, as the true frequency value of the closest point to time instance $t$ could be itself an outlier and misrepresent the true cluster's slope at that time.

With this, it is possible to estimate the average cluster slope difference for each cluster with the following expression:

$$ClustSlopeDif(c) = \frac{\sum_{i=1}^{\|T\|-1} S_{c,t_{i+1}} - S_{c,t_i}}{\|T\| - 1} \tag{6}$$

where $ClustSlopeDif(c)$ is given by the average of the difference between the slopes $S$ of adjacent time intervals $[t_i, t_{i+1}]$ in cluster $c$. As a cluster only accounts for a segment of a given vocalization, the final aimed feature is expressed by the average value for each cluster in the spectrum:

$$AvgSlopeDif = \frac{\sum_{c \in C} ClustSlopeDif(c)}{\|C\|} \tag{7}$$

The metric regarding the number of inflexion points, i.e. shifts between upsweeps and downsweeps, in the vocalization is given by:

$$InflexNum = \sum_{c \in C} \sum_{i=1}^{\|T\|-1} 0^{(S_{c,t_i} \times S_{c,t_{i+1}} + |S_{c,t_i} \times S_{c,t_{i+1}}|)} \tag{8}$$

By taking advantage of the property that expresses that $0^0 = 1$ and $0^n = 0\ \forall n \in \mathrm{R}_{>0}$, we multiply slopes $S_{c,t}$ in adjacent time intervals in such a way that if they carry opposite signs the power will be 0 and an inflexion in the pattern would be detected. This operation is applied to every cluster in a signal, resulting in the final number of inflexions in the vocalization.

## 3. CLASSIFICATION

### 3.1. Data

In order to train and validate the proposed feature set, we assembled a dataset containing vocalizations of four distinct dolphin species: Short-beaked common dolphin (*Delphinus delphis*), Atlantic spotted dolphin (*Stenella frontalis*), Striped dolphin (*Stenella coeruleoalba*) and Bottlenose dolphin (*Tursiops truncatus*). This dataset comprises 910 one-second recording samples, downsampled to 40 kHz. This particular sampling rate was chosen due to being the minimum sampling rate of the recordings used in our study. This dataset encompasses recordings obtained by the Madeira Whale Museum (MWM) and others from the Watkins Marine Mammal Sound Database (WMMSD) [20].

The small dolphins acoustic recordings from Madeira were collected by the MWM scientific team during dedicated boat surveys. Whenever there were good weather conditions and a group of dolphins of the species of interest was sighted, the boat stopped in the vicinity of the group and a compact self-contained underwater sound recorder (SoundTrap 300 series, model HF, recording the 20 Hz to 150 kHz bandwidth) was deployed in continuous recording mode. The device recorded at 10 m depth and was kept floating by a system of buoys of different sizes connected by an elastic rope to the recorder. This layout was used to minimize the waves driven vertical movement of the device which generates noise as the recorder moves through the water. The boat waited 100 m away with the engine off while the device was recording.

To diversify our recording samples, we ensure at least two distinct recording locations for each of the species present in the dataset, which is done by inlcuding data from WMMSD. Table 1 shows the source distribution of the recordings among the four species.

In order to reduce low frequency noise, which is predominant in some recordings, a 4th order Butterworth highpass filter was applied with a cutoff frequency of 1000 Hz. Due to the wide frequency range the vocalizations of these species can reach, a higher cutoff value was not used, as it could have withhold relevant information.

| Species | WMMSD | MWM | Total |
|---|---|---|---|
| *Delphinus delphis (Dd)* | 238 | 164 | 402 |
| *Stenella frontalis (Sf)* | 165 | 31 | 196 |
| *Stenella coeruleoalba (Sc)* | 134 | 0 | 134 |
| *Tursiops truncatus (Tt)* | 42 | 136 | 178 |

**Table 1**. Distribution of recordings by its original source.

| | | $Dd$ (%) | $Sf$ (%) | $Sc$ (%) | $Tt$ (%) | $M_{acc}$ (%) |
|---|---|---|---|---|---|---|
| $STFT$ | $a$ | 81.54 | 94.62 | 87.69 | 88.46 | **88.08** |
| | $b$ | 86.15 | 90.00 | 84.62 | 80.00 | **85.19** |
| | $c$ | 79.23 | 93.08 | 80.00 | 83.08 | **83.85** |
| | $d$ | 82.31 | 98.46 | 86.15 | 88.46 | **88.85** |
| $MFCC_{20}$ | $a$ | 85.38 | 96.15 | 90.00 | 88.46 | **90.00** |
| | $b$ | 71.54 | 93.85 | 89.23 | 83.08 | **84.42** |
| | $c$ | 70.77 | 96.92 | 93.08 | 80.00 | **85.19** |
| | $d$ | 79.23 | 96.15 | 93.08 | 87.69 | **89.04** |
| $MFCC_{40}$ | $a$ | 91.54 | 97.69 | 90.77 | 91.54 | **92.88** |
| | $b$ | 86.15 | 82.31 | 93.85 | 81.54 | **85.96** |
| | $c$ | 78.46 | 92.31 | 93.08 | 86.15 | **87.70** |
| | $d$ | 83.08 | 95.38 | 96.92 | 90.00 | **91.35** |

**Table 2**. Accuracy results for KNN. The first column shows the data representation and the second column shows the set of features.

| | | $Dd$ (%) | $Sf$ (%) | $Sc$ (%) | $Tt$ (%) | $M_{acc}$ (%) |
|---|---|---|---|---|---|---|
| $STFT$ | $a$ | 83.03 | 96.92 | 83.08 | 90.77 | **88.46** |
| | $b$ | 81.54 | 96.15 | 84.62 | 79.23 | **85.38** |
| | $c$ | 73.08 | 80.00 | 92.31 | 90.77 | **84.04** |
| | $d$ | 80.77 | 95.38 | 81.54 | 93.08 | **87.69** |
| $MFCC_{20}$ | $a$ | 85.38 | 93.08 | 94.62 | 91.54 | **91.15** |
| | $b$ | 80.77 | 96.15 | 89.23 | 88.46 | **88.65** |
| | $c$ | 84.62 | 90.77 | 86.92 | 93.08 | **88.85** |
| | $d$ | 85.38 | 96.92 | 96.15 | 86.92 | **91.35** |
| $MFCC_{40}$ | $a$ | 86.92 | 93.08 | 95.38 | 97.69 | **93.27** |
| | $b$ | 86.92 | 89.23 | 93.85 | 86.92 | **89.23** |
| | $c$ | 85.38 | 90.00 | 93.08 | 93.85 | **90.58** |
| | $d$ | 88.46 | 95.38 | 96.92 | 94.62 | **93.85** |

**Table 3**. Accuracy results for SVM with RBF kernel. The first two columns show the data representation and the set of features.

### 3.2. Training phase

Before proceeding to the training of the classification models, as the features span different values of magnitude, they were normalized to become equally weighted. Even though we presented five features, one of them, $MS_{fc}(f)$, can be reflected in up to 257 *sub features* (number of frequency bins in the spectrogram). This led us to use independent component analysis (ICA) for dimensionality reduction (with the FastICA algorithm). This reduced the whole feature set into 8 independent components while minimizing the amount of mutual information among them [21].

The dataset was split into three distinct sets for the purposed of training (70%), validation (20%) and testing (10%). To estimate the hyper-parameters of the models ($C$ and $gamma$ for SVM with the radial basis function (RBF) kernel, and $k$ for KNN), a grid search approach was used with 5-fold cross validation over the validation set. Following the estimation of the optimal hyper-parameters, the training and validation sets were used for training the models. Finally, the model's accuracy is estimated from the test set.

## 4. RESULTS AND DISCUSSION

In order to have representative results, we run several tests, each with a given combination of features, a specific spectral representation and classification model. Each test was run 10 times. With the cumulative predictions of those runs, we estimated the general model accuracy ($M_{acc}$) and each species accuracy for a given test parameterization.

As mentioned in Sections 1 and 2 we used distinct time-frequency representation approaches: ($i$) the STFT, and MFCCs with ($ii$) 20 coefficients, and ($iii$) 40 coefficients. For each of these representations we tested four different combinations of features: $a$ – the frequency analysis feature subset ($Fs_1$); $b$ – both $Fs_1$ and the vocalization contour analysis feature subset ($Fs_2$); $c$ – $Fs_1$ and the $AvgSlopeDif$ feature from $Fs_2$; and $d$ – $Fs_1$ and the $InflexNum$ feature from $Fs_2$. The results of these tests for both KNN and SVM can be seen in Tables 2 and 3, respectively.

Using the MFCCs representation produces more accurate models than the STFT while reaching a maximum overall model accuracy of 93.85% (table 3, test $MFCC_{40} - d$). Also, doubling the number of MFCCs yielded better results with both classifiers for any of the shown combination of features. This may be due to the increased number of coefficients over higher frequencies which do not get enough detailed representation with fewer coefficients.

The joint test of both $Fs_1$ and $Fs_2$ ($b$) provided worse results than $Fs_1$ alone ($a$). However, as some species showed individual accuracy improvements with both feature subsets (e.g. the classification accuracy of specie $Sf$ improved by 3% with SVM and 20 MFCCs, see tests $MFCC_{20} - a$ and $b$ in table 3), this led us to test each individual features in $Fs_2$ with the whole feature set $Fs_1$. These tests showed that in general, feature $InflexNum$ combined with $Fs_1$ (test $d$) outperforms both the results with all features (test $b$) as well as the results with feature $AvgSlopeDif$ combined with $Fs_1$ (test $c$). Nonetheless, test $c$ showed there may be higher accuracy improvements for some species with feature $AvgSlopeDif$ (e.g. table 3 shows an accuray of 92.31% for specie $Sc$).

## 5. CONCLUSION

This paper proposes a new set of acoustic features capable of distinguishing small dolphin species by their vocalizations. These features were developed with two objectives in mind: (1) to identify a signal's predominant frequency components and (2) to identify higher-level details in the vocalization patterns. The obtained results suggest that these approaches complement each other as they contribute to an improvement in the accuracy of the models. The best results were obtained using SVM with 40 MFCCs and all features except $AvgSlopeDif$. An accuracy of 93.85% was achieved ($Delphinus\,delphis$: 88.46%; $Stenella\,frontalis$: 95.38%; $Stenella\,coeruleoalba$: 96.92%, $Tursiops\,truncatus$: 94.62%), which, to the best of our knowledge, surpassed the results of previous studies on the task of dolphin classification. The application of these results can be highly relevant in the context of small dolphin PAM taking place in the archipelago of Madeira and worldwide. As future work we plan to extend our study to other cetacean species.

## 6. ACKNOWLEDGEMENTS

The authors wish to thank Ruth Esteban and Pauline Gauffier for their assistance in the field work to collecting the acoustic recordings of small dolphins in Madeira coastal waters, in the context of Project META - Marine Mammal and Ecosystem: anthropogenic Threat Assessment (Fundo Azul Edital nº6/2017), carried out by the Madeira Whale Museum. This work was partially supported by the Portuguese Foundation for Science and Technology under project NOVA-LINCS (PEest/UID/CEC/04516/2019).

## 7. REFERENCES


[1] Mohammed Bahoura and Yvan Simard, "Blue whale calls classification using short-time fourier and wavelet packet transforms and artificial neural network," *Digital Signal Processing*, vol. 20, no. 4, pp. 1256–1263, July 2010.

[2] Ramón Miralles, Guillermo Lara, Alicia Carrión, and Jose Antonio Esteban, "Automatic detection and classification of beluga whale vocalizations," *Advances in Applied Acoustics*, vol. 2, no. 2, pp. 61–70, 2013.

[3] Pablo Peso Parada and Antonio Cardenal-López, "Using gaussian mixture models to detect and classify dolphin whistles and pulses," *The Journal of the Acoustical Society of America*, vol. 135, no. 6, pp. 3371–3380, June 2014.

[4] Simone Baumann-Pickering, Sean M. Wiggins, John A. Hildebrand, Marie A. Roch, and Hans-Ulrich Schnitzler, "Discriminating features of echolocation clicks of melon-headed whales (peponocephala electra), bottlenose dolphins (tursiops truncatus), and gray's spinner dolphins (stenella longirostris longirostris)," *The Journal of the Acoustical Society of America*, vol. 128, no. 4, pp. 2212–2224, Oct. 2010.

[5] Douglas Gillespie, Marjolaine Caillat, Jonathan Gordon, and Paul White, "Automatic detection and classification of odontocete whistles," *The Journal of the Acoustical Society of America*, vol. 134, no. 3, pp. 2427–2437, Sept. 2013.

[6] Marta Azzolin, Alexandre Gannier, Marc O. Lammers, Julie N. Oswald, Elena Papale, Giuseppa Buscaino, Gaspare Buffa, Salvatore Mazzola, and Cristina Giacoma, "Combining whistle acoustic parameters to discriminate mediterranean odontocetes during passive acoustic monitoring," *The Journal of the Acoustical Society of America*, vol. 135, no. 1, pp. 502–512, Jan. 2014.

[7] Tzu-Hao Lin and Lien-Siang Chou, "Automatic classification of delphinids based on the representative frequencies of whistles," *The Journal of the Acoustical Society of America*, vol. 138, no. 2, pp. 1003–1011, Aug. 2015.

[8] Maheen Nadir, Syed Muhammad Adnan, Sumair Aziz, and Muhammad Umar Khan, "Marine mammals classification using acoustic binary patterns," 2020.

[9] Florence Erbs, Simon H. Elwen, and Tess Gridley, "Automatic classification of whistles from coastal dolphins of the southern african subregion," *The Journal of the Acoustical Society of America*, vol. 141, no. 4, pp. 2489–2500, Apr. 2017.

[10] Thiago Orion Simões Amorim, Franciele Rezende De Castro, Juliana Rodrigues Moron, Bruna Ribeiro Duque, Juliana Couto Di Tullio, Eduardo Resende Secchi, and Artur Andriolo, "Integrative bioacoustics discrimination of eight delphinid species in the western south atlantic ocean," *Plos One*, vol. 14, no. 6, 2019.

[11] Bernardo B. Gatto, Eulanda M. dos Santos, Juan G. Colonna, Naoya Sogi, Lincon S. Souza, and Kazuhiro Fukui, "Discriminative Singular Spectrum Analysis for Bioacoustic Classification," in *Proc. Interspeech 2020*, 2020, pp. 2887–2891.

[12] Juan J. Noda, David Sánchez-Rodríguez, and Carlos M. Travieso-González, "A methodology based on bioacoustic information for automatic identification of reptiles and anurans," in *Reptiles and Amphibians*. InTech, July 2018.

[13] Steven Ness, *The Orchive: a system for semi-automatic annotation and analysis of a large collection of bioacoustic recordings*, Ph.D. thesis, University of Victoria, 3800 Finnerty Road, Victoria, British Columbia, Canada, V8P 5C2, 2013.

[14] Yin Xian, Andrew Thompson, Qiang Qiu, Loren Nolte, Douglas Nowacek, Jianfeng Lu, and Robert Calderbank, "Classification of whale vocalizations using the weyl transform," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Apr. 2015, IEEE.

[15] Whitlow W. L. Au (auth.), *The Sonar of Dolphins*, chapter 7 - Characteristics of Dolphin Sonar Signals, p. 134, Springer-Verlag New York, 1 edition, 1993.

[16] Robert John McAulay and Thomas Quatieri, "Sinusoidal coding," in *Speech Coding and Synthesis*, W.B. Kleijn and K.K. Paliwal, Eds., chapter 4, pp. 121–173. Elsevier Science B.V., 1995.

[17] Robert John McAulay and Thomas Quatieri, "Speech analysis/synthesis based on a sinusoidal representation," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. ASSP-34, no. 4, pp. 744–754, August 1986.

[18] Julius Smith and Xavier Serra, "PARSHL: an analysis/synthesis program for non-harmonic sounds based on a sinusoidal representation," in *Proceedings of the International Computer Music Conference*, 1987, pp. 290–297.

[19] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. 1996, KDD'96, p. 226–231, AAAI Press.

[20] Laela Sayigh, Mary Ann Daher, Julie Allen, Helen Gordon, Katherine Joyce, Claire Stuhlmann, and Peter Tyack, ," in *The Watkins Marine Mammal Sound Database: An online, freely accessible resource*. 2016, Acoustical Society of America.

[21] Aapo Hyvärinen and Erkki Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, June 2000.

# II

# Annex: CNN model schematics

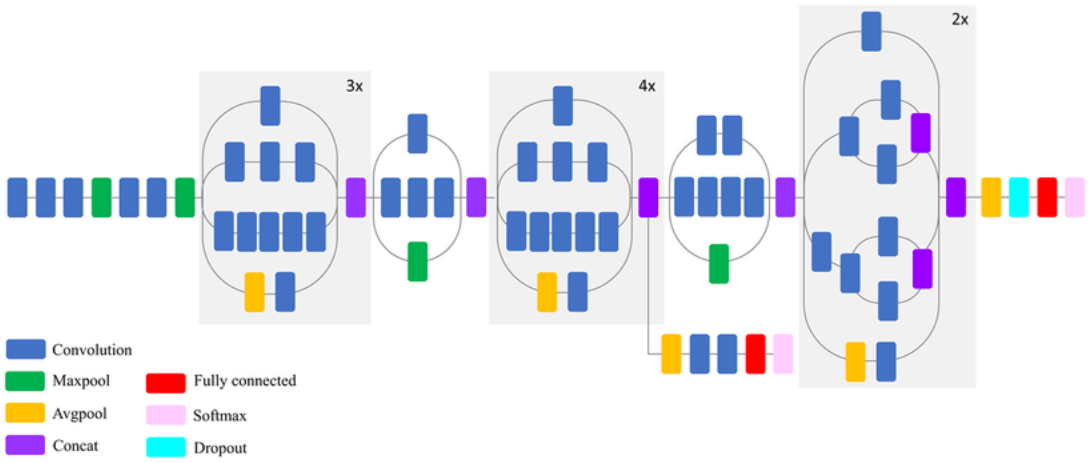



Figure II.1: Compressed view of the schematic diagram of the InceptionV3 model [76]

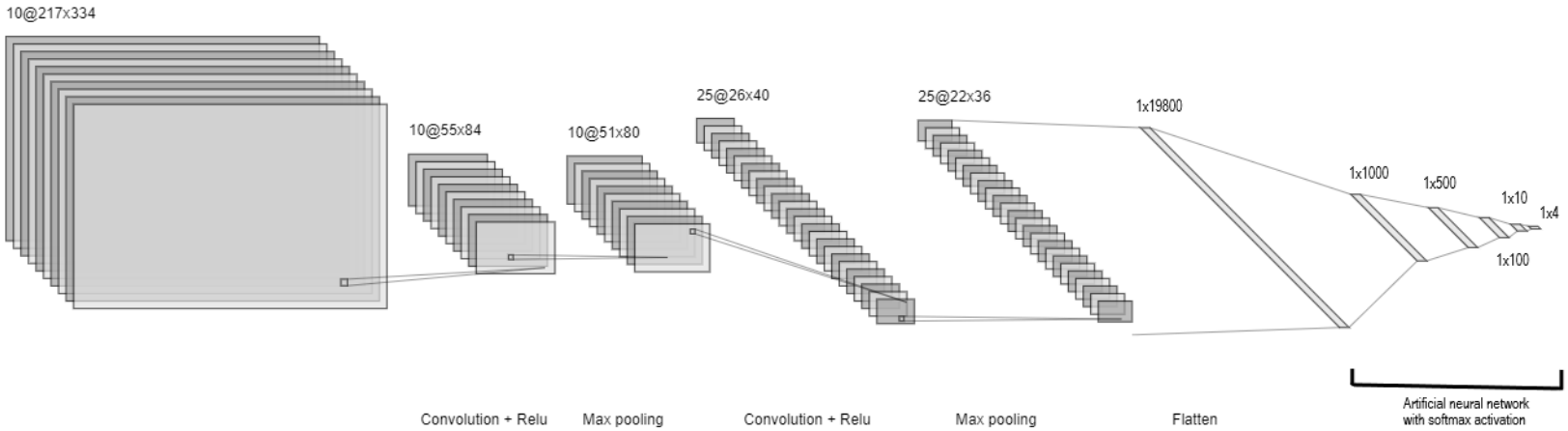



Figure II.2: Architectural diagram of the CNN costum model 1 (CM1).

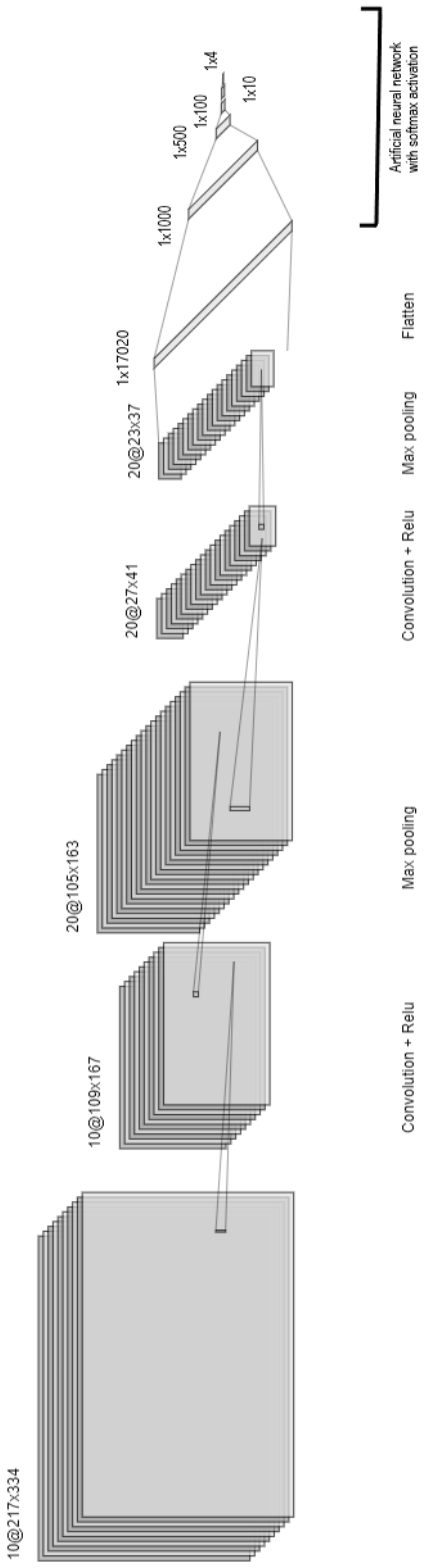



Figure II.3: Architectural diagram of the CNN costum model 2 (CM2).