Right Whales Up-Call Detection using Deep Classifiers over Underwater Noisy Recordings

Master in Artificial Intelligence

2019/10/21

Author: Jorge Rodriguez Molinuevo

Director: Enrique Romero Merino

Department of Computer Science (formerly Llenguatges i Sistemes Informàtics)

Co-Director: Ludwig Roland Houegnigan

Abstract

This project evaluates the potential of convolutional Neural Networks in classifying Right Whales' Up-Calls from short audio clips of environmental sounds. Two deep models with different architectures are presented evaluating them over different preprocesses on the same dataset. One architecture is based on Alexnet with three Convolutional layers and two dense layers. The other one with two Convolutional Layers and one dense layer. Different metrics are presented to evaluate these models.

Acknowledgments

First of all I would like to express my gratitude to my Director Enrique Romero and Co-Director Ludwig Houegnigan for their guidance and support in completing my project. Without them I would have never finished it.

I would also like to extend my gratitude to my roommates that let me occupy the living room for the last two months while I was writing the memory, giving me the silence I needed for concentration. Especially one of them that always reminded me to stay focused and keep on the good work. I would also like to thank other friends that were always there when I needed to air out.

Finally I would like to thank my family that were always there asking me about the project and cheering me up.

1 - Introduction

1.1 - Motivation

In a world where Climate Change and Pollution have become pressing issues, keeping a census of sea animals is important to keep track of the damage done to the sea environments and be able to recover or at least minimize our harmful effect.

Sea observation has always been tricky. Oceans have always been mysterious and strange. Counting the number of members in an animal species is always a hard problem on earth and lots of observations are needed to produce as close as possible estimates.When animals live in an unknown environment the problem gets harder, taking accurate observations of sea animals is difficult and estimates are often far from the truth. Many ocean species are discovered every time an expedition is sent to the San Andreas Fault, and many are yet to be discovered.

Estimating the number of fish in a lake is a classical problem in Statistics, there are plenty of studies and methodologies. The main approach is to catch a portion of the species, mark it and then release it. This is a very slow process, and requires the use of several researchers to catch by hand each animal and tag it, measure it and then release it. Nevertheless, this approach can be very invasive and traumatic for the animals and also very difficult for certain species that live in the bottom or are hard to catch. In the ocean the process is slower and results are less accurate. There are species that can live in depths of thousands of meters. Some mammals in the ocean, such as fin whales, blue whales and sperm whales can dive to very deep waters and hold their breath for more than an hour. Catching and releasing this species is not an easy task.

Since tracking marine species it is a hard task, many different approaches have been presented: from satellite images to underwater microphones known as hydrophones. These non-invasive techniques can not only reduce the effect of human presence in marine life, but also produce better and more accurate results. The problem is that to achieve this, thousands of Terabytes of data are produced and need to be processed every day in order to have a glance of the ocean fauna.

Marine mammals communicate with each other using different kinds of frequencies and sounds which are broadly called vocalizations. Blue whale vocalizations can travel up to 800 kilometers [1]. Many other marine mammals use them not only for communication but also to echolocate their prey or obstacles in the darkness of the ocean. Marine mammals are really intelligent animals, that due to human influence have changed their behaviour. In a study done by the Oregon State University [2] scientists have discovered that due to the presence of humans in the oceans some whales have been changing their vocalizations: “Other baleen whales in the North Pacific have been

recorded in recent years generating vocalizations that are missing the "overtone" portions of their calls” in [2].

In the sea, sound travels at a speed of 1500 m/s, whereas in air the speed is only 340m/s. Sounds and noises travel 5 times faster underwater than in the atmosphere. Sound is a pressure wave, but this wave behaves slightly differently through air as compared to water. Water is denser than air, so it takes more energy to generate a wave, but once a wave has started, it will travel faster than it would do in air. This is due to the higher density of water which reduces the loss of energy from the transmission of vibrations between particles. This does not only affect speed, but also the distance traveled, the better the energy is transmitted, the more particles can reach. This relation between speed and distance is not linear, distance traveled can be hundreds of times greater than on the air. NOAA, the National Ocean and Atmospheric Administration has an interesting article [30] about how water's pressure and temperature can affect sounds in the water. The same government service has another interesting article [31] about underwater sound and whales calls. Summarizing both articles, sound underwater can travel for thousands of miles through what are called “sound channels” that are formed due to pressure and temperature changes in water. Through these channels sound can travel for miles without losing considerable energy.

Noise pollution is a problem that multiplies underwater, sound speed increase does not only affect whale vocalization,but all the sounds in the sea, which means that human created sounds have a bigger effect underwater. With the huge commercial routes created during the last 100 years and the new motors and engines, human noise in the ocean has increased, producing changes in the habits of underwater wildlife. These changes in the whales vocalization have complicated the task of searching for marine fauna, thus producing the creation of new algorithms and systems.

It is worth mentioning that there are systems like sonar that have an even bigger impact on sea animals, as shown in the following article, “Does Military Sonar Kill Marine Wildlife?” [6].

Noise is not the only threat for marine mammals. Whale hunting has been a hot issue during these last years in countries like Japan, some species of whales have gone close to extinction due to hunting and governments have put retristiction to this kind of practice. Recovering of these species is key for ecosystems and very difficult, due to their size and reproduction cycles. Having a good estimation and control of whale populations can have a huge impact on the measures taken for these species preservation.

Whales are the biggest animals in the sea and have a huge impact on phytoplankton populations. These microscopic creatures contribute at least 50% to the global oxygen production by taking huge amounts of CO_2 from the atmosphere [5], helping whale conservation can make phytoplankton populations recover and reducing global warming. At the same time, the carbon taken from the atmosphere is converted to organic matter which other species feed and later is stored on their bodies. When these

animals die, their carcasses sink to the seafloor, bringing a lifetime of trapped carbon with them. This is called Deadfall Carbon. On the deep seafloor, it can be eventually buried in sediments and potentially locked away from the atmosphere for millions of years, therefore reducing greenhouse gas emissions [22].

Whales are endangered animals. Their size and characteristics makes them susceptible to small changes in their ecosystems ecosystems. Many of these animals are found stranded in the shores every year and several investigations have been opened by the NOAA (National Oceanic and Atmospheric Administration of the USA) to investigate this event [19][20].

Sound can become a tool that can be used to monitor and control marine mammals populations. These tools can provide a way of avoiding these tragic events and save the whales. It also makes population studies less invasive and more precise.

1.2 - Task Definition

Each specie and class of animal presents a different problem, depending on their habitat and lifestyle. The main focus of this project will be on how to detect whales using Deep Learning and sound, specifically the characteristic Right Whales' Up-Call.

Right Whales correspond to three species of large baleen whales: the North Atlantic right whale, the North Pacific right whale and the Southern right whale. Right whales have always been a preferred target for whalers because of their docile nature, their slow surface-skimming feeding behaviors, their tendencies to stay close to the coast, and their high blubber content, which makes them float when they are killed, and which produced high yields of whale oil.

North Atlantic and North Pacific Right Whales are among the most endangered whales in the world[21], and both species are protected in the United States by the Endangered Species Act. The western populations of both are currently endangered, with their total populations numbering in the hundreds. The eastern North Pacific population, on the other hand, with fewer than 50 individuals remaining, is critically endangered [21] – further still, the eastern North Atlantic population, may already be functionally extinct. Although the whales no longer face pressure from commercial whaling, mankind remains by far the greatest threat to these species: the two leading causes of death are being struck by ships and entanglement in fishing gear. Ingestion of plastic marine debris also presents a growing threat.

This makes this particular species of whales an interesting target since the populations are moving and recovering efforts are at a critical state. Several scientists from different fields are putting their efforts on the protection, recovering and control of these species.

If it is possible to correctly detect each whale characteristic up-call, then it should be possible to appropriately estimate the number of individuals that are left. First it is needed to find a way to detect the characteristic sound of these whales from the rest of the whale species and the underwater noise. There has been an effort from the scientific community not only to preserve but also to control and detect whale populations.

1.3 - Previous Work

There are several organisms working on this topic: The University of Rhode Island, the United States' National Oceanic and Atmospheric Organization and Marine Acoustic, Inc, or "Laboratori d'Aplicacions Bioacústiques" of UPC. They are making an effort on oceanic conservation and have some interesting articles on *Sound in the Sea*, all this information can be found on their web page: https://dosits.org/ [10]. These Institutions provide resources that can be really interesting to understand the topic of underwater sound propagation and its importance in marine conservation.

Apart from these United States' Institutions and Universities many other researchers are working on this issue. There are several approaches to this problem, the classical one is to apply handcrafted features or templates to the sound's spectrograms and later use any machine learning classifier[4][7][32]. There are some other algorithms that try to match and trace the sea mammals' vocalizations, dolphins in particular which a large range of vocalizations and often happens at the same time [26].

Other more modern approaches are using Deep Neural Networks, such as AlexNet for the task of feature extraction [42] or directly applying Deep Learning to solve the problem: [27][33]. There are similar problems solved with ANN such as discriminating different species calls from each other [25].

Other examples of Deep Learning for this task is the use of satellite images to detect whale population surfacing [3][12] or even recognise each of the individuals using face recognition techniques on images of whales captured from ship-based surveys or drones[13].

Since the main focus of the project will be the detection Right Whale Up-Call the following 2 articles will be used as reference:

> *27 - Smirnov, Evgeny. "North Atlantic right whale call detection with convolutional neural networks." Proc. Int. Conf. on Machine Learning, Atlanta, USA. 2013.*

33 - Xu, Kele, et al. “North Atlantic Right Whale Call Detection with Very Deep Convolutional Neural Networks.” The Journal of the Acoustical Society of America

The first article presented a similar solution close to the one presented in this project with a similar architecture on the same dataset. Their results will be used as a reference. In this article they show that Convolutional Neural Networks can be used for this problem. The second article presents a better solution with a much larger Neural Network.

Both articles will be used as reference for the models created in this project, each one being similar to each of the models presented in these articles.

2 - Data

2.1 - Data Origin

There have been two main sources of data, both of them came from Kaggle [34] (https://www.kaggle.com/) a web page where difficult Data Science challenges are thrown so as the community try to solve them. Problems that big or small companies cannot solve by their own and leave them for people all around the world to try. The bounty is usually an amount of money proportional to the difficulty of the problem and a job opportunity, similar to the Netflix competition, where a Million dollar was at stake. Big companies like Google, Santander Bank or NFL upload all kinds of problems involving data science.

One of these notorious competitions was The Marinexplore and Cornell University Whale Detection Challenge [16], where the first place got 8000$. The challenge was to use the 30000 sounds in the training database to predict the secret 54000 test samples. Therefore there are 30000 label examples and 54000 unlabeled examples. Since the competition has finished, thel focus will be on the labeled samples, although many participants used the unlabeled data to pretrain their models in an unsupervised fashion to gain some advantage. Later I will talk about the competition and what models where used and with what success. This data set will be called Kaggle's original dataset from now on.

Following this competition, another one was held, The ICML 2013 Whale Challenge - Right Whale Redux [11]. This complemented the previous competition with another 47841 labeled samples for training purpose and 25468 test samples which are not labeled and serves as a measurement of a model's effectiveness. Since the access to the labels of these test samples is restricted, these samples will be discarded. The 47841 samples are enough to create a successful dataset. The competitions are connected and both datasets can be used together or separately. For reasons later explained, both ways are being presented. Taken the name from the competition this dataset will be called Redux from now on.

2.2 - Data Information

Kaggle's original dataset consists of 30000 2-seconds sound clips with a sample rate of 2 kHz (2000 points per second). The samples contain the North Atlantic Right Whale

calls, non-biological noise, underwater background noise or other whale calls. The objective is to find an algorithm able to detect the right whale calls.

Once the data has been obtained the next thing that is needed is to preprocess it. Working directly with sound waves is hard, in this case there is the following:

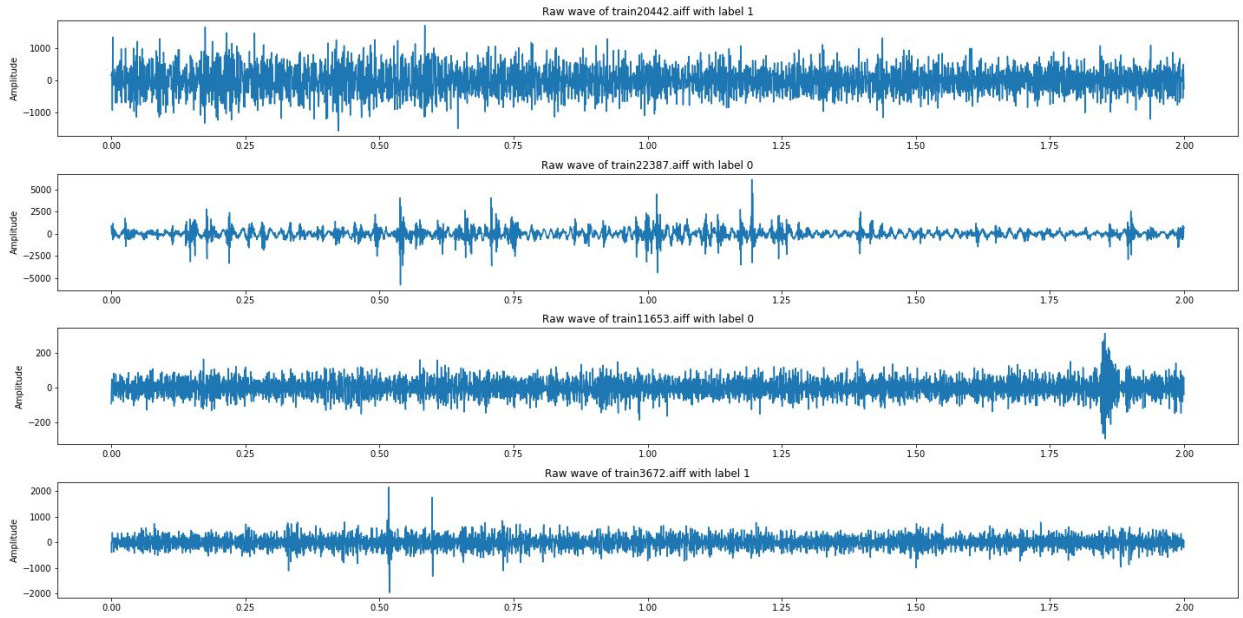
*2000 samples per second * 2 seconds = 4000 samples per clip*

Working with 4000 dimensional vectors is complicated and there is the sequential time relation, so the first approach could be a Recurrent Neural Network, however there are not many examples of it. Even Convolutional Neural Networks have been used, but never directly on the sampled values, there is always some kind of preprocessing.

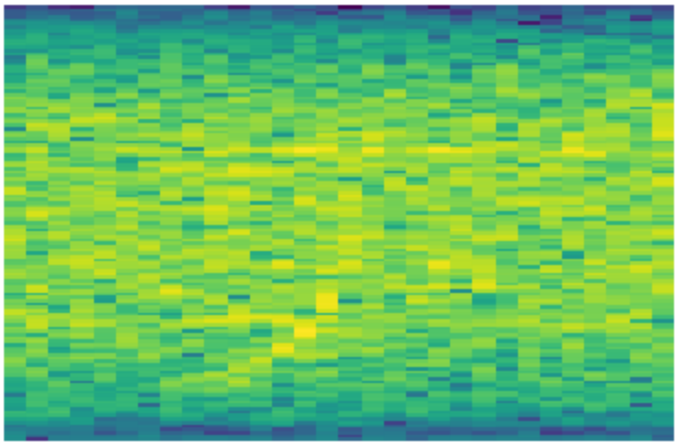
The first two teams in the competition, “Sluicebox” and “alfnie” were able to obtain their scores using SVM on hand crafted feature created to target the whales up-call vocalization’s characteristic curve in the spectrogram. The third one, which has already been mentioned before used CNN with data augmentation techniques and powerful computers which let him easily explore the hyperparameters and try several random states, also used a preprocess unknown since the code was never released, which means that there could be better techniques to create more useful spectrograms, normalization used in the process can also mean a lot. When working with Neural Networks normalization can help the model to be more stable and train faster.

2.3 - Data preprocessing, creating spectrograms

The wave doesn’t give any useful information, so the different frequencies that generate the wave needs to be extracted or a similar approach to get a series of features that can help to interpret the ups and downs in a more comprehensive way, for this Fourier transforms are usually used to obtain the Spectrogram representation, so the waves transform from this:

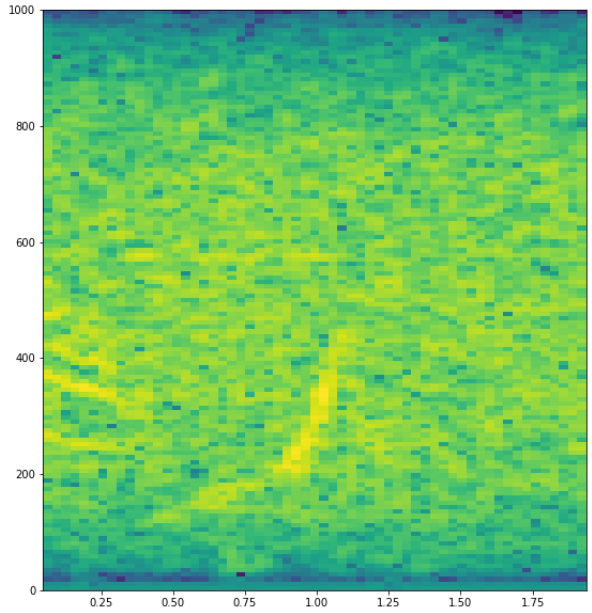


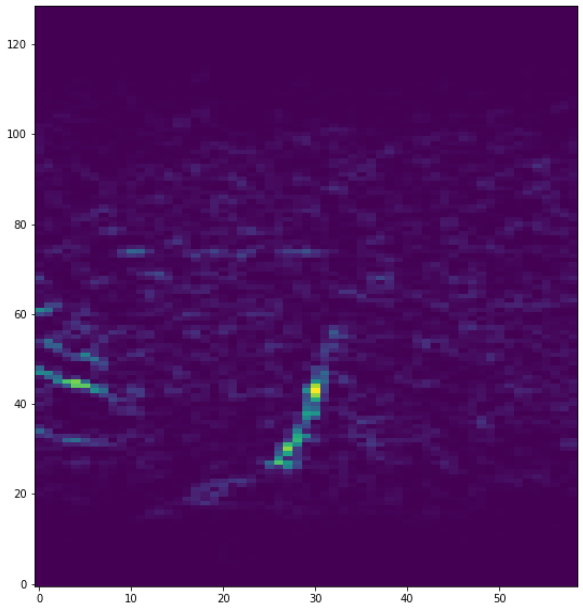
To this:



There are several ways of creating the spectrograms, many parameters can be tweaked to improve quality, such as, windows size, overlap, type of window used for sampling… The software used to create these representations also obtains different spectrograms for the same wave. The following images show 2 different examples using the same

software and parameters. Both images are from the same audio clip containing a Right Whale Up-Call, which is the characteristic curve that rises from the bottom until the half of the image.



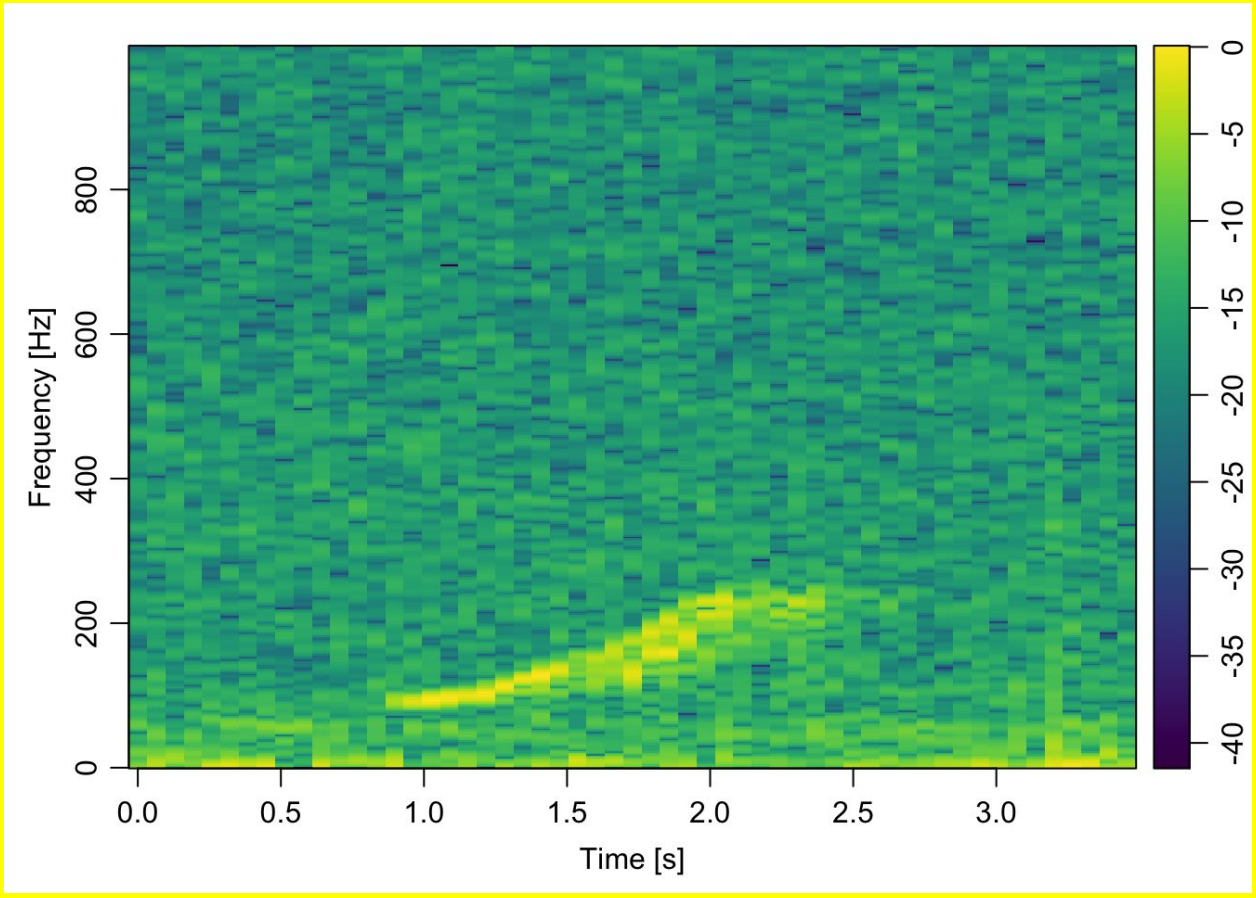


Spectrograms come in a variety of colors and definition. The theory is the same, what changes is the colour patterns chosen. The one on the left has been mapped onto a colour template using a logarithmic scale, blue for low values and yellow for higher ones, the one on the right was created using a linear scale. This preprocess was done by the visualization software, sometimes these libraries can even apply some kind of inner preprocess such as low filtering or noise cleaning that is not part of the original algorithm. In this case Python was used for the creation of the spectrograms. Although there are many libraries that can get these images from audio data, each one has a different internal process. For this project the library “matplotlib” was used, which includes the module “mlab” that imports several of Matlab functions to python and pyplot that allow the visualization of the results.

The details on how to get this representation are not important so only a brief description of the way this is calculated will be presented, since it is important to know what the algorithm yields.

A spectrogram is a visual representation of the Short-Time Fourier Transform, i.e. a 3 dimensional representation of sound, for every time step, and frequency, the algorithm calculates a value called amplitude. This representation then can be shown as an image, where the X-Axis represent time, Y-axis represent frequencies and the colour or the brightness represents the amplitude. Usually Spectrograms are created with a

colour map, for human interpretation, where yellow/red represent higher amplitude and blue/black represent lower amplitudes.



The previous images were extracted with Python's Pyplot library, and although they seem to be completely different, they are all equally valid. The only thing that changes is the colour map and some preprocesing used by the displaying software itself, not only the way it was calculated affects the representation, but also the visualization tool.

There are several parameters that can be tweaked to change the spectrograms results. The ones that make a bigger difference and are the ones that will be optimized for the generation of the Spectrograms are:

- Fs: Sampling frequency of the x time series. It is used to calculate the Fourier frequencies in cycles per time unit.
- Window: This is a window centered in each audio point, used in the calculation of the Fourier transform. The values of the windows represent the relation

between that point and the surrounding ones, for example, the further the points are, the less important they are.

- Nfft: The window length, the number of data points used in each block for the Fast Fourier Transform.
- Number of overlap: The values that are overlapping between windows/points, it has to be at maximum half the size of the windows.

Now that the sound waves have been converted into images, image recognition algorithms and preprocessing can be applied for classification.

To obtain the spectrograms the software used was Matplotlib a Python library that implements Matlab functionalities. The parameters mentioned before for creating the spectrograms have been fixed to: a NFFT of 256, number of overlap 128 and the default window function.

2.4 - Data statistical analysis

2.4.1 - Cornell's Kaggle Original Right Whale Competition

Kaggle's original dataset consists of the samples previously presented. The competition delivered 30000 training samples which are labeled and 54,503 test samples that are not labeled. These test labels are the ones used by the competition to evaluate the best model, in this case the metric chosen for this purpose was AUC of ROC (Area Under Receiver Operating Characteristic Curve). Since the access of these labels is restricted and the competition is no longer held there is no way of retrieving them and so they will be left apart from the project.

So there are 30000 samples of two seconds clip with a framerate of 2000 frames per second which correspond to 40000 values order by time. From these samples there are 22973 clips not containing a right whale vocalization and 7027 that contain one, which means there is only a 23.42% of samples with right whales, meaning that for every sample with a call there are three without one. This imbalance between classes could be a problem for the model.

2.4.2 - Kaggle Competition Redux

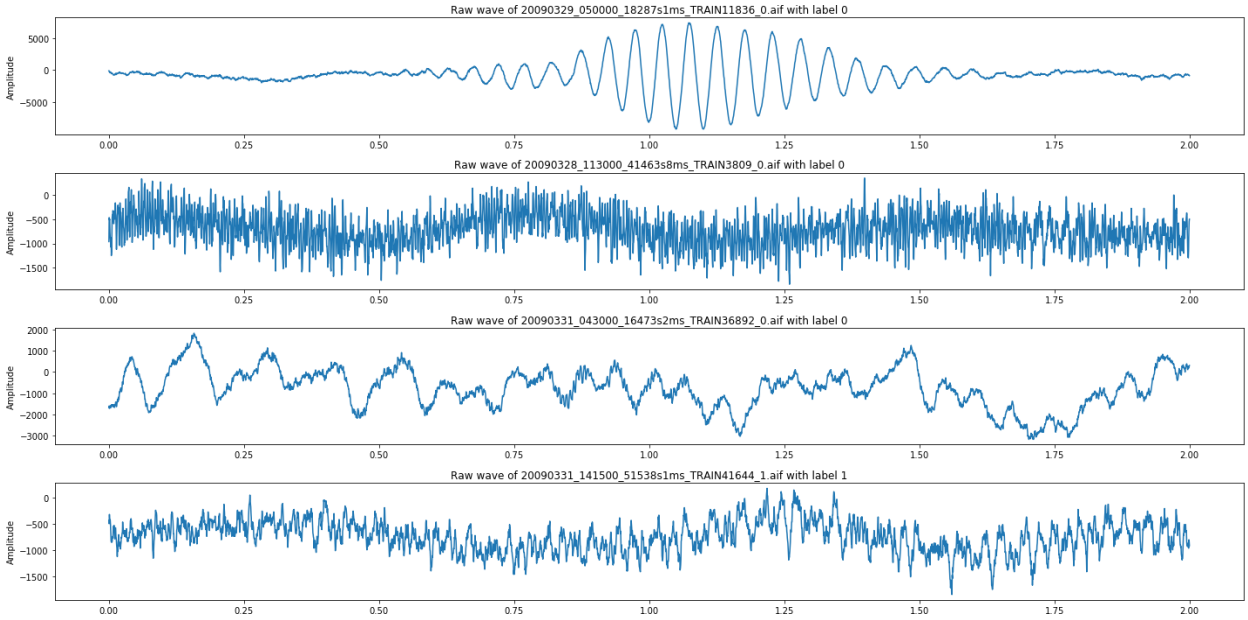
As explained before this data is a complement for the one above, however it is very different from the one above. First let's take a look up to the name of the files:

'20090328_103000_38509s5ms_TRAIN3608_1.aif'

'20090328_014500_6323s8ms_TRAIN862_0.aif'

'20090331_033000_12699s6ms_TRAIN36543_0.aif'

The names have the labels and instead of having a generic name such as before, that was ‘train1.aifc’, ‘train2.aif’..., as the competition description tells, they seem to come from different datasets, which helps increase the number of samples, however a closer look to the audio clips gives an interesting result. These are the waves for 4 randomly chosen images.



The first thing to notice is that the waves don’t look nothing like the ones before, it is possible to have a good understanding of some of these waves just by looking, for example the first one looks suspicious, as if it was artificial. This could be an issue, artificial clips could be easy to classify, making the classifier to overfit.

This is not the only thing that has changed from the previous competition, when trying to use the same software as before an error occurred, not all the audio clips have the same length, some of the clipc have around 3000-3500 frames with the same framerate, 2000 frames per second, which is around 1.5-1.8 seconds, the previous clips were all of 4000 frames with a framerate of 2000 frames per second, 2 seconds of sound. This means that the clips need to be rescaled to have the same size.

Fortunately the amount of different sized clips is really small compared to all the other clips. These clips can be ignored and since this occurs more often in the “no right whale” class it can help reduce the imbalance between classes.

So for this dataset there is a worse class distribution:

From the 47841 samples, there are only 5276 clips that actually contain a Right Whale vocalization, 11.03% of ‘1’ labels.

From those 47841 samples 5537 samples have a different length than the others, from which the class distribution is 241 whale calls and 5296 empty calls which is a 4.35% of calls between the sample that don’t have a standard length. These simples will be ignored during training.

After ignoring the non standard length samples the dataset has become slightly smaller with a 11.90% of Right Whale Calls. This is far from optimal but it can help maintain the coherence of the data.

2.5 - Kaggle Data Integrity

It seems that there could have been some problems with the data from the competition. Although it didn’t stop people from getting really high scores on the competition. Some of the audio clips tagged as Right Whales’ Up-Calls did not sound like an Up-Call, and vice versa as commented by some users. The following are two additional results that are needed to keep in mind for the particular dataset used in this competition:

- Some audio clips had very low signal-to-noise ratio (SNR).
- An audio clip tagged as a Right Whales’ Up-Call might actually be a non-biological sound or a sound from a different species.

When both occur simultaneously, things can get tricky. The energy from a right whale call might be much lower than the energy from the other sound object in the sound sample. On the other hand, some audio clips tagged as “no-call” sounded like and could appear similar to an up-call in a spectrogram.

One possible explanation for this could be that Humpback Whales, which are renowned for their vocal virtuosity, are responsible for these confounding calls. However when Humpbacks produce up-call like sounds, they typically produce them in a repetitive sequence. Thus, if a longer acoustic sample had been provided, instead of just the 2-sec clip, discrimination between a single call occurrence (i.e. a Right Whales’ Up-Call) and a sequence (i.e., a Humpback song note or call sequence) might have been more obvious, thereby improving correct classification of the sound as commented by some user on Kaggle.

Also there were people that showed that the model could be improved when adding as a parameter the size of the clip, this is due to the fact that many non whales calls were

generated artificially to increase the amount of samples, causing some models to perform better from having extra meta information.

This together with other competitors scores were calculated over the test dataset which labels are protected and cannot be obtained makes both models not comparable. Probably competitors models could outperform the ones presented in this memory since the test dataset is bigger and with much uncertainty that the train one for both competitions.

3 - Deep Classifiers

In general when it comes to audio data there are many different Machine Learning algorithms which are used on hand crafted features, this was the approach used by the majority of the community in the competition, including first and second places.

Audio data is really hard to treat, even after getting the spectrograms, there is a lot of preprocessing that can be done before applying a general purpose Machine Learning algorithm. Since spectrograms are matrices and can be considered images, even if these are not usual images, many image processing techniques can be applied to this problem. Therefore whales calls and other interesting sounds can be detected using object recognition.

One of the first approach could be to determine what makes a particular pattern interesting and then to handcraft features to detect it. Then a classifier would be used over this new feature space, many use simple classifiers such as Logarithmic Regression or SVMs. This means that for resolving this problem there needs to be the help of an expert on audio data to obtain the best results. However this is costly and not always possible, depends a lot of human interaction and is very problem dependant, since many handcrafted features can not be used in other similar problems, such as other animals sounds detection. For these reasons, a different approach has been taken: a model where spectrograms can be fed directly and has been pretty successful for object recognition and image classification, Artificial Neural Networks.

Artificial Neural Networks (ANN) have become a standard in image recognition since in 2012 when AlexNet was able to beat every other classifier in the ImageNet competition [14]. Since then ANN and more specifically Convolutional Neural Networks have evolved and are now the benchmark of image recognition industry. The book “Deep Learning” [35] has a chapter dedicated to Convolutional Neural Networks. The following section explains the basis of CNNs.

3.1 - Convolutional Neural Networks and Deep Learning

Convolutional networks [36], also known as convolutional neural networks or CNNs are a specialized kind of neural network for processing data that has a known, grid-like topology. Convolutional networks have been tremendously successful in practical applications. The name “convolutional neural network” indicates that the network employs a mathematical operation called convolution. Convolution is a specialized kind of linear operation. Convolutional networks are neural networks that use convolution in place of general matrix multiplication in at least one of their layers. Usually, the operation used in a convolutional neural network does not correspond precisely to the definition of convolution as used in other fields, such as engineering or pure mathematics. Convolutional networks, however, typically have Sparse Interactions . This is accomplished by making the kernel smaller than the input. Each kernel is

composed of a series of neurons, the kernels in the same convolution share the weights, obtaining this way trainable convolutions. This means that we need to store fewer parameters, which both reduces the memory requirements of the model and improves its statistical efficiency.

For the last 10 years there has been a boom in this technology, in part due to new research and new applications, but mainly due to the computer power now available that makes possible to train these models in a few hours instead of days. Many things have changed since the Perceptron was invented back in 1957 by Frank Rosenblatt, nonetheless the math behind is still the same and much of the original algorithm has remained as it was, after the original model several adjustments were made in the following years, Multi Layer Perceptron was created, back propagation was used for training and new gradient. After 1969 (due to the publication of Perceptrons [18]) and until 1997 there was a lack of interest in Neural Networks, later with the expansion of Recurrent Neural Networks with Long short-term memory (LSTM) [9] network sand Yann LeCun's article [15] the topic became popular again.

The main breakthroughs around this technology during this century have been around two main points:

- New ways of optimizing learning with more sophisticated activation functions, such as: ReLu SeLu, Leaky ReLu… And optimizers such as: AdaGrad, MSProp, Adam…

- Not only computer power and specialised hardware like TPUs, but also optimization of software to work with certain hardware like CPU and GPU that makes the training of huge models possible, for example libraries such as CUDA, PyTorch, TensorFlow...

All this has made possible to create what now are called Deep Neural Networks and the field of Deep Learning. These models are very appropriate for the task of Whale Call Detection for the following reasons:

- Spectrograms can be considered images and, as previously mentioned, CNNs are one of the best classifiers for working with this kind of data. Detecting a whale vocalization can be considered as an Object Recognition problem, where the object to be detected is the characteristic Right Whale Up-Call.

- The amount of data: 30000 samples is a lot of information and ANN work best when having a lot of data, this helps generalize better, more on this topic in data augmentation section.

- ANN are very effective for high dimensionality problems, able to deal with complex relations between variables.

- ANN have powerful tuning options to prevent over- and under-fitting.

- When working with sound recordings noise is a common problem and underwater this can be magnified, as mentioned before oceans are noisy places. ANN deal pretty well with noise and it can even be helpful. This matter will be expanded in the data augmentation section.

Once selected a model there are many ways of implementing it, for this project the Library called TensorFlow [29] has been selected. Tensorflow is a Machine Learning Framework for Python, focused on Deep Learning, created by Google as an internal tool and later made open source, now is one of the most popular Frameworks to work with. Tensorflow's web page [29] presents it as follows:

"TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in ML and developers easily build and deploy ML powered applications."

3.2 - Evaluating the Models

There are many ways of evaluating the performance of a model. A NN uses a function called loss and tries optimized it, this function acts as a general measure of how effective is the model predicting the final values. This is a great function of performance internally, in other words, it tells how the Network is improving in each step or how far away is the network from the values in a general way, however it is not easy to interpret and it is very problem and dataset dependent. In this case, since is a binary classification problem, Binary Cross Entropy will be used as the loss function.

Another measure that is much easier to understand is accuracy, which is just the number of labels correctly guessed divided by the total number of labels. This seems to be a good estimator if the performance, although in this case with only two classes and with some imbalance between them accuracy doesn't seem as good as an option, one class have much more weight than the other. Other similar metrics that help with distincting between classes are:

- Precision: Fraction of the number of correct predictions between values predicted as True for a class.
- Recall: Fraction of the number of targets correctly predicted in the Positive class.
- F1 Score: Harmonic mean between Precision and Recall for that class.

This is calculated by building the confusion matrix. Confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of a model, each row represents the predicted class while each column represents the actual class. So the layout is the following:

	Actual Positives	Actual Negatives
Positive Predictions	True Positive (TP)	False Positive (FP)
Negative Predictions	False Negative (FN)	True Negative (TN)

So the previously mentioned metrics would be rewritten as:

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

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

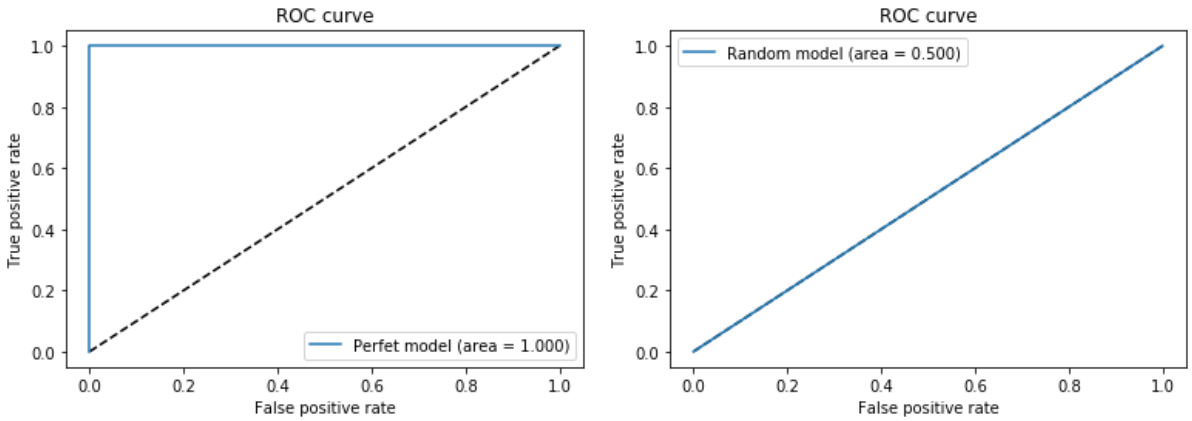
$$Precision = \frac{TP}{TP + FP}$$

$$F1 - score = \frac{2 * Recall * Precision}{Recall + Precision}$$

All these metrics can help, but it depends on the objective of the model. Recall tells the prediction accuracy among only actual positives. It means how correct the prediction is among whales. That could matter in this case. When False Positives can not be tolerated, precision should be favoured. A model for spam detection serves as a great example for this. F1-Score on the other hand is a balance between both, what could be an alternative to accuracy.

In this problem False Negatives and False Positives are not so determining, a general balance between predicting both classes is preferable. F1 score works for a class, it measures general performance over a class, which could work over the Right Whale class. Nevertheless, there is a different metric that measures the separability of two classes and was used in the original competition; Area Under the Curve Receiver Operating Curve, AUC for shortening. This was the metric used in both competitions as the score of the model.

This metric is calculated by drawing the curve that connects the points of the True Positive Rates in the y-axe and False Positive Rates in the x-axes for different thresholds and calculating its area. A perfect model would have 100% certainty about the class of each sample so the ratio would be always 1 making the curve a square with area 1.



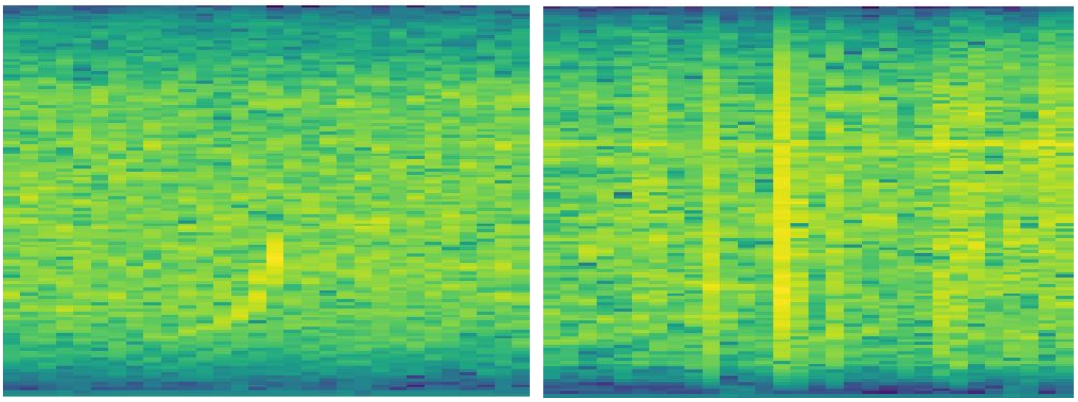
Left image shows a perfect model, right one shows the random baseline.

A random model would separate randomly the samples having a ratio linearly dependent of the number of samples on the threshold being a straight line that cuts the square in half. A model that has a curve under the middle line separates the classes but it does it in the opposite way, classifying one class into the other and it would have an area under 0.5. In general a good model should have an area as close to 1 as possible. Note that there is some correlation between accuracy and AUC, the area is determined by the accuracy in each class for different thresholds, ie the ponderated accuracy (making both classes have the same weight for calculating the accuracy) for each threshold, although models with high accuracy can have low AUC and models with lower accuracy can have higher AUC.

So for measuring the models the AUC will be use as final determiner but all the metrics will be looked at to make sure how is working each time.

3.2 - First Models for Right Whale Detection

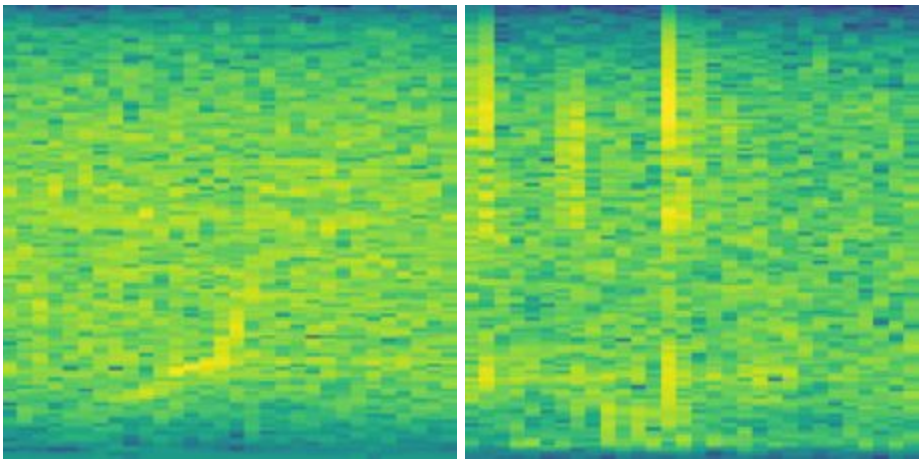
Once selected the tools and software to use the next step is creating the network, the first layer takes the Spectrogram images and feed them to the Convolutional Layers. The images have a resolution of 531x398 when the spectrogram are created, they look something like this:



Left one has a whale call, whereas the right one is just background noise.

For the first part the images were scaled to a different dimension, all of them were squared so pretrained general purpose Neural Networks could be applied which are constructed over square images and for convenience. The image size is considered too big, 531x398 with the three channels means that the first layer needs to have around 634014 parameters for each of the neurons in the convolutional layer, which make this network hard to manage, so the images have been rescale to 3 more manageable sizes: 224x224, 128x128 and 32x32.

The images change slightly but still the whale vocalization can be easily recognized.



The images above where rescaled to fit on the page, however these ones are in the same dimensions that are fed to the neural network.

3.2.1 - Result for image scales

After several failed attempts, two final networks were created, and each trained with different scaled images. The first networks try to follow the original AlexNet architecture reduced to more manageable size and for a simpler problem that is this one (ImageNet had over 15 million labeled high-resolution images belonging to roughly 22,000 categories). The second was based on the one presented by one of the top three teams in the Kaggle competition who used CNNs.

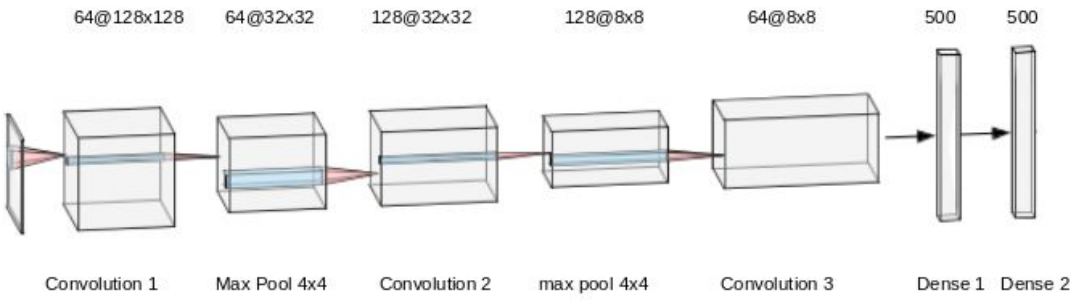
AlexNet had an Architecture composed of five convolutional layers and three dense layers. The convolution layers were the following: 96 11x11 filters, 256 5x5 filters, 384 3x3 filters, 384 3x3 filter and 256 3x3 filters, all of them separated into two, so it could be trained into two separate GPUs. Then two dense layers with 4096 neurons each were added and a final 1000 dense layers with softmax activation for the output.

The first network keeps the two dense layers, but downsized to 500 neurons which proved to be more than enough. Only three convolutional layers were kept, each with a max pooling between layers. For smaller images, the ones with 32x32 only one max pooling was kept and kernel sizes were reduced to keep dimensions from getting to zero.

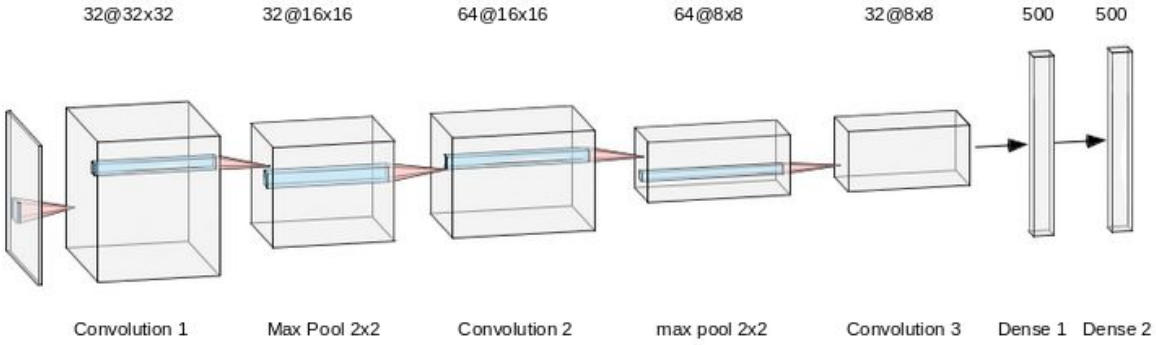
The first CNN is as follows: 64 7x7 filters, 128 5x5 filters, 64 3x3 filters, then two dense layers of 500 neurons each and a dense layer with 2 neurons and soft max activation function.

For the smaller images this CNN is smaller: 32 7x7 filters, 64 5x5 filters, 64 3x3 filters, them two dense layers of 500 neurons each and a dense layer with two neurons and soft max activation. The following pictures contain a diagram of these networks at the style of AlexNet diagram.

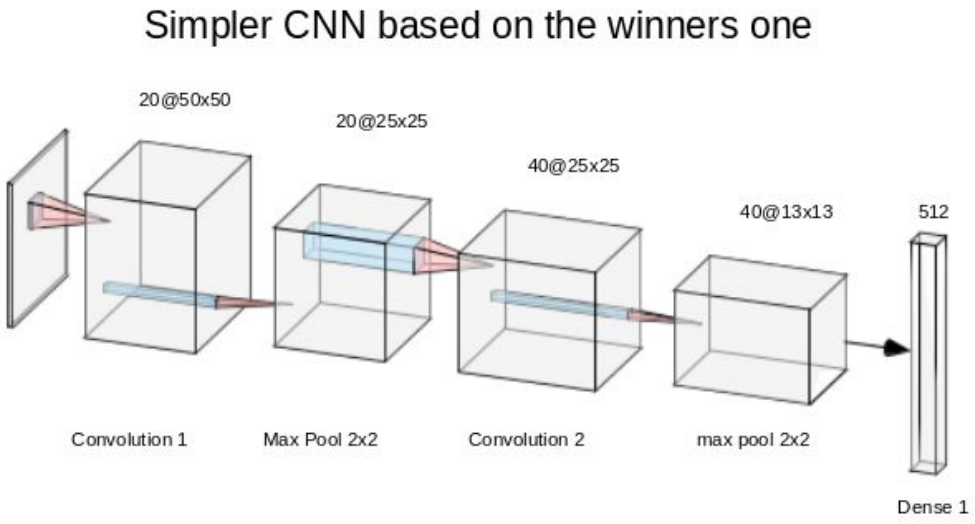
CNN for 128x128 images



CNN for 32x32 images



The second is a much simpler network, prepared to be trained fast, with only 2 convolutions and 1 dense layer. On this network there was also applied DropOut, in the first Convolution and in the dense layer.



The networks based on Alexnet were able to get much better results than the one based on the winners network. For now, this network will be the focus and the winners one will be left for later. The maximum accuracy on the test dataset was around 0.79, since there is around a 76% percent of samples containing no Right Whales' Up-Call it barely improves the accuracy of a model that classifies everything as no call or a random model. This results weren't as good as desired and since there was no validation split, the training was blind, so a validation split was created for the experiments from now on, containing the 20% of the training samples.

The main difference between the image size was the network size and computation time, for small images it is needed a much smaller network and the overall weight of the network and data is smaller, the number of parameters is decreased and thus the training time is lower. As the models performance are similar from the different scenarios, from now only the smallest images will be considered. This way the number of models is decreased as the training time. For now on only two networks will be considered, the one similar to the winners one and the one similar to AlexNet.

The best results with this configuration was obtained with the smallest of the networks. This does not mean that is a better model, since it is smaller and has less parameters the training is faster which allows a better optimization of parameters. This results are displayed for an accuracy of 0.8956 over 100 epochs:

	Actual No Right Whale	Actual Right Whale
Predicted No Right Whale	4387	413
Predicted Right Whale	213	987

Between the 6000 samples on this particular split there are 1400 samples containing a right whale call with:

$$\textit{Right Whale precision} = 0.82$$

$$\textit{Right Whale Recall} = 0.70$$

$$\textit{Right Whale f1-score} = 0.76$$

$$\textit{AUC ROC} = 0.7336$$

The network was able to improve the baseline (classifying all samples as Right Whales), although there is also plenty of distance between the AUC and the desired value.

The model seems to have a problem finding what defines a whale call. The spectrograms have a problem: the part containing a whale call is really small, the vocalizations last for less than half a second, which means that more than 75% of the duration is of no use. Apart from that, whale calls have a frequency between 50 and 300 Hz, the lower frequencies are cut by the spectrograms automatically because this kind of sounds are not even picked by the computer and are only used in dedicated problems, the higher ones are mainly from background noise.

This means than a huge portion of the image is just noise that could be discarded. Knowing when a whale call starts it is really difficult, they are more or less centered but still there are lots of variations in the starting point. However the top frequency is a known value that have been recorded in various studies and can be ignored for a better definition of the calls. This way the training could be sped up and accuracy boosted.

3.2.2 - Cropping the images

After a look at spectrograms the exact cut point is located around 150-200 pixels from the bottom. To be sure there is no information loss the cut is done in the 200.

The images have been cropped before resizing them, maintaining the 32x32 size, now with less scaling, since the ratio of pixels is bigger and more information is preserved. This shows an improvement in accuracy of a few points, that could be thanks to the elimination of noise and preserving more information. The mean accuracy for several runs in a 5-fold Cross Validation (0.2 test split in each fold, 6000 samples) for 100 epochs each is 0.8966. Other metrics for this model are the following:

	Actual No Right Whale	Actual Right Whale
Predicted No Right Whale	4262	338
Predicted Right Whale	282	1118

Between the 6000 samples on this particular split there are 1456 samples containing a right whale call with:

Right Whale precision = 0.77

Right Whale Recall = 0.80

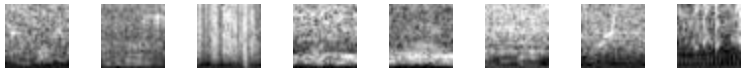
Right Whale f1-score = 0.78

AUC ROC = 0.7554

These results are from one of the 5 folds, the rest are quite similar. As shown by the Confusion Matrix the worst performing class is the one with the right whale calls. Even if the accuracy is high, the imbalance in the problem favours samples with no Right Whale Calls.

3.2.3 - Gray scale image and Spectrogram Parameters

Until now colored images have been used. However, as mentioned before, the coloring is only a mapping use for humans to be able to see the intensity for each pixel and make it easier to understand the representation. A Neural Network does not need a colored representation to understand the images so the spectrogram were transformed into gray scaled images. Now, with only one channel, if the model is able to perform as well as it has been doing it means that from now on the network could only have one channel, meaning that the amount of parameters is lower and the training faster. Here some of the new spectrograms are shown, they are shown in scale, meaning these are the images which the network is being fed.



As can be seen they are small and gray, however after running the same experiments the accuracy of the Neural Network remains the same while the complexity of the data has decreased significantly and thus the number of weights in the network, which means that the training becomes less computationally intensive and needs less time to be done, so more experiments can be run to optimize high level parameters.

The parameters have been settled to the values of the last experiment and the spectrograms have been resized and cropped. The first Neural Network has been trained using the Stochastic Gradient Descent, with a Learning rate of 0.01, a decay of 0.000001 and a momentum of 0.9, also 5 fold Cross validation has been used, training each time for 150 epochs. The best accuracy is 0.9117. Other metrics for this model are the following:

	Actual No Right Whale	Actual Right Whale
Predicted No Right Whale	4320	227
Predicted Right Whale	303	1150

Between the 6000 samples on this particular split there are 1377 samples containing a right whale call with:

Right Whale precision = 0.79

Right Whale Recall = 0.84

Right Whale f1-score = 0.81

AUC ROC = 0.8637

The performance has improved hugely, gray scale images combined with the cropping have left to huge improvement over all the metrics, mainly on AUC. Note that since the training is much faster this time the model was trained for 150 epochs which may also be the reason for the improvement, with previous setups so many epochs supposed too much time of training.

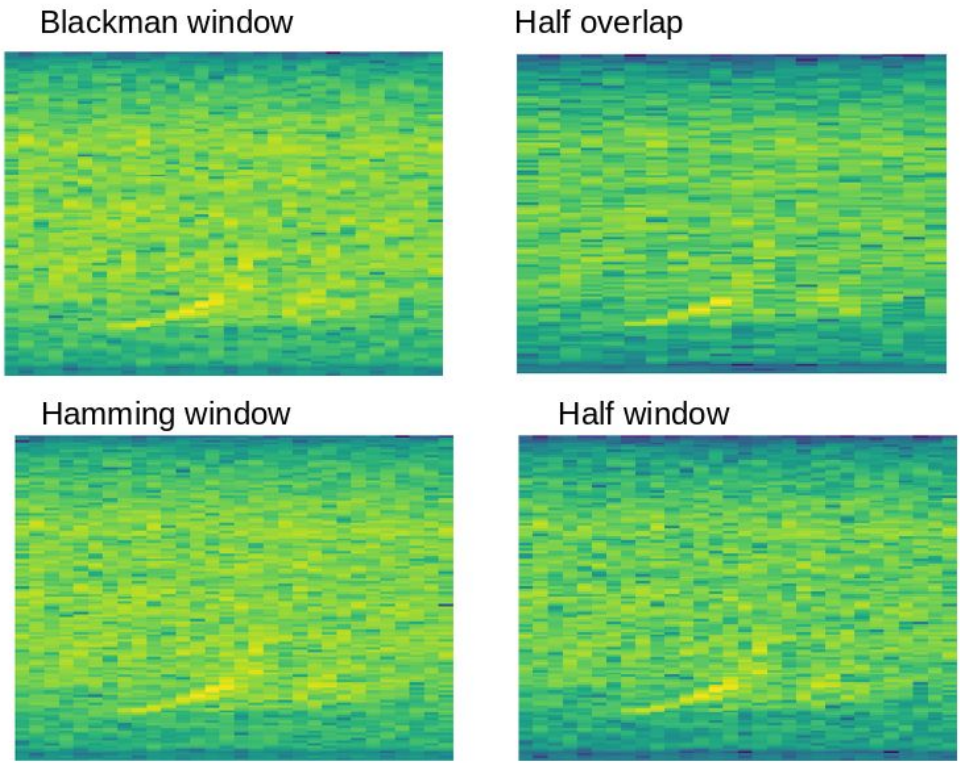
The final image has been fixed, however there is still something that can be tweaked apart from the sizes and colours, namely the way the spectrograms are created. As mentioned before in the chapter 2 there are several parameters that can be modified, the most importants ones been window, window size and overlap. These were already

settled to the recommended ones, however there are other parameters to explore. For this, 4 different scenarios have been constructed:

- Same Window Half the overlap
- Same Window half the window size
- Same NFFT and overlap with Blackman window
- Same NFFT and overlap with Hamming window

Let's see if there is some improvement. Before continuing, note that the Window size and overlap number directly affect the time to obtain the spectrograms, the bigger both numbers the higher the complexity.

The results of each of the different preprocesses using the different parameters is not presented since it was similar, there is no clear preprocessing better than the rest for this problem. The Blackman Window yielded slightly better results but it is not statistically significant compared to the second best results. The following image shows the differences between spectrograms:



3.2.4 - Data augmentation and Regularization

A new approach was tried to improve generalization: Data Augmentation and Regularization. When dealing with images, Image Augmentation is usually used for

Deep Learning, i.e. artificially increasing the number of samples to increase the dataset. The most common techniques are:

- Shift: Shifting the image in any direction so the model learns that the object can be placed anywhere in the image and focus on the context.
- Rotation. Rotating the image for a more general approach, the objects remain the same even if they are up and down.
- Zoom: Zooming certain parts of the image so the model has a more general idea of the object and don't focus only on the pixel level.
- Rescaling: Images can have different resolutions. For some problems scaling the image up and down losing some resolution can help with generalization.
- Histogram Equalization: Increases global contrast of an image using the image intensity histogram.
- Contrast stretching: enhances the contrast by rescaling ("stretching") the range of intensity values of an image to a desired range of values.

These techniques are used in Object Recognition tasks. However, for this problem it cannot be directly applied since spectrograms unlike images have dimensions, time and frequencies. The relationship between them needs to be preserved which discards Rotations and Zooms.

The others techniques can be applied, but have to be reinterpreted to maintain the spectrogram representation, the following augmentation techniques are presented:

- Horizontal Shift
- Adding noise

Horizontal shift has been applied by instead of taking the whole two seconds audio clip, only a fraction have been taken three times for each clip, one starting at the beginning, one in the middle and one in the end, this way there are three clips containing the calls but slightly moved, in the original all whale calls where centered. This improves generalization. After several tests and experiments the optimal value seemed to be 1.7 seconds. Several calls are not so centered and could be caught if smaller values were taken. This was done before creating the spectrograms and rescaling the model, which again, as cropping did, helps maintaining more information when rescaling since the images are already smaller.

For the noise addition there are many algorithms that can be used, however in this case the images without whale calls were used. These images, in the majority of the cases, contain just background noise, this is mostly random and with very low intensity sounds from the ocean. This data augmentation technique was only used on the Right Whale containing examples because adding random noise to samples already composed with

noise makes little sense and this way the imbalance between classes is reduced. Noisy samples were multiplied by a constant 0.25 before adding them to the original to reduce the effect and them renormalized the images.

With the applications of these techniques training dataset is 3.7 times bigger and the percentage of Right Whale containing samples is about 40% instead of 23%.

For this problem the original test spectrograms were generated in the same way as before, with no data augmentation for it to be comparable. In fact the test datasets are the same but only taking the middle 1.7 seconds in order to be comparable. The results are the following:

After a 5-hold the average accuracy is of 0.8967 after taking a better look at each fold, the confusion matrix for the worst fold looks like:

	Actual No Right Whale	Actual Right Whale
Predicted No Right Whale	4262	282
Predicted Right Whale	338	1118

Between the 6000 samples on this particular split there are 1400 samples containing a right whale call with:

Right Whale precision = 0.77

Right Whale Recall = 0.80

Right Whale f1-score = 0.78

AUC ROC = 0.8663

So the model has a smaller accuracy than before, however the precision, recall and AUC remains the same. The training dataset accuracy is about 1 and loss is about 0, which means there could be overfitting, i.e. the model has become very well predicting the training data set and is not able to generalize even after augmenting the data.

To reduce this effect there are some workarounds. One modern approach for this is using Dropout. Dropout [44] provides a computationally inexpensive but powerful method of regularizing a broad family of models. The way it works is by randomly deactivating some units in chosen layers, in general a proportion of units is chosen to be dropped in each batch and the selected units are ignored for that training batch. It reduces training accuracy but also overfitting. Dropout provides an inexpensive approximation to training and evaluating a bagged ensemble of exponentially many

neural networks. Specifically, dropout trains the ensemble consisting of all subnetworks that can be formed by removing non output units from an underlying base network. In most modern neural networks, based on a series of affine transformations and nonlinearities, we can effectively remove a unit from a network by multiplying its output value by zero. An intuitive way to understand it is to consider each subnetwork with the dropped out units as a model on each own so as each model learns a different part of the space yielding an overall better knowledge of the data space. It is a simple yet powerful way of avoiding overfitting.

After the addition of the Dropout the results for the experiment are the following, with an accuracy of 0.9240.

	Actual No Right Whale	Actual Right Whale
Predicted No Right Whale	4350	208
Predicted Right Whale	252	1190

Between the 6000 samples on this particular split there are 1398 samples containing a right whale call with:

Right Whale precision = 0.83

Right Whale Recall = 0.85

Right Whale f1-score = 0.84

AUC ROC = 0.8996

Thanks to Dropout, the model was able to improve accuracy, f1-score and AUC, But still the AUC is not close enough to 1. Accuracy have been boosted to the maximum value until now, on the other hand AUC does not surpass 0.9. There are still many points that can be improved, one of them being the representation. In the next section a different approach to the problem is presented. This approach will give more control onto the spectrogram representation.

3.3 - Spectrogram representation

Until now the way of working was by creating the spectrograms and saving them in disk which required a lot of time. This permitted the use of Keras functionality to load the images from disk in batches and not needing to calculate the spectrograms each time.

Although working with pure images may not be desirable, saving and reading from disk has many advantages. One of the drawbacks of this method is the library used to save and manipulate images. In order to treat spectrograms as images, Matplotlib does some preprocessing that may be avoidable and controlled for better results. After running a bunch of different experiments and seeing no performance improvements. Calculating the spectrograms and keeping them in memory is going to be used from now on to try to improve results.

When saving the spectrograms as images a library was used that preprocessed the spectrograms to an image format, giving it the colour scale and scaling the values. By not using this process there is the possibility of improving the quality of the spectrograms by using other kinds of preprocesses. Directly working with spectrogram matrix presents a new problem, since the matrices with the raw data have many drawbacks:

1. The original dimensions are bigger, as there is a much larger range of frequencies.
2. Values range from 0 to infinity, there is no limit in the value, opposed to images where values may vary between 0 and 255 or 0 and 1 (in gray scaled images). Input normalization is required.
3. Human interpretation is harder, raw spectrograms are matrices.
4. Data enhancing techniques still work but needs to be adapted, not all techniques could be used.

Problems 1, 3 and 4 are easily overcome. The coefficient matrices can still be considered as images and use very similar algorithms. The point 2 requires a bit of care, since there are many ways of doing this and not all of them yield the same results. As mentioned before, a logarithmic transformation was used to obtain the original images, yet it could not be the optimal preprocess. In the following section the different approaches to the normalization problem will be discussed.

3.3.1 - Normalization

Neural Networks work best with Normally Distributed data, with mean close to 0, and it is preferable to have values no bigger than 1 which helps avoiding exploding gradient and makes training faster [28]. For this reason 4 different pre processes have been proposed.

Previous images well calculated with a software that transformed the spectrograms into values between 0 and 255, [0, 255], and later fed to the ANN in a scale [0, 1]. The first calculation was done by calculating the logarithm of the values and then scaling it to 255 by dividing by the highest value and then multiplying by 255. The mathematical formula is presented below. Note that 1 is added to the values before the logarithm is calculated in case the value is *0s*.

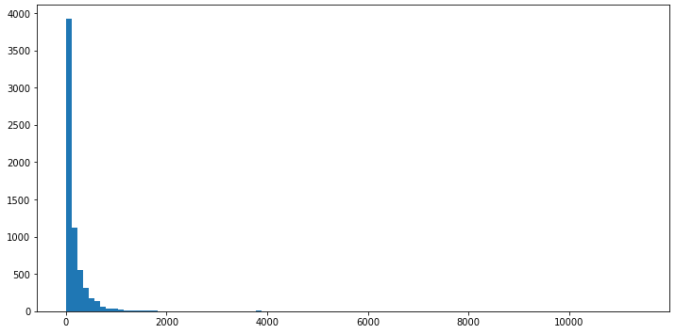
$$y_{new} = \frac{log(y_{old} + 1)}{log(max(y_{old}) + 1)} * 255$$

Later this value was rescale to [0, 1] by dividing by 255 to feed it to the network. This may not be the best way to normalize data, many experiments [4] has shown the importance of normalization of data to ensure a fast training convergence and reduce overfitting, Normalization also helps with other usual problems of NN such as: Exploding gradient and Vanishing gradient.

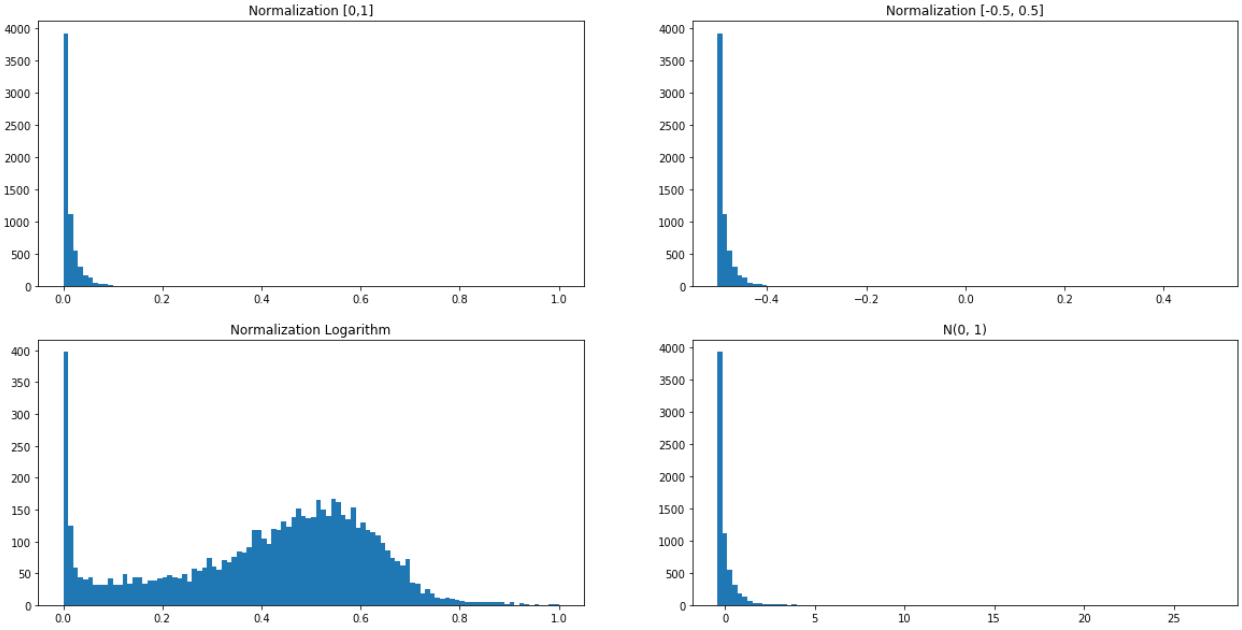
For this problem, 4 different normalization functions have been proposed:

- Dividing by the maximum to ensure all values are between 0 and 1.
- Logarithmic scale and rescale to [0, 1].
- Rescaling values between -0.5 and 0.5.
- Normalize each spectrogram to variance 1 and mean 0.

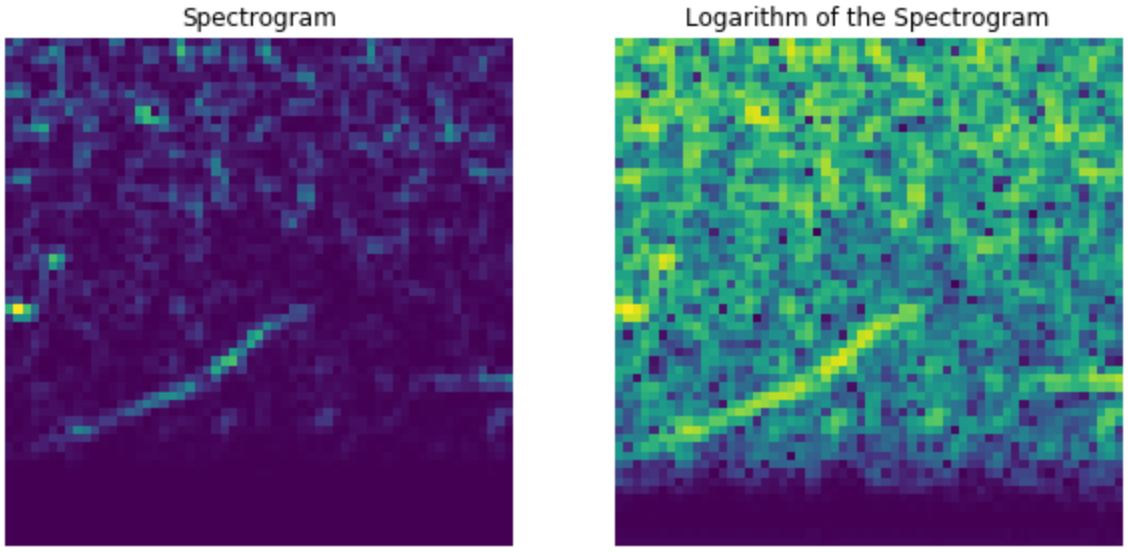
Dividing by the maximum is the simplest way of normalization and will be used as a baseline to compare with the rest of methods. The values in a Spectrogram are distributed in the following way:



After applying normalization the distribution of the values in the spectrograms is the following:



The only function that changes the distribution is the Logarithmic function, since it is a monotonous function it does not change the values ordering, only make higher values closer to lower values, having a more interpretable distribution. The original spectrogram image changes:



Spectrogram Normalization changes the values making the call more visible but also the rest of the image. This could be positive, the network learns the better what is and

what isn't a whale call, or could be negative making everything too similar for the network to separate.

With this setup and after cropping the spectrograms to ignore high frequencies, the size of the matrices is of 50x50. For these dimensions the second Deep CNN will be used, the one proposed before for the images with size 32x32. The other network that will be compared to this one is the one based on the Kaggle's third positioning model. This network is composed of 2 convolutional layers of 20 and 40 convolutions respectively and a kernel size of 7x7 each with linear regularization and Dropout of 0.2, a 2x2 max pooling layer after each convolution; a dense layer with Dropout of 0.6 of 512 neurons and a final dense layer with 2 neurons. All the layers make use of the ReLu (rectified linear unit) activation function except from the last that uses Softmax function. The original network created by team that got the third place in the competition had a slightly different architecture, they used one neuron in the output layer with nonlinear activation function and fewer Dropout and Convolutions. Their network was performing worse with this setup and these changes were made to adapt the network, since their preprocess wasn't detailed enough.

After several runs with the 32x32 network based on AlexNet and used in section 3.2 First Models for Right Whale Detection was ignored. The performance of the network was disappointing. After changing the set up the time needed for training the augmented dataset went from an average of 119 seconds per epoch and 0.0031 ms per sample to 1724.32 seconds per epoch and 24.3ms per samples, making the training sessions much larger with far worse results. The best result with an accuracy of 0.8370 and around 24 hours training for 150 epochs, no cross validation was used for lack of time. The results are the following:

	Actual No Right Whale	Actual Right Whale
Predicted No Right Whale	3881	277
Predicted Right Whale	700	1142

Between the 9000 samples there are 1419 samples containing a right whale call with:

$$\textit{Right Whale precision} = 0.62$$

*Right Whale Recall = 0.*80

*Right Whale f1-score = 0.*70

*AUC ROC = 0.*8259

After these results and the time needed to run each of the experiments this model was left aside. The model seems to struggle with this data in terms of time and convergence. The other model based on the original competition winners with 2 convolutional layers and 2 dense layers performed much better. The amount of test samples was increased and a validation split was added, which reduced the amount of training samples. With the baseline normalization (diving by the maximum) the model was able to get an accuracy of 0.9162. The results are presented below.

	Actual No Right Whale	Actual Right Whale
Predicted No Right Whale	6567	457
Predicted Right Whale	297	1679

Between the 9000 samples on this particular split there are 2136 samples containing a right whale call with:

*Right Whale precision = 0.*85

*Right Whale Recall = 0.*79

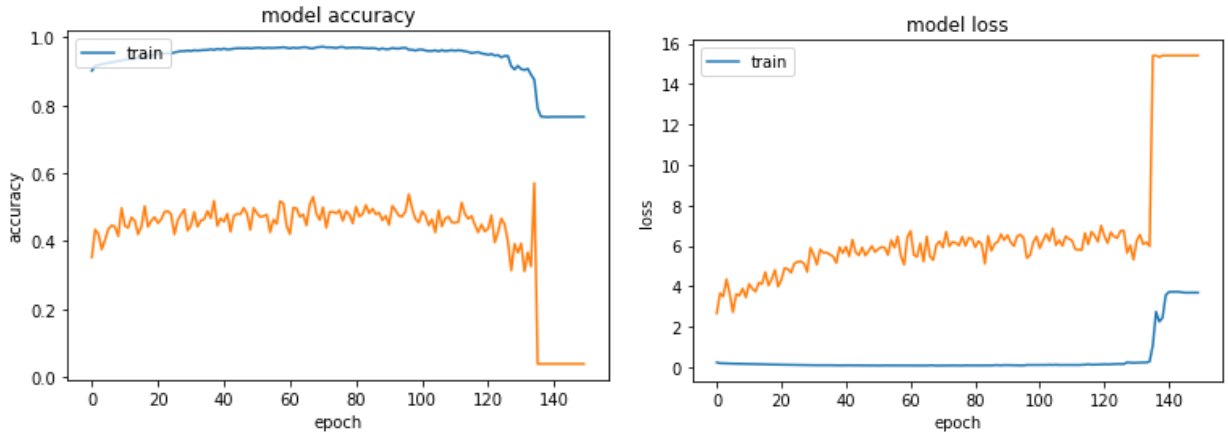
*Right Whale f1-score = 0.*82

*AUC ROC = 0.*9697

these readjustments were able to keep the same accuracy but improve the AUC incredibly, getting closer to the Kaggle competition results. With the baseline calculated the other normalization techniques are presented below:

	Accuracy	AUC	Whale Precision	Seconds per epoch
Baseline Max Normalization	0.9162	0.9697	0.85	155s
Log Normalization	0.9213	0.9724	0.85	154.2s
[-0.5, 0.5] normalization	0.9173	0.9673	0.82	162.7s
μ=0, σ=1 Distribution Normalization	0.9168	0.9689	0.82	179.4s

In this table a summary of the results obtained by the experiments with each normalization can be seem. Note that the time per epoch is an approximation, other normalizations apart from the last one performed similarly. The one that performed best was the Logarithmical one, the detailed results are presented later. The rest performed more similarly, however the one with mean 0 and standard deviation 1 was the worst performing, not only had less accuracy and AUC but also the model needed to be trained several times, it was really prone to fall to a local minima causing the training to get stuck and have a random performance. The following graph presents the evolution of training and validation loss and accuracy:



Both graphs represent a normal training until the Network weights stuck in a local minima causing a negative performance. This only happened with this normalization

and happened often. Before the details of the [0, 1] normalization was presented, the details for the rest are presented below:

Logarithm normalization with an accuracy of 0.9213:

	Actual No Right Whale	Actual Right Whale
Predicted No Right Whale	4363	259
Predicted Right Whale	213	1165

*Right Whale precision = 0.*85

*Right Whale Recall = 0.*82

*Right Whale f1-score = 0.*83

*AUC ROC = 0.*9724

For the [-0.5, 0.5] normalization, with an accuracy of 0.9173:

	Actual No Right Whale	Actual Right Whale
Predicted No Right Whale	4354	247
Predicted Right Whale	249	1150

Between the 9000 samples on this particular split there are 1397 samples containing a right whale call with:

*Right Whale precision = 0.*82

*Right Whale Recall = 0.*82

*Right Whale f1-score = 0.*82

*AUC ROC = 0.*9673

For the mean 0 and standard deviation 1 normalization, with an accuracy of 0.9168:

	Actual No Right Whale	Actual Right Whale
Predicted No Right Whale	4367	265
Predicted Right Whale	234	1134

Between the 9000 samples on this particular split there are 1369 samples containing a right whale call with:

*Right Whale precision = 0.*83

Right Whale Recall = 0.81

*Right Whale f1-score = 0.*82

*AUC ROC = 0.*9689

Normalization helped to get the final edge on the results to maximize the AUC and accuracy. It proved that can be helpful when done right but also harmful if not taking care of it.

3.3.2 - Data augmentation Kaggle Competition Redux

Once selected the best parameters and data augmentation techniques there is still another way to increase the data. As mentioned in the Section 2 - Data there is another secondary dataset that complements the original one, however it has many drawbacks. After several attempts and data cleaning the data set is ready to be used.

First of all, the best of the models were trained with this data to check how the model performs on this dataset. Test set contains 20% of the whole dataset 8446 samples, after data augmentation train set have 113718 samples, both with a size of 50x50. For the whale call proportion on the whole dataset there was a 11.03% of Right Whale Calls. For this experiment the train set contains a 21.36% of whale calls after data

augmentation and test set contains 11.69%. Spectrograms were normalized using the Logarithm Normalization.

After 150 epochs the model got an accuracy of 0.9666, which is greater than the one for the original dataset. The Confusion matrix and other metrics are the following:

	Actual No Right Whale	Actual Right Whale
Predicted No Right Whale	7371	125
Predicted Right Whale	123	839

$$\textit{Right Whale precision} = 0.87$$

$$\textit{Right Whale Recall} = 0.87$$

$$\textit{Right Whale f1-score} = 0.87$$

$$\textit{AUC ROC} = 0.9916$$

The first results with the first run of the best of the model was impressive. The results were even better than for the first dataset. Even though the classes suffer for a bigger imbalance the AUC, Whale Recall and Weighted average were better. Thus the model was able to achieve a better separation of classes, ie. the dataset is easier to separate. The top results were easily obtained, but is this trained model able to to perform equally well on the previous dataset? The following results are the evaluation of this trained model in the original test:

The accuracy of the model is 0.8526, better than a random model but not as good as the accuracy with the test set. Closer look with the confusion matrix and the rest of the metrics is presented below:

	Actual No Right Whale	Actual Right Whale
Predicted No Right Whale	21121	2569
Predicted Right Whale	1852	4458

*Right Whale precision = 0.*71

*Right Whale Recall = 0.*63

*Right Whale f1-score = 0.*67

*AUC ROC = 0.*8858

The model is not able to classify whales on the first data set, only a 10% of Whale Calls were considered as such by the model. On the other the No Right Whale Call class got a precision of close to 1. As for AUC and f1-score the model is not able to correctly identify the whale calls on this data set. This could be due to the model being not able to find the features that define a Right Whale call or because both database whale calls are defined differently, sound recording techniques or this particular whales' calls could have varied.

For this the previous trained model on the Kaggle's original dataset was used to classify this the Redux's samples. The following results were obtained, with an accuracy of 0.9203:

	Actual No Right Whale	Actual Right Whale
Predicted No Right Whale	37044	3146
Predicted Right Whale	225	1889

*Right Whale precision = 0.*89

*Right Whale Recall = 0.*38

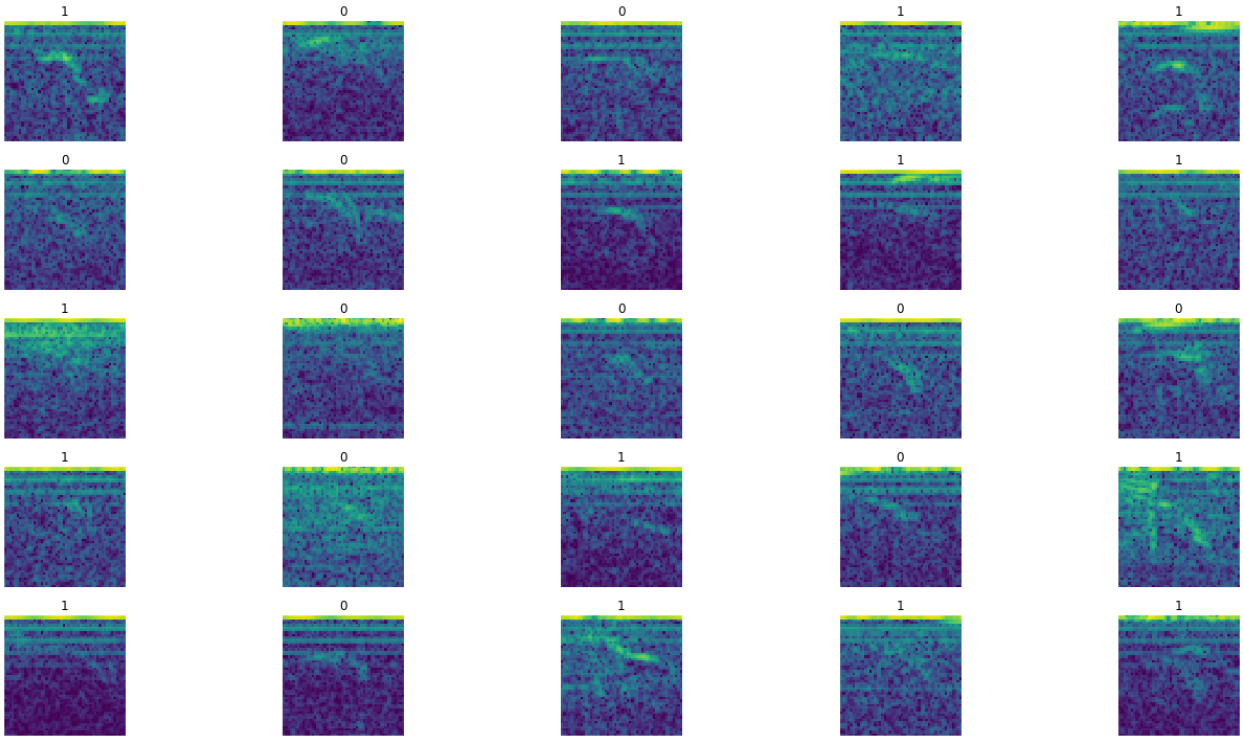
*Right Whale f1-score = 0.*53

AUC ROC = 0.9315

As can be seen both dataset seem to be different, the same model trained on each of the dataset is not able to classify correctly the other.

3.3.3 - Analysis of the results

The best results were obtained with the spectrogram representation saved in memory and the simpler CNN based on the competition, the one with high dropout, was able to achieve an accuracy of 0.9213 and an AUC of 0.9724 on the Kaggle's Original Competition. On the Competition Redux was able to get even better results with an accuracy of 0.9666 and an AUC of 0.9916. There are few spectrograms in both models that were classified incorrectly, some of these are presented below.



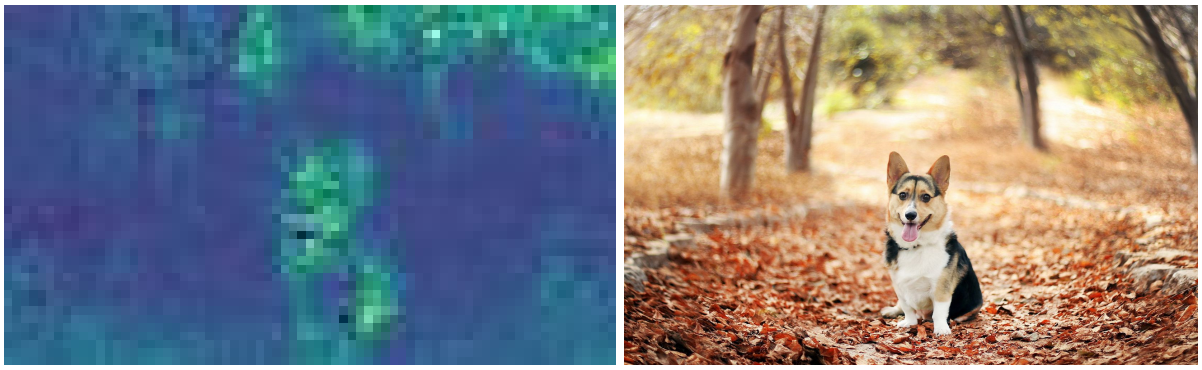
The top number of each image is the class, 0 for not having a Right Whales' Up-Call and 1 for having one. The last process used for obtaining the spectrograms takes the 0 value on top of the image giving an inverted representation, equally valid anyway. As can be seen there are some images classified with the 0 label that have some kind of sound similar to an Up-Call, probably from another whale species. On the other hand there are many clips labeled as 1 that since to have too much noise.

Some of these errors could be solved with a better dataset or even a better representation. Some others are not able to be recognised even by a human, the label could be wrong or just the quality of the clip is not good enough to be correctly classified.

3.4 - Using Neural Networks for feature extraction

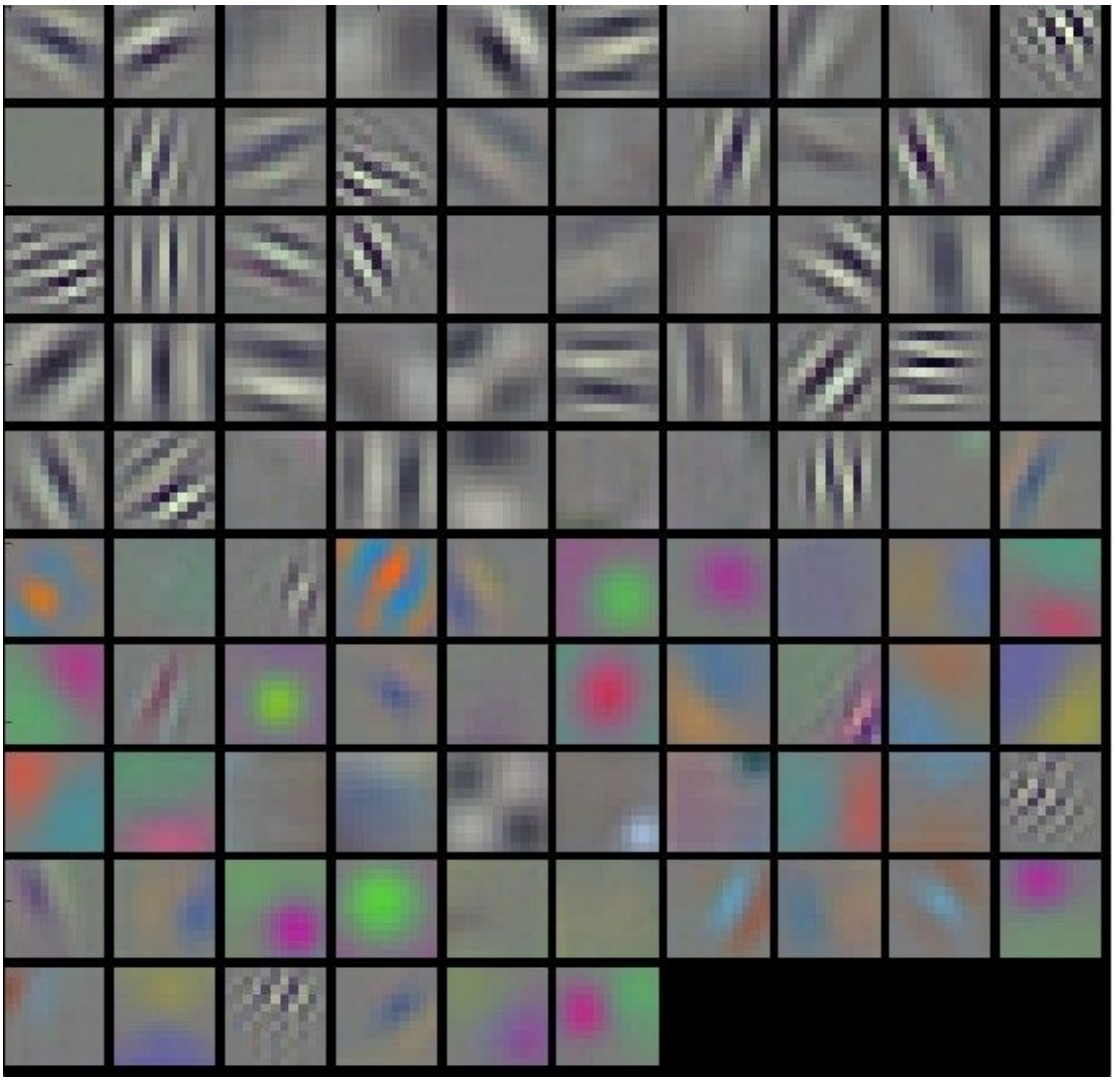
As mentioned in Section 1.3 - Previous Work, there has been some experiments [42] that have used already trained CNNs for feature extraction, avoiding the requirements of using hand crafted or predefined templates. It is an interesting experiment since it avoids the training phase of large CNNs that is time consuming by using well known and trained networks.

Last layers of CNNs are almost always dense layers. These layers are considered to be the real classifiers, using the features and transformations obtained in previous layers are able to obtain amazing results. This last classifier could be very simple, often using a single dense layer, ie Single Layer Perceptron[17]. Being the Convolutional layers the real hard workers, that are able to create complex transformations that separate classes and find the features in the data. Being NN the black boxes they are many people have tried to visualize these features, recomposing the images from the convolutional layers and finding the most excited neurons (the ones which are activated and yield the highest values) for example:



In this image it can be seen how the pixels around the face of the dog are the ones that yield the highest values for this layer of the convolution, showing that the convolutional layer detects the dog. These values are completely dependant on the data used for training and the convolutional layer chosen. This image was built using the code from a repository called Heuritech [38], there more examples can be looked at.

If the data is more general, such as ImageNet data with 14 million images and more than 20 thousand classes, the obtained filters in the CNN are more general and can be transferred to other problems:



This image shows the 96 filters in the first layer of the CNN, each of the layers get excited by a different kind of shape, acting almost like classical features from Image Recognition.

These filters can be helpful for our problem, so we could use them to obtain some features and be able to train a simpler and faster models over this filters values. This process is called Transfer Learning [37], it is the ability of learning in a new task through the transfer of knowledge from a related task that has already been learned. If the tasks are close this fastens the training and ensures a better convergence. This could be achieved, by retraining a network, i.e. using the already trained network as the starting

weights of the network. In this case it is still really costly since the networks that we have already trained are much simpler than the one from ImageNet and any other network used for such general problems, for this reason a simple classifier will be trained for the features obtained after the Convolutional Layers.

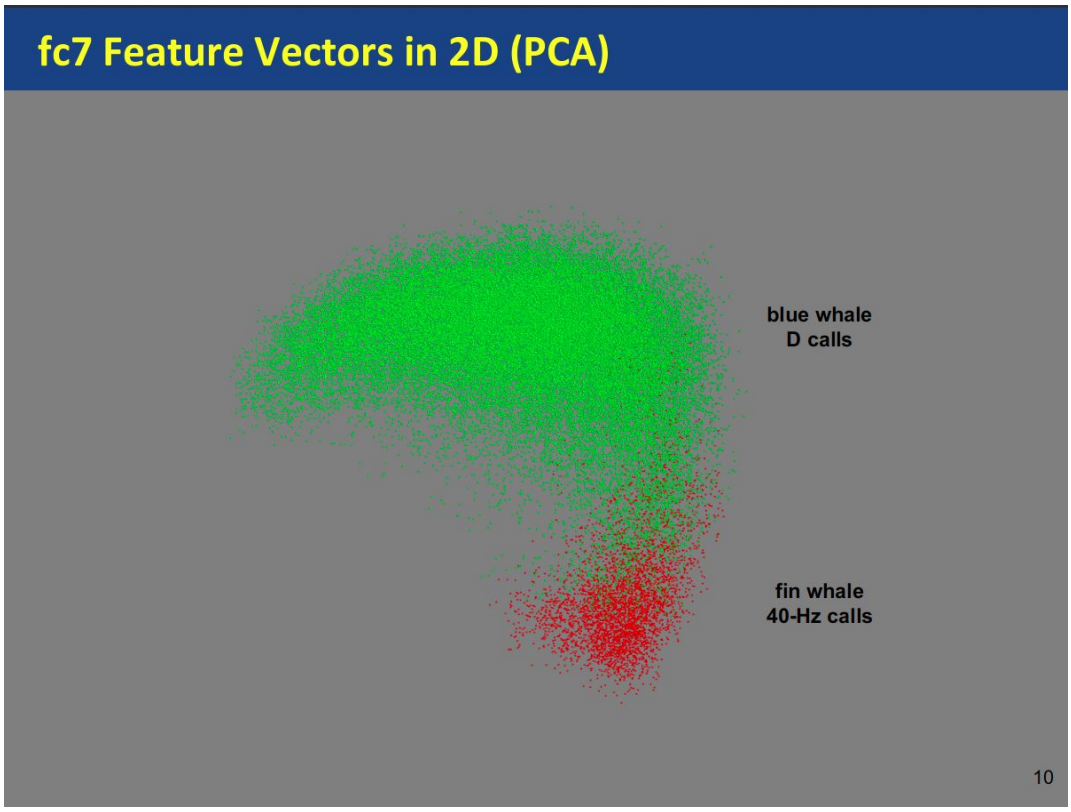
Only the first dense layer was computed, in this network, we will use AlexNet since there is a previous work done by Karnowski [42] in a similar problem that yielded outstanding results. Karnowski used Alexnet and took the values of the 2nd dense layer yield by the spectrograms. He worked with a similar problem, differentiating two different whale call, from Blue Whale and Fin Whale. He obtained an Accuracy of 0.9760 on his dataset.

For this problem the results were very different, after computing the last layer the resulting features were saved. Then several SVM were trained to optimize the parameters. After several attempts the results were far from what was expected, the accuracy was around 0.76, which, having only 24.3% of right whale calls it means it classified almost all samples as no whales. This was shown in the confusion matrix, where almost all the whales were classified as not the Right Whales.

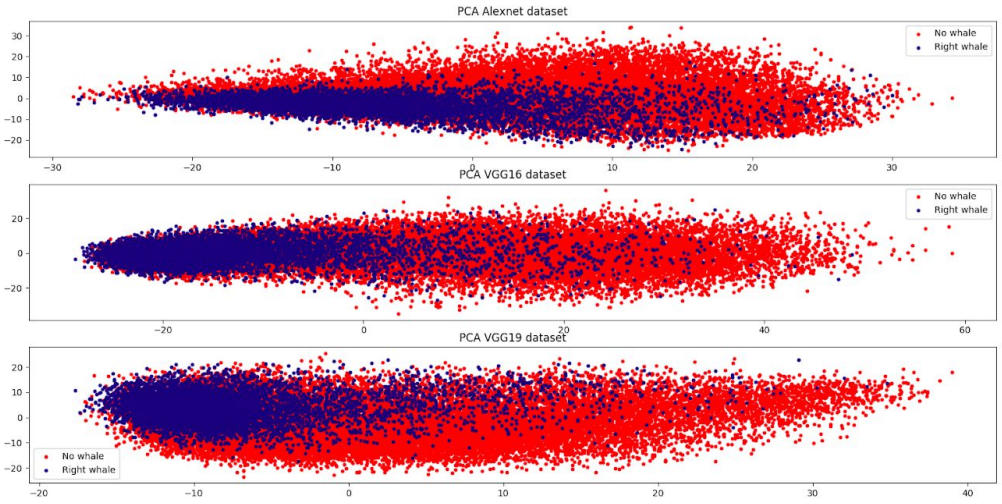
	Actual No Right Whale	Actual Right Whale
Predicted No Right Whale	4143	1043
Predicted Right Whale	381	433

Karnowski did the experiment back in 2015, several other CNN have been able to outperform Alexnet for image classification. So some of these pretrained networks were also used to try to obtain descriptive features from the last dense layers, Nevertheless, the results were similar, both VGG16 and VGG19 (Networks presented in the article Very Deep Convolutional Networks for Large-Scale Image Recognition[43], VGG stands for Oxford's Visual Geometry Group, team that submitted the work and the number corresponds to the number of layers) failed to classify right whales as such.

Karnowski proved results of his work by showing the PCA representation and separation of data for the two top components of the feature vector. As follows:



For this problem the representation wasn't as good as his, the three networks yielded a very mixed representation, there is no simple way of separating both classes. The top components of a PCA of the features for our data yielded the following:



Which in part explains the results from the SVM experiments. In the two top components of the results shows how close some of the Right Whale calls are from the others, making it difficult to classify the correct ones, ending in very simple hyperplanes that just classify everything as whales which minimizes the error. Knowing this, several measures could be taken to ensure that at least some of the Right Whales are classified correctly. This could be of help to improve the model, however the results are not promising enough to try to use much time optimizing the SVM. It is easier and faster to just optimize preprocessing and CNN architecture and hyperparameters.

4 - Conclusions

4.1 - Project Summary

After several experiments two models were refined for this classification problem. The two networks were able to outperform each other in 2 different scenarios. A deep CNN following AlexNet architecture for spectrogram images and a simpler network for the matrices of the spectrogram. Proving the effectiveness of AlexNet architecture once more in the Object Recognition field and CNN in general for complex data. Both approaches for the problem were valid but each one had a different model performance depending on the scenario, which was interesting. The data was always the same but the results varied when treating spectrograms as images or as matrices of sparse data. The two models presented in this work are the following:

- The first model consisted of three Convolutional Layers, each followed by a Max Pooling layer, two Dense Layers and the output layer. For smaller images, the model was adapted reducing the Max Pooling size, the kernel size and the number of different convolutions to half the original.
- The second model was composed of two Convolutional Layers with dropout, followed each by a Max Pooling layer, one Dense Layer with Dropout and the final output Neurons. This model has less Convolutions and the kernel size was the same for both layers.

Along with these architectures several ways of preprocessing the spectrograms were presented:

- Scaling the images, from the original 631x398 to the fixed sizes of 224x224, 128x128 and 32x32.
- Cropping the images to extract the frequencies between which the Up-Calls can be found.
- Different window sizes and Window Functions.
- Converting the images to one channel gray scale images.
- Data augmentation by shifting and adding noise.
- Spectrograms kept in memory, instead of treating them as images and saving them in disk.
- Using Redux Dataset.

Preprocess	Best Model	Best Model's accuracy	Best model's AUC
Scaling	First CNN for 32x32	0.8956	0.7336
Crop	First CNN for 32x32	0.8660	0.7554
Gray Images	First CNN for 32x32	0.9117	0.8637
Data augmentation	First CNN for 32x32 with Dropout	0.9240	0.8996
Spectrograms in memory	Second CNN with Logarithmic Normalization	0.9213	0.9724
Redux Data	Second CNN with Logarithmic Normalization	0.9666	0.9916

Each of the preprocesses yielded different results for each of the models. With each step, the dimensions of the spectrograms were reduced which permitted a faster training of the models. Cropping the data to ignore higher frequencies boosted accuracy whereas data augmentation boosted AUC. From the original result of accuracy 0.79 to the best results with the first model with accuracy 0.911 and AUC 0.866 outperforming the others models. The second model was able to beat the first one when working with spectrograms in memory, with an accuracy of 0.92 and an AUC of 0.98. These results were not only obtained in Kagle's original dataset, but also in the Redux dataset.

4.2 - Future Work

Although the results were good, these results were obtained in the Train set of the competition. Unfortunately the competition closed the submissions, making it impossible to test these models in the original competition for comparison.

As we have seen, audio data can be interpreted and work with as images, using the spectrogram transformation. This approach is the one most used in the literature, but that does not mean it is the best approach. There are lots of drawbacks when considering spectrograms as images. During this project many of these issues were faced but other experiments could be added:

- The calls are assumed to be in the middle of the image, with a very short duration, although this is not always true.
- After the spectrograms are cut the highest frequencies get ignored which is useful for this problem but other whale calls can have different frequencies.
- Original spectrograms are much heavier than the reduced ones, however this only affects the training time and the size of the model which is something that in many situations can be taken on for a better generalization or problem transference. Spectrograms before cropping took an average of 700 seconds per epoch, reduced to 311 seconds when dimensions were reduced and 150 seconds after all preporcesses. Size of the dataset in memory varied from 4.93GB to 2.11GB.
- The model is able to easily separate whale calls from random noise, but fails from other whale calls or similar noises, as shown in Section 3.3.3. The dataset could be improved by adding more of these similar noises.
- As shown by the performance of both networks the models are very software dependant. The way the spectrograms are calculated, normalized and audio data is sampled, can affect greatly the classification. If data is taken from another kind of hardware with a different set up results could vary. This was shown by the two Dataset by Kaggle in Section 3.3.2.
- Audio data tends not to be very consistent. Environmental noise and other sources of noise can make the classification fail, forcing to have more examples for these cases.

Although feature extraction technique didn't perform as well as directly applying a classifier, there is plenty of room for improvement, such as using other algorithms: Convolutional Autoencoders [8] or Siamese Networks[41]. Or any other pretrained ANN for Transference Learning [39].

Even Though CNN have proved to work well for audio data in this project and others, [12][24][33][42], vanilla CNNs can not be the optimal way of working with this kind of data as mentioned.

There is also an inherent problem in the way the spectrograms are treated. Discrete sound events do not separate into layers on a spectrogram. Instead, they all sum together into a distinct whole. That means that a particular observed frequency in a spectrogram cannot be assumed to belong to a single sound as the magnitude of that frequency could have been produced by any number of accumulated sounds or even by

the complex interactions between sound waves such as phase cancellation, when 2 different sound waves with the same frequency are out of phase cancel each other. This makes it difficult to separate simultaneous sounds in spectrogram representations.

The axes on the spectrogram carry different meanings, one is time and the other is frequency. In images, neighbouring pixels can be assumed to belong to the same object, in a spectrogram pixels are not locally grouped but moved together following the same relationship.

Sound is serial, in a spectrogram each value comes after the other and sound can be traced temporarily. CNNs ignore this property and try to match certain structures inside the image ignoring time and frequencies. There are other NN architecture that could be explored with a better understanding of the temporal relation. Google AI introduced a deep generative neural network that can work directly with sound waves [40].

5 - Bibliography

1. “Blue Whale - Balaenoptera Musculus.” *Blue Whale Facts and Pictures - Balaenoptera Musculus*, www.coolantarctica.com/Antarctica fact file/wildlife/whales/blue_whale.php.

2. “Are Blue Whales Finding New ‘Microphone Channel’ to Communicate in?” *Life at OSU*, 5 Oct. 2017, today.oregonstate.edu/archives/2017/aug/scientists-blue-whale-calls-lowering-frequency---and-they-may-be-controlling-it.

3. Bogucki, Robert, et al. “Applying Deep Learning to Right Whale Photo Identification.” *Conservation Biology*, vol. 33, no. 3, 2018, pp. 676–684., doi:10.1111/cobi.13226.

4. Bouffaut, Léa, et al. “Passive Stochastic Matched Filter for Antarctic Blue Whale Call Detection.” *The Journal of the Acoustical Society of America*, vol. 144, no. 2, 2018, pp. 955–965., doi:10.1121/1.5050520.

5. Chami, Ralph. “Nature's Solution to Climate Change – IMF F&D.” *Nature's Solution to Climate Change – IMF F&D*, www.imf.org/external/pubs/ft/fandd/2019/12/natures-solution-to-climate-change-chami.htm.

6. “Does Military Sonar Kill Marine Wildlife?” *Scientific American*, 10 June 2009, www.scientificamerican.com/article/does-military-sonar-kill/.

7. Freitas, Guilherme Kamizake De, et al. “Using Spectrogram to Detect North Atlantic Right Whale Calls from Audio Recordings.” *2016 35th International Conference of the*

Chilean Computer Science Society (SCCC), 2016, doi:10.1109/sccc.2016.7836034.

8. Geng, Jie, et al. “High-Resolution SAR Image Classification via Deep Convolutional Autoencoders.” *IEEE Geoscience and Remote Sensing Letters*, vol. 12, no. 11, 2015, pp. 2351–2355., doi:10.1109/lgrs.2015.2478256.

9. Hochreiter, Sepp, and Schmidhuber Jürgen. *Long Short Term Memory*. Inst. für Informatik, 1995.

10. “Home.” *Discovery of Sound in the Sea*, 14 May 2019, dosits.org/.

11. “The ICML 2013 Whale Challenge - Right Whale Redux.” *Kaggle*, www.kaggle.com/c/the-icml-2013-whale-challenge-right-whale-redux.

12. Kabani, A., and M. R. El-Sakka. “North Atlantic Right Whale Localization and Recognition Using Very Deep and Leaky Neural Network.” *Mathematics for Application*, vol. 5, no. 2, 2017, pp. 155–170., doi:10.13164/ma.2016.11.

13. Konovalov, Dmitry A., et al. “Individual Minke Whale Recognition Using Deep Learning Convolutional Neural Networks.” *Journal of Geoscience and Environment Protection*, vol. 06, no. 05, 2018, pp. 25–36., doi:10.4236/gep.2018.65003.

14. Krizhevsky, Alex, et al. “ImageNet Classification with Deep Convolutional Neural Networks.” *Communications of the ACM*, vol. 60, no. 6, 2017, pp. 84–90., doi:10.1145/3065386.

15. Lecun, Y., et al. “Gradient-Based Learning Applied to Document Recognition.” *Proceedings of the IEEE*, vol. 86, no. 11, 1998, pp. 2278–2324., doi:10.1109/5.726791.

16. “The Marinexplore and Cornell University Whale Detection Challenge.” *Kaggle*,

www.kaggle.com/c/whale-detection-challenge/overview.

17. Minsky, Marvin, and Seymour Papert. *Perceptrons: an Introduction to Computational Geometry*. The MIT Press, 2017.

18. Minsky, Marvin Lee., and Seymour Popeit. *Perceptrons*. MIT Pr.), 1969.

19. Noaa. “2017-2019 North Atlantic Right Whale Unusual Mortality Event.” *NOAA Fisheries*, 4 Oct. 2019, www.fisheries.noaa.gov/national/marine-life-distress/2017-2019-north-atlantic-right-whale-unusual-mortality-event.

20. Noaa. “2019 Gray Whale Unusual Mortality Event along the West Coast.” *NOAA Fisheries*, 4 Oct. 2019, www.fisheries.noaa.gov/national/marine-life-distress/2019-gray-whale-unusual-mortality-event-along-west-coast.

21. Noaa. “North Pacific Right Whale (Eubalaena Japonica) Five-Year Review, 2017.” *NOAA Fisheries*, 18 June 2018, www.fisheries.noaa.gov/resource/document/north-pacific-right-whale-eubalaena-japonica-five-year-review-2017.

22. Pearson, Heidi. “Sea Creatures Store Carbon in the Ocean – Could Protecting Them Help Slow Climate Change?” *The Conversation*, 9 Aug. 2019, http://theconversation.com/sea-creatures-store-carbon-in-the-ocean-could-protecting-them-help-slow-climate-change-108872.

23. Phan, Huy, et al. “Robust Audio Event Recognition with 1-Max Pooling Convolutional

Neural Networks.” *Interspeech 2016*, 2016, doi:10.21437/interspeech.2016-123.

24. Piczak, Karol J. “Environmental Sound Classification with Convolutional Neural Networks.” *2015 IEEE 25th International Workshop on Machine Learning for Signal Processing (MLSP)*, 2015, doi:10.1109/mlsp.2015.7324337.

25. Potter, John R., et al. “Marine Mammal Call Discrimination Using Artificial Neural Networks.” *The Journal of the Acoustical Society of America*, vol. 96, no. 3, 1994, pp. 1255–1262., doi:10.1121/1.410274.

26. Rui, Gao, et al. “Automatic Template Matching for Classification of Dolphin Vocalizations.” *OCEANS 2008 - MTS/IEEE Kobe Techno-Ocean*, 2008, doi:10.1109/oceanskobe.2008.4530981.

27. Smirnov, Evgeny. "North Atlantic right whale call detection with convolutional neural networks." Proc. Int. Conf. on Machine Learning, Atlanta, USA. 2013.

28. Sola, J., and J. Sevilla. “Importance of Input Data Normalization for the Application of Neural Networks to Complex Industrial Problems.” *IEEE Transactions on Nuclear Science*, vol. 44, no. 3, 1997, pp. 1464–1468., doi:10.1109/23.589532.

29. “TensorFlow.” *TensorFlow*, www.tensorflow.org/.

30. US Department of Commerce, and National Oceanic and Atmospheric Administration. “How Far Does Sound Travel in the Ocean?” *NOAA's National Ocean Service*, 23 Sept. 2014, oceanservice.noaa.gov/facts/sound.html.

31. “Understanding Ocean Acoustics.” *NOAA Ocean Explorer Podcast RSS*, oceanexplorer.noaa.gov/explorations/sound01/background/acoustics/acoustics.html.

32. Urazghildiiev, Ildar R., and Christopher W. Clark. “Acoustic Detection of North Atlantic Right Whale Contact Calls Using the Generalized Likelihood Ratio Test.” *The Journal of the Acoustical Society of America*, vol. 120, no. 4, 2006, pp. 1956–1963., doi:10.1121/1.2257385.

33. Xu, Kele, et al. “North Atlantic Right Whale Call Detection with Very Deep Convolutional Neural Networks.” *The Journal of the Acoustical Society of America*, vol. 141, no. 5, 2017, pp. 3944–3945., doi:10.1121/1.4988946.

34. “Your Home for Data Science.” *Kaggle*, www.kaggle.com/.

35. Bengio, Yoshua, et al. *Deep Learning*. MIT Press, 2017.

36. LeCun, Y. (1989). “Generalization and network design strategies”. Technical Report CRG-TR-89-4, University of Toronto. 331 , 351

37. Torrey, Lisa and Jude Shavlik. "Transfer Learning." *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques.* IGI Global, 2010. 242-264. Web. 6 Oct. 2019. doi:10.4018/978-1-60566-766-9.ch011

38. Heuritech. “Heuritech/Convnets-Keras.” *GitHub*, 13 June 2017, github.com/heuritech/convnets-keras.

39. Lilun Zhang, Dezhi Wang, Changchun Bao, Yongxian Wang, Kele Xu. “Large-Scale Whale-Call Classification by Transfer Learning on Multi-Scale Waveforms and Time-Frequency Features”. Applied Sciences - 03 / 2019

40. Oord, Aaron van den, et al. "Wavenet: A generative model for raw audio." *arXiv preprint arXiv:1609.03499* (2016).

41. Shon, Suwon, Ahmed Ali, and James Glass. "Convolutional neural networks and language embeddings for end-to-end dialect recognition." arXiv preprint arXiv:1803.04567 (2018).

42. Jeremy Karnowski and Yair Movshovitz-Attias. “Classification of blue whale D calls and fin whale 40-Hz calls using deep learning”. SCRIPPS Whale acoustic lab. The 7th International DCLDE [Detection, Classification, Localization, and Density Estimation] Workshop 2015. http://www.cetus.ucsd.edu/dclde/docs/pdfs/Monday/14-Karnowski.pdf

43. Simonyan, Karen & Zisserman, Andrew. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. arXiv 1409.1556 (2014).

44. Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, Ruslan Salakhutdinov. “Dropout: A Simple Way to Prevent Neural Networks from Overfitting” Journal of Machine Learning Research. 15. 1929-1958 (2014).