Dyni Odontocete Click Classification 2020 Challenge (DOCC10)

Mathieu Orhan
Master MVA, ENS Cachan
mathieu.orhan@eleves.enpc.fr

Bastien Dechamps
Master MVA, ENS Cachan
bastien.dechamps@eleves.enpc.fr

Abstract

The aim of this project was to try different prediction models to solve the DOCC10 Challenge[1] which consists in classifying *clicks* emitted by biosonar of 10 cetaceans species (odontocetes). In particular, we test the limits of a linear model trained on extracted features compared with more powerful models based on convolutional neural networks.

1 Introduction

The odontocetes parvorder regroups cetaceans with teeth – commonly called Teeth Whales, which includes for example dolphins or porpoises. Odontocetes such as the killer whale are super predators of the ocean, thus they largely affect the sea's population dynamic. Keeping a census is therefore useful to monitor these species and broadly the oceans' marine life. One way to track these species is to record underwater sounds and identify the echolocation signal [6, 11], a transient called a *click*. To navigate and hunt underwater, odontocetes do not rely on sight but rather on a biological sonar. These sonars emit trains of clicks at various frequency ranges. Using hydrophones, they can be recorded, and the resulting signal might be used to classify the species. See [17] for a recent review of the echolocation mechanism, which is not fully understood.

A trained analyst can aurally distinguish these signals, but treating large-scale data is difficult and costly. Automatic click classification has a literature. Previously, machine learning classifier such as Support Vector Machine were used on top of aggregated and carefully crafted handmade features, using expert knowledge and signal processing tools [21, 14, 15, 18]. Recently, [4] uses deep neural network approach with a CWT on a denoised signal along with a temporal feature vector.

The Dyni Odontocete Click Classification Challenge provides more than 100000 labelled samples of such signals for 10 species of odontocetes. The recordings are collected at various locations, in different conditions, and most of them are very noisy. Indeed, underwater sounds are generated by a great variety sources, such as waves, rain, marine life, or shipping [20]. Fortunately, the energy of this ambient sound is largely concentrated at low frequencies, below 1000 Hz. The clicks themselves are noisy, and vary both between individuals and for a given individual. [17] For instance, Sperm whales can produce multiple clicks, whether they communicate, navigate, or detect prey.

In this work, we tried to put as much prior information to build a low dimensional representation of the signal to train a linear classifier and maximize the accuracy. We also provide a deep learning model on time-frequency features that outperform the baseline and the linear model by a large margin.

[1] Dyni Odontocete Click Classification, 10 species, organized by the Scaled Acoustic BIODiversity platform (SABIOD).

2 Feature Engineering

We first try to extract the most relevant features to tackle this classification problem, as we cannot just feed directly the signal values into a classifier. Indeed, the data contains a lot of *a priori* information that can be used to form meaningful features. In the following, we denote by x the N-dimensional signal with $N = 8192$.

2.1 Description and general features

The *clicks* are already preprocessed, all being composed of 8192 values sampled at 200 kHz. There is a total of 11312 samples per class in the train dataset and 2096 for the test dataset, thus it is a balanced classification problem. Some of the samples are shown in figure 1.

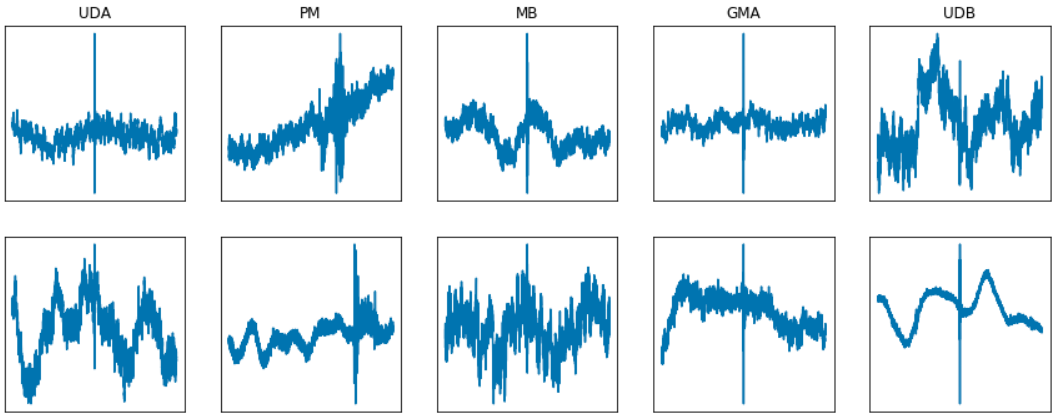


Figure 1: Raw data for 5 different classes (UDA, PM, MB, GMA, UDB)

We first extracted very general features directly from the raw signals: the maximum and minimum value, the mean μ and the standard deviation σ. We also computed for each sample the *skewness* and the *kurtosis*, which are respectively defined by

$$\tilde{\mu}_3(x) = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{x[n]-\mu}{\sigma}\right)^3 \quad , \quad \beta_2 = \frac{1}{N}\sum_{n=1}^{N}\left(\frac{x[n]-\mu}{\sigma}\right)^4 \quad .$$

However, those 6 features are clearly not sufficient as the *clicks* are very localized in time and the samples also contain noise of lower frequencies that could correspond to ambient sounds from the sea for example.

2.2 Click localization and segmentation

The ambient sound can be used to identify the species, but we wanted to rely only on *clicks sounds*. There are certainly features in the background noise that might correlate with a given species. It can be sounds relative to the environment or other sounds produced by the species. We tried to avoid the pitfall of learning from these unknown features in our linear model as they could lead to poor generalization given the ambient noise. In particular, it is known [17] that for some odontocete species, whistles accompany the echolocation clicks. These whistles are continuous narrowband FM signals that typically last between 100 and 200 milliseconds and have several harmonic in the 5kHz - 20 kHz frequency range [8].

However, the *click* only represent a small part of the entire signal, so we want to isolate it. All the samples were claimed to be centered on the *clicks*, but we found out that it was not the case for the class `'PM'` (see figure 1 for example). Thus, we implemented a method to find the *click* on a given signal and to crop the latter on a small window around it. An example of click detection is represented on figure 2.

To perform this, we first apply a Butterworth high-pass filter on the signal with a cutoff frequency of 10 kHz to get rid of the low frequency noise. Then we smooth the signal using a Wiener filter (with a

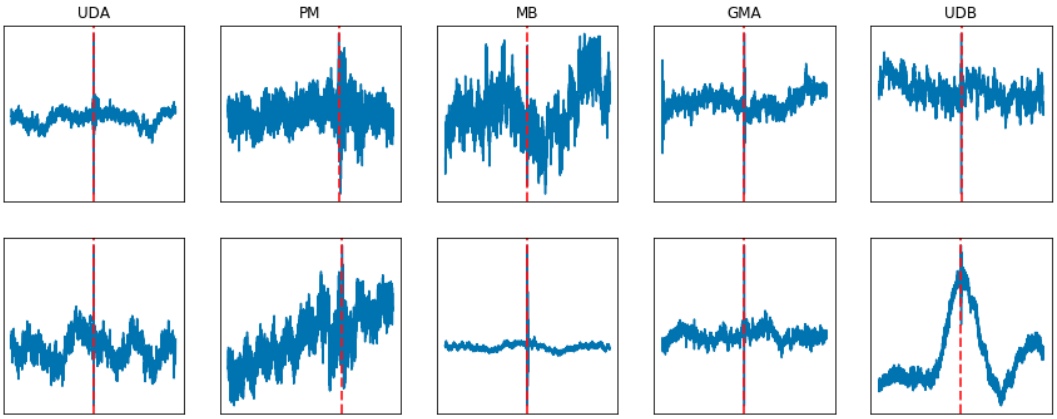


Figure 2: Click detection on raw signal (UDA, PM, MB, GMA, UDB)

window of size 50) followed by a gaussian 1D convolution ($\sigma = 1.0$). Finally, we can assume that the signal has the maximum amplitude during the click and we extract the corresponding time. The signal is then cut in a window around the click of size $w = 2^p$.

We will now denote by $x_{c,p}$ the signal segmented around the click estimate c with a width of size $w = 2^p$.

2.3 Spectral features

Clicks are largely characterized by spectral features, and even though the energy distribution can greatly vary for a given label, some species are easy to separate from other. For instance, the Sperm Whale (PM class) emits clicks from 0.2 to 32 kHz, but most of the energy is between 3 and 15 kHz [9]. All the other species emit clicks of higher frequencies, therefore this class is very well characterized by its power spectrum. The clicks of the Sowerby's beaked whales (MB class) have a median frequency peak of 33 kHz for most of the sample of [3]. However, some classes are hard to separate and form very similar patterns.

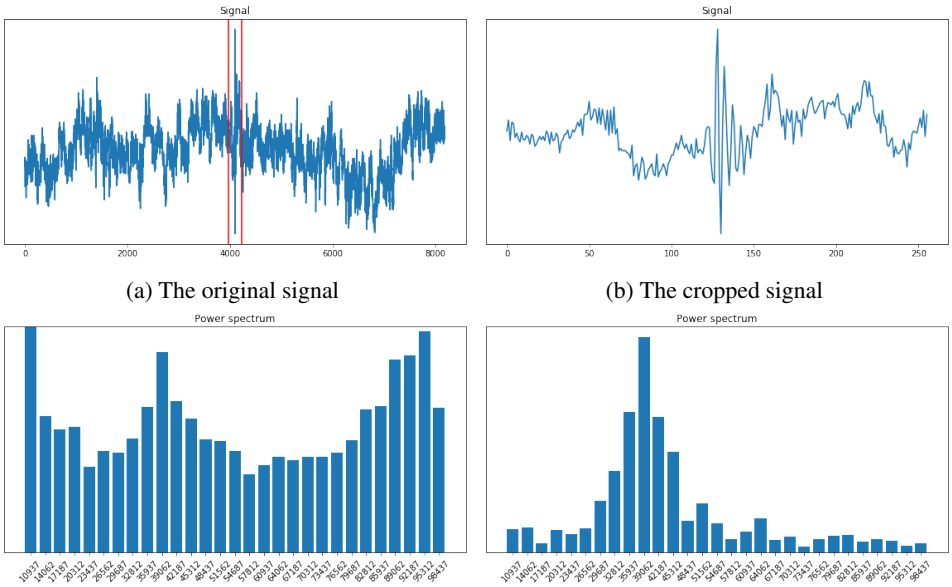
First we estimated the power spectrum using the Welch's method [19], a popular choice to reduce the noise, with a Gaussian window with $\sigma_w = 30$ and of size 64, segments of size 256, and a FFT length of 64. The use of a Gaussian window helps to smooth the power spectrum. We apply the method on $x_{c,8}$ instead of x to remove most of the noise energy. We finally remove frequencies below 10000 Hz that are very noisy. We get a feature of size 29. Figure 3 compares the signal and the features with and without cropping. Figure 4 shows the features for all classes on several random samples. Figure 5 show the distribution of the feature on the training set. Some classes are easy to identify, e.g. PM (only frequencies below 15 kHz), MB (mostly high frequencies, regular single mode). Some are harder to differentiate, e.g. UDA and LA.

Then, we computed the FFT of the centered signals $x_{c,8}$ and found the 2 frequencies with the highest amplitude (after having smoothed the spectrum). We added to the current features the values of those two frequencies as well as their amplitudes. We also compute the spectral width at half power.

2.4 Time-frequency features

We tried to characterize the signal by using time-frequency or time-scale features provided by a Short Time Fourier Transform (STFT), a mel-spectrogram or a Wavelet Transform. We found the Continuous Wavelet Transform (CWT) power spectrum to provide a promising representation of the signal.

We choose the Complex Morlet wavelet which is closely related to human hearing perception [10]. It also has a good compromise between compacity and smoothness in both time and frequency domain. We compute scales corresponding to the periods 1 to 20, and keep only these scales. By using $x_{c,7}$ as input, we get a *time-scale signature* of the click, isolated from the background noise. We use the `PyWavelets` Python library [7]. Figure 7 displays such scaleograms for all classes.



(a) The original signal

(b) The cropped signal

(c) Power spectrum of the original signal

(d) Power spectrum of the cropped signal

Figure 3: Power spectrum features for a sample (UDA)

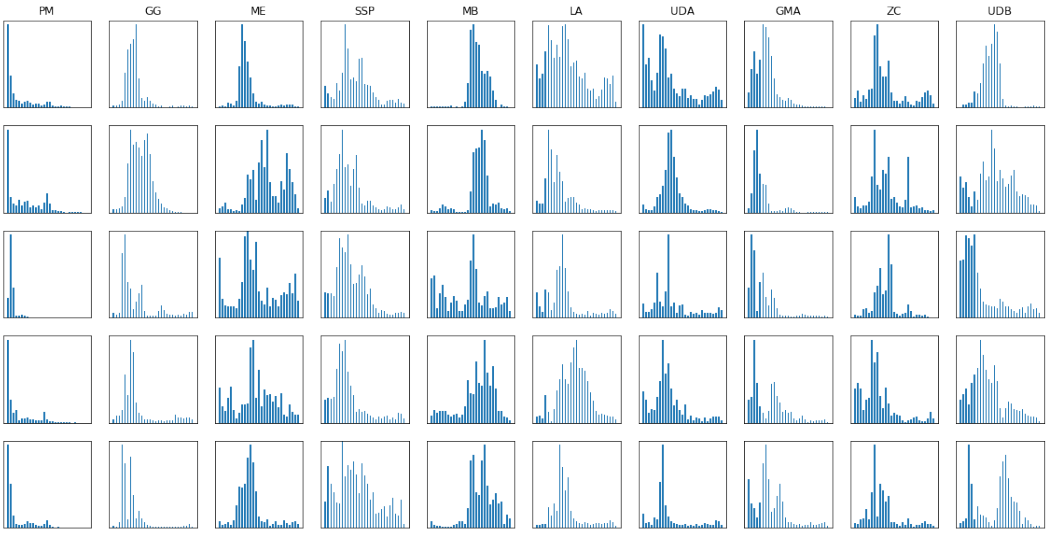
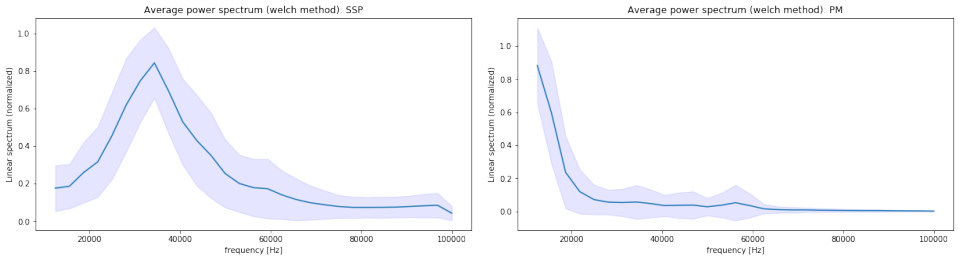


Figure 4: Power spectrum features for random sample of all classes, normalized

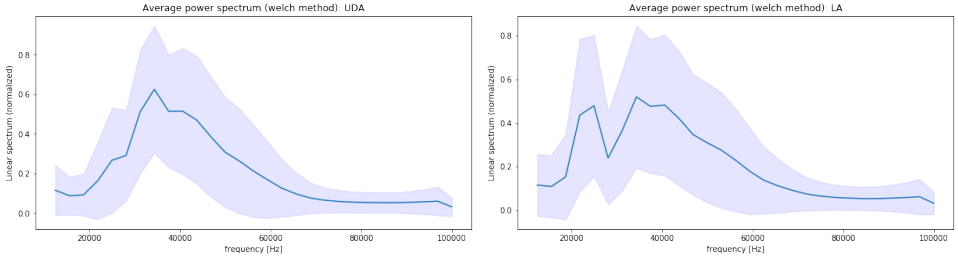
Interestingly, some classes hard to differentiate with the power spectrum have now very regular patterns. The ME blobs are slightly rotated, the UDA's blobs are vertical and have a very regular shape.

Extracting low dimensional features is not obvious. One idea to characterize scaleograms is to extract the center and shape of the main mode. To do so, we computed the histogram on the scale and the time axis. We then find the largest local maximum, and compute its width at 75% of its height. After being normalized, the center and the extracted shape can be used as an additional feature vector. Figure 6 displays the extraction for a sample signal.

This feature is not very informative, and does not characterize well the distribution, for instance it does not capture rotation. To construct better features, we think that fitting a parametric function e.g.



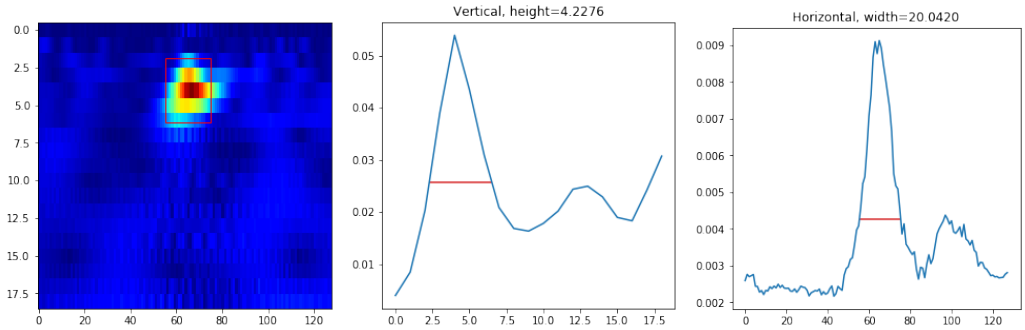
(a) SSP (b) PM



(c) UDA (d) LA

Figure 5: Power spectrum mean for 4 classes (std is highlighted)

a two dimensional Gaussian and use their parameters as a feature vector might produce much richer features.



(a) Scaleogram (b) Scale histogram (c) Time histogram

Figure 6: Scaleogram feature extraction.

2.5 Logistic Regression

As one of the goals of this project was to try to get the best results with a simple linear model, so we chose to train a logistic regression with l_2 normalization using the `scikit-learn` Python library [13]. The input data was composed of 40 different concatenated features $\phi(x)$ including simple statistics and the previous spectral features. The dataset is randomly split in a train set and a validation set so that they are kept balanced. Using a grid search we found an optimal regularization parameter $C = 50$ for the logistic regression.

The resulting accuracies on the different datasets are given in the table 1. We also report the confusion matrix on the validation set figure 8. While some classes are mostly solved (PM, MB), there are still important sources of confusion between e.g. UDA and GG.

We notice a large gap between the validation accuracy and the test accuracy [2] that we will also retrieve on the convolutional neural network models. We first thought that this gap was due to the fact that

[2] given by submitting the predictions on the challenge website www.challengedata.ens.fr

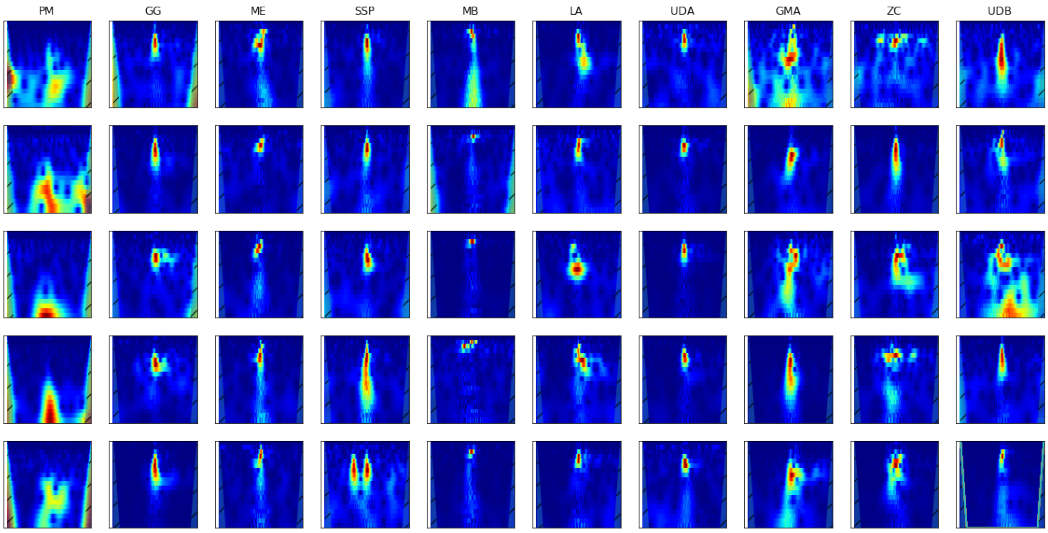


Figure 7: Scaleograms (amplitude) for random samples of all classes, normalized. The cone of influence is highlighted. The highest values are red.

Setting	Train	Valid.	Test (LB)
S	0.3135	0.3124	–
PS	0.6379	0.6327	–
F	0.3807	0.3798	–
W	0.3775	0.3798	–
PS+F	0.6562	0.6593	–
S+PS+F	0.7009	0.6983	–
S+F	0.5367	0.5333	–
S+PS+W+F	0.7031	**0.7049**	0.4504

Table 1: Accuracies for the logistic regression. S: simple features, PS: power spectrum features, W: scaleogram features, F: spectral width and modes.

test *clicks* were not centered. But as we managed to center the samples around their *click*, it seems to come from another reason. We tried changing the global normalization to a sample normalization but it gave the same discrepancy.

3 Convolutional Neural Networks

After having optimized a simple logistic regression over hand-crafted features, we tried another approach using deep neural network architectures such as those presented in [2]. We considered the scaleograms presented on section 2.4 but also mel-spectrograms as 2D inputs for the network.

3.1 Mel-spectrograms

As we have at disposal a lot of samples, a widely-used approach to classify sounds is to train convolutional neural networks on mel-spectrograms. The mel scale [16] correspond a logarithmic scale that match the human perception of sound. These perceptual features make sense, as [1] points out, human can aurally distinct the species. We built those spectrograms from the original signals using the Python library `Librosa` [12]. As the clicks are generally composed of high frequencies, we chose the [10 kHz – 100 kHz] as the range of frequencies, and we used a filterbank of 64 filters to compress the spectrograms and convert the frequencies onto the mel-scale. We use a hop length of 64 and a FFT length of 256. Finally, we normalize them using sample normalization.

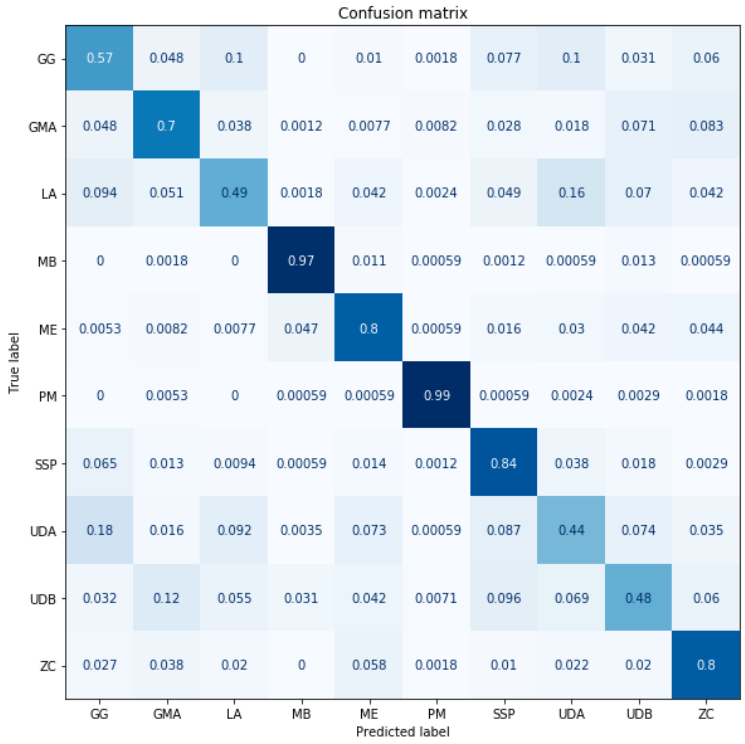


Figure 8: Confusion matrix (best performance of the logistic regression)

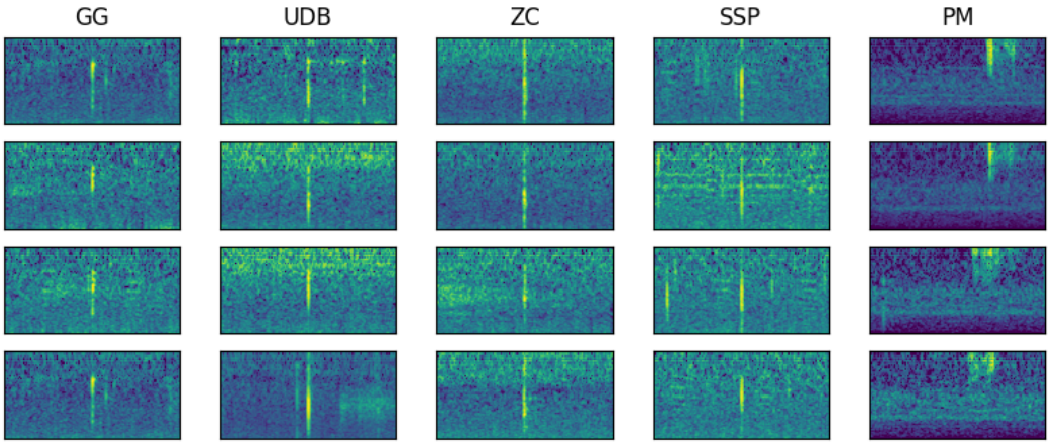


Figure 9: Mel-spectrograms for random samples taken from 5 classes.

On figure 9 are represented some mel-spectrograms from random classes. Once again, the added class PM has a more noticeable pattern than the other classes.

3.2 Model

We used a 2D convolutional network with skip connections for the classification task, represented on figure 10

where each block of convolution is defined by the following architecture:

```
Model(
  (conv1): Sequential(
    (0): Conv2d(in1, out1, kernel\_size=3, stride=(1, 1), padding=(1, 1))
```

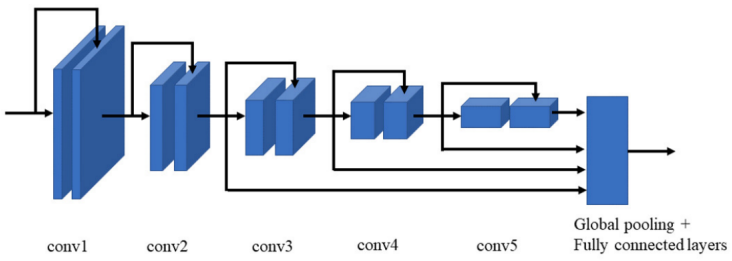


Figure 10: Deep architecture used for classification.

```
    (1): BatchNorm2d(out2, eps=1e-05, momentum=0.1, affine=True)
    (2): Linear(in_features=512, out_features=256, bias=True)
    (3): ReLU()
  )
  (conv2): Sequential(
    (0): Conv2d(in2, out2, kernel_size=3, stride=(1, 1), padding=(1, 1))
    (1): BatchNorm2d(out2, eps=1e-05, momentum=0.1, affine=True)
    (2): ReLU()
  )
)
```

One can also use pretrained models such as Resnet or Inception architectures. We trained the whole network using a batch size of 32, the Adam optimizer [5] with a learning rate of $\eta = 0.0005$, during 10 epochs.

3.3 Results

We trained the network using mel-spectrograms as well as scaleograms. The use of scaleogram in combination with neural networks was introduced by [4]. The results on the different datasets are given in table 2.

	Train	Valid.	Test (LB)
logreg	0.7031	0.7049	0.4504
scaleo	0.9173	0.9025	0.7715
melspec	0.9317	**0.9208**	**0.7953**

Table 2: Accuracies for the CNN.

We obtain a better accuracy with mel-spectrograms than with scaleograms, after having tuned the hyperparameters for both methods. Once again, there is a gap between validation and test scores, that is not explained by over-fitting issues nor centering issues. Furthermore, the CNN model clearly outperforms the logistic regression built on the previous features. Indeed, although the features it creates are less interpretable, it brings non-linearity to the problem and performs very well when dealing with large amount of data.

4 Conclusion and limits

The journey to construct click features has been very instructive and is certainly not over. We extracted temporal, spectral, and time-frequencies features using a variety of signal processing tools and prior knowledge. This exercise is also yet another demonstration of the power of feature representation learning of deep neural networks. Using a raw time-frequency representation, a convolutional neural network is able to learn features and outperform our linear model by a large margin. We also comfortably beat the challenge baseline on the leaderboard, and rank first in academic ranking at the time of submission.

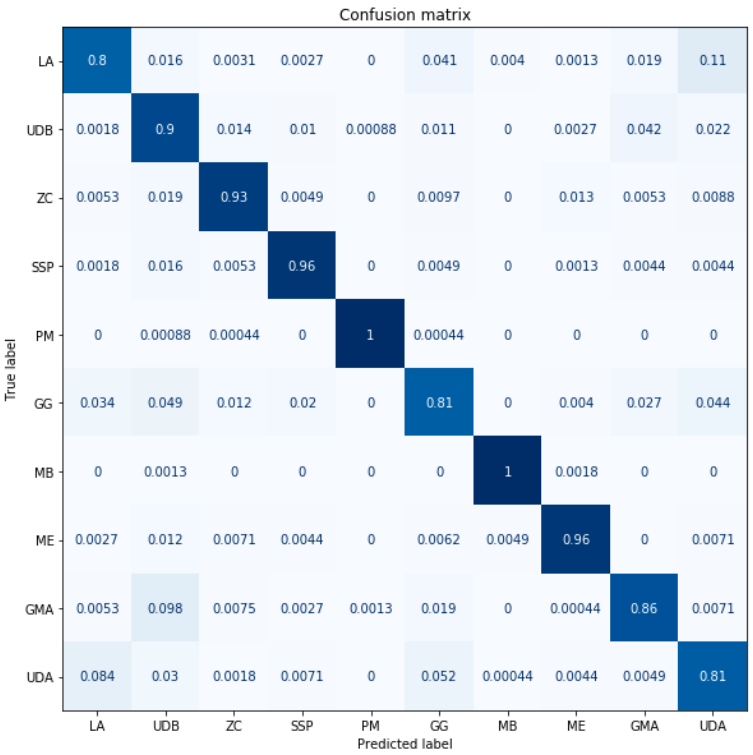


Figure 11: Confusion matrix for the CNN model trained with mel-spectrograms.

However, we miss domain expert knowledge to design better features. A complete literature review on the subject should provide hints to design filters or design features. It was also our first experience with wavelets. We think that with more mastery of them we might push further performances on both the linear and the deep models.

References

[1] Carolyn M Binder and Paul C Hines. Automated aural classification used for inter-species discrimination of cetaceans. *The Journal of the Acoustical Society of America*, 135(4):2113–2125, 2014.

[2] Keunwoo Choi, George Fazekas, and Mark Sandler. Automatic tagging using deep convolutional neural networks, 2016.

[3] Danielle Cholewiak, Simone Baumann-Pickering, and Sofie Van Parijs. Description of sounds associated with sowerby's beaked whales (mesoplodon bidens) in the western north atlantic ocean. *The Journal of the Acoustical Society of America*, 134(5):3905–3912, 2013.

[4] Jia-jia Jiang, Ling-ran Bu, Xian-quan Wang, Chun-yue Li, Zhong-bo Sun, Han Yan, Bo Hua, Fa-jie Duan, and Jian Yang. Clicks classification of sperm whale and long-finned pilot whale based on continuous wavelet transform and artificial neural network. *Applied Acoustics*, 141:26–34, 2018.

[5] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[6] Elizabeth T Küsel, David K Mellinger, Len Thomas, Tiago A Marques, David Moretti, and Jessica Ward. Cetacean population density estimation from single fixed sensors using passive acoustics. *The Journal of the Acoustical Society of America*, 129(6):3610–3622, 2011.

[7] Gregory Lee, Ralf Gommers, Filip Waselewski, Kai Wohlfahrt, and Aaron O'Leary. Pywavelets: A python package for wavelet analysis. *Journal of Open Source Software*, 4(36):1237, 2019.

[8] Zhang Liang, Guo Longxiang, and Mei Jidan. Analysis and identification of cetacean sounds based on time-frequency analysis. *Procedia Engineering*, 29:2922–2926, 2012.

[9] Peter Madsen, Magnus Wahlberg, and Boris Møhl. Male sperm whale (physeter macrocephalus) acoustics in a high-latitude habitat: implications for echolocation and communication. *Behavioral Ecology and Sociobiology*, 53(1):31–41, 2002.

[10] Stéphane Mallat. *A wavelet tour of signal processing*. Elsevier, 1999.

[11] Tiago A Marques, Len Thomas, Jessica Ward, Nancy DiMarzio, and Peter L Tyack. Estimating cetacean population density using fixed passive acoustic sensors: an example with blainville's beaked whales. *The Journal of the Acoustical Society of America*, 125(4):1982–1994, 2009.

[12] Brian McFee, Colin Raffel, Dawen Liang, Daniel PW Ellis, Matt McVicar, Eric Battenberg, and Oriol Nieto. librosa: Audio and music signal analysis in python. In *Proceedings of the 14th python in science conference*, volume 8, 2015.

[13] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

[14] Marie A Roch, Melissa S Soldevilla, Rhonda Hoenigman, Sean M Wiggins, and John A Hildebrand. Comparison of machine learning techniques for the classification of echolocation clicks from three species of odontocetes. *Canadian Acoustics*, 36(1):41–47, 2008.

[15] Melissa S Soldevilla, E Elizabeth Henderson, Gregory S Campbell, Sean M Wiggins, John A Hildebrand, and Marie A Roch. Classification of risso's and pacific white-sided dolphins using spectral properties of echolocation clicks. *The Journal of the Acoustical Society of America*, 124(1):609–624, 2008.

[16] Stanley Smith Stevens, John Volkmann, and Edwin B Newman. A scale for the measurement of the psychological magnitude pitch. *The Journal of the Acoustical Society of America*, 8(3):185–190, 1937.

[17] Jeanette A Thomas, Cynthia F Moss, and Marianne Vater. *Echolocation in bats and dolphins*. University of Chicago Press, 2004.

[18] Mike Van der Schaar, Eric Delory, and Michel André. Classification of sperm whale clicks (physeter macrocephalus) with gaussian-kernel-based networks. *Algorithms*, 2(3):1232–1247, 2009.

[19] Peter Welch. The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on audio and electroacoustics*, 15(2):70–73, 1967.

[20] Gordon M Wenz. Acoustic ambient noise in the ocean: Spectra and sources. *The Journal of the Acoustical Society of America*, 34(12):1936–1956, 1962.

[21] Serge Zaugg, Mike Van Der Schaar, Ludwig Houégnigan, Cédric Gervaise, and Michel André. Real-time acoustic classification of sperm whale clicks and shipping impulses from deep-sea observatories. *Applied Acoustics*, 71(11):1011–1019, 2010.