Project – Part 2

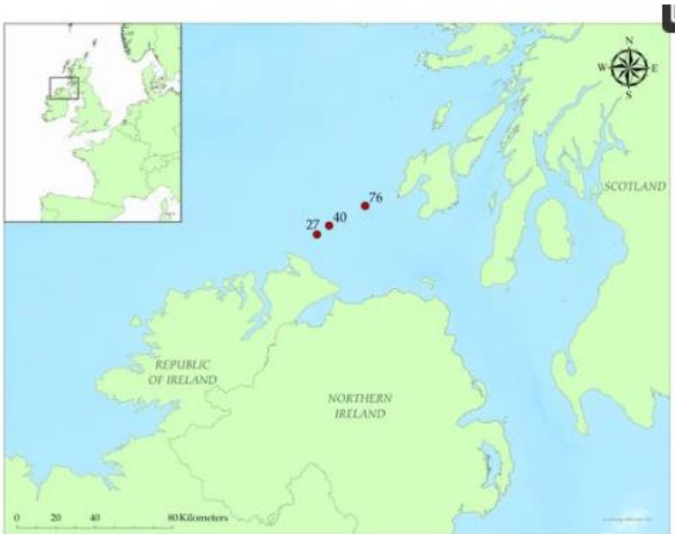
This sub-project is worth 50% of the overall project grade.

This part of the project will focus on data processing/cleansing, followed by training and evaluation.

Dataset – Background:

The dataset provided for this project is private to ATU. It must not be shared with any entity or individual outside ATU.

The dataset originates from a European-funded project conducted by the MFRC (Marine and Freshwater Research Centre) in ATU. The dataset comprises underwater recordings taken by devices deployed by the Celtic Voyager Research Vessel in the Malin Sea. You may read further details on the data collection process and rationale here: Static Acoustic Monitoring of Harbour (Phoca vitulina) and Grey Seals (Halichoerus grypus) in the Malin Sea: A Revolutionary Approach in Pinniped Conservation . This MFRC-led conservation project aims to collect data on grey seals to learn more about their populations and behaviour. Currently many seal (pinniped) populations are declining due to the effects of climate change and noise from offshore windfarms and shipping. The MFRC research aims to better understand these effects.



The Objective

The purpose of this Machine Learning project is to analyse the recorded data set and investigate whether it's possible to discriminate between the different seal calls.

It is advised to commence analysis on discriminating the simplest case (e.g. between just 2 calls – Rupe A and Rupe B or between Rupe B and "no call") before progressing to further analysis. I will provide annotations as well as the raw recordings.

The ultimate goal would be to build a detector that can extract seal calls from an audio recording but that will not necessarily be the outcome of this project.

The features provided to a Machine Learning model will be a spectrogram of the calls. For consistency, each spectrogram (for each call) should be the same size.

Step 1 – Data Pre-processing and management:

The provided uncompressed wav files are large and there is lots of the recordings whether there are no calls present.

The first task will be to build a dataset from the annotated data. I've provided a jupyter notebook that will assist in this regard.

My advice would be to extract a spectrogram for each call. Each spectrogram should be the same size so there will be some pre-pre-processing to find what is the longest call (in time) and the broadest in frequency. This will serve as the baseline for the largest spectrogram. Extract and save (with the metadata) a spectrogram for each call (you can calculate the central time of each call from the metadata). Note, don't save as images but as a raw 2d array of numbers.

You can also create spectrograms for the extra class "no-call" where you build a set of spectrograms from times when there is no annotated call. You may assume that any unannotated region has no call in it. Take care to ensure that the extracted "no-call" spectrograms are from the same frequency region as the call spectrograms.

I would recommend having a single jupyter notebook file that does all this preprocessing.

Step 2 – Model Training

After you've created your dataset of "images", it's time to train your model.

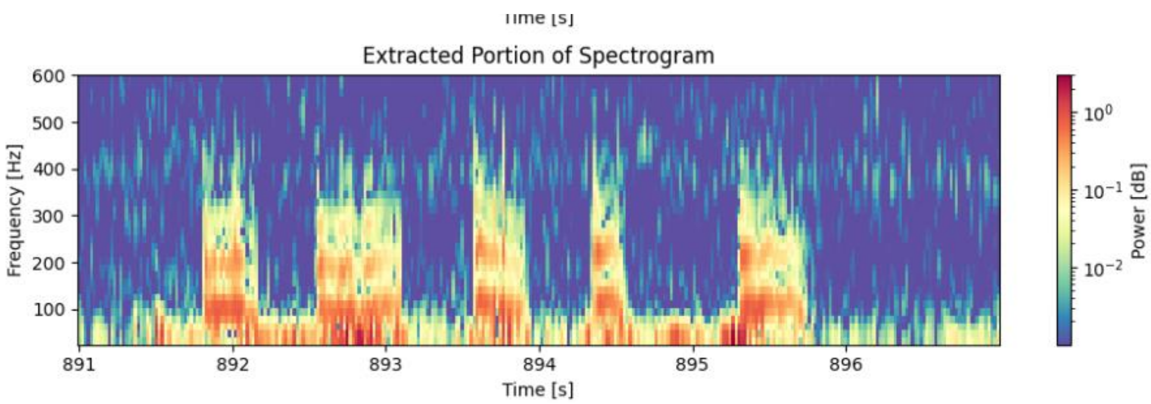
Experiment with different hyper-parameters like we did with faces. You could also experiment with CNNs/Transfer Learning (on similar models)

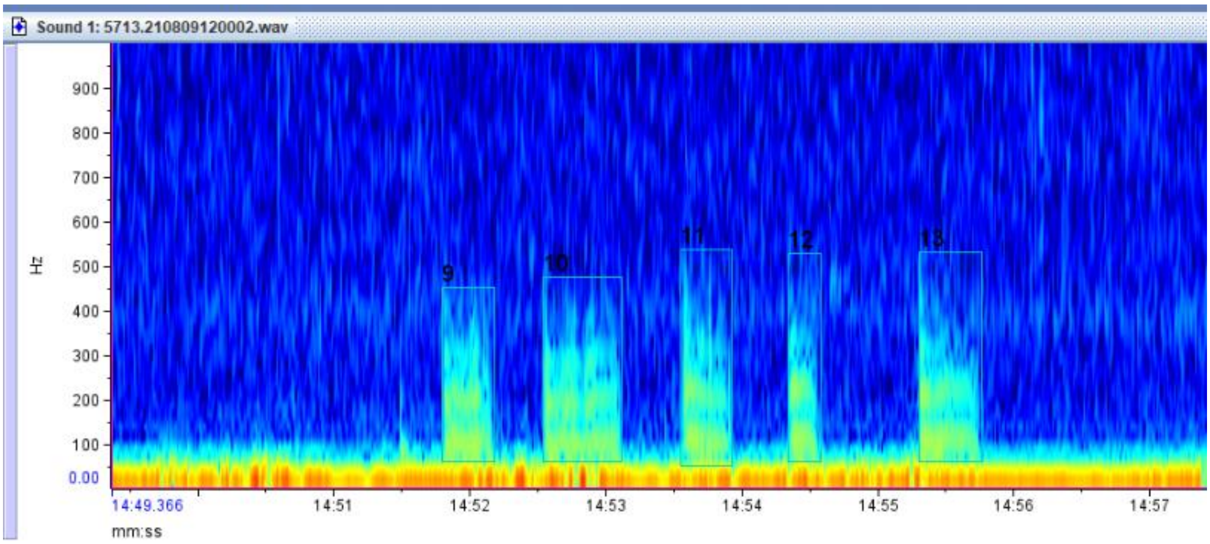
Evaluate the performance of the model on a test set.

Step 3 – Refine

Regardless if step 2 works or not, repeat your process and tune parameters.

You could try adjusting the parameters in my data-extraction/spectrogram notebook (increasing nfft and noverlap will increase the resolution of the spectrogram but will also increase computational cost – so you may need to split the wav files (you could use pydub [python - break up a .wav file by timestamp - Stack Overflow]) or try where you can get more computational resources (e.g. Google Colab)). See the images below:





You can also try different training strategies.

You could also validate your approach on a similar dataset that's been widely investigated ([bird song data set](#) / https://www.kaggle.com/code/sophiagnetneva/cnn-for-sound-classification-bird-calls-90)

If it is working well, you could run your simple Rupe B/No Call classifier across an entire wav file (you will have to hold an entire wav file back during training to achieve this). You will need to ensure if there are Rupe B calls being classified when there is actually no-call, that there isn't another annotated call there instead.

Project Outcome

This is a research project so your marks will be based on your research approach (and not on whether you get a working classifier or not)

Rubric:

- Consistency: 20%
- Research: 25%
- Development: 30%
- Documentation: 25%

As usual, you are bound by ATU Student Code of Conduct (Student Code_Final_August_2022.pdf).

Development/Documentation:

All documentation and development should be performed within a Jupyter Notebook. Regular commits should be made to a **private** GitHub repository. You must add me (brianmcgatu) as a collaborator. The data files will be too big for GitHub so they must be stored exactly as they are in my Spectrogram.ipynb file and your new files should be in the same location. This will allow me to easily run your notebook.

Your final committed notebook should be complete with all code cell outputs populated (i.e. it doesn't require a viewer (me) to replicate the environment and re-execute the notebook to view the results)

Your repository should also include a README – detailing how to recreate your environment for the notebook to execute etc.

Any submission that does not have a full and incremental git history with informative commit messages over the course of the project timeline will be accorded a proportionate mark.

Research:

All research must be captured through the notebook and collated in a bibliography at the end of the notebook. An academic referencing style must be used.

If doing research/comparisons between different models, include that analysis in a separate research notebook and refer to the research in the main notebook. This is to keep the main notebook analysis concise.

Useful Links:

Audio Deep Learning Made Simple (Part 1): State-of-the-Art Techniques | by Ketan Doshi | Towards Data Science

Static Acoustic Monitoring of Harbour (Phoca vitulina) and Grey Seals (Halichoerus grypus) in the Malin Sea: A Revolutionary Approach in Pinniped Conservation

python - break up a .wav file by timestamp - Stack Overflow

bird song data set | Kaggle

https://www.kaggle.com/code/sophiagnetneva/cnn-for-sound-classification-bird-calls-90

https://matplotlib.org/ipympl/