Whale Sound Detection By Lev Shuster

# Problem

# The commercial marine industry is a major contributor to whale death and disruption. In an effort to reduce this industries harm, Marinexplore and Cornell University collected hydrophone recordings with and without whale calls in the North Atlantic with the eventual hope of building hydrophones with compute onboard to live detect whales to prevent collisions. In 2013, Cornell University posted a competition for the most accurate ML models to identify whale calls in these recordings. I wish to try solving this problem using the advances in machine learning in the last decade.

# A collage of images of a television set Description automatically generatedI used to work on a commercial ship in Washington state. While chatting with my ex-boss, he mentioned how interesting the proposed new Bremerton-Seattle ferry project was. The route would be made zero emissions by building a novel electric hydrofoil ferry. One of the major blockers to the project with the extremely endangered orca pods who occasionally venture into Seattle's bay. Relative to past ferries, this project would be much faster, quieter, and effectively have a guillotine cutting through the water. The risk of killing one of the Orcas would be much higher than past project. In response, some advocates proposed launching hydrophone buoys across the ferries route that would listen for orcas and signal the ferry to cut speed and avoid the pod. This solution would require a machine learning algorithm to be created. I found this challenge interesting and wish to take a crack at the problem.

# Data

**To download the dataset, go** [**Here**](https://www.timeseriesclassification.com/description.php?Dataset=RightWhaleCalls) **and click the “Download this dataset” button. Be warned that the audio files are all bundled into an uncommon file format. However, I’ll be using a pre-existing library which converted these audio clips into spectrogram PNGs. This allows my work to be based more on in-class content. Shown on my left are some example images that I’ll be using as my KNN and CNN inputs.**

# I will be using the data from the [original competition](https://www.kaggle.com/c/whale-detection-challenge/data) which is 30000 2-second audio files in the .aiff file format. Each file has been labeled as having a whale vocalization or no whale vocalization. This data is freely available at Kaggle under a copyright that allows educational use. Looking at the old leaderboard, competitors had the greatest success analyzing the spectrograms. This would allow me to treat the project as an image recognition task which seems to be common in the industry and more closely relates to the content of this class.

# Approach

My minimum viable project will be running k-nearest neighbor on the image version of the dataset. I wish to do this with scikit learn to better mirror what I would do in industry. This would also allow me to use K-D Tree to reduce the computation time of each prediction and make even the MVP novel from classwork. I did not see any use of k-nearest neighbors so this will be a novel model for this problem. I do not expect this model to perform well, however, it lets me start working on this project in familiar territory that closely mirrors what we have done in class.

My primary goal will be running ResNet on the image version of the dataset. Resnet is a general-purpose image recognition CNN, that is used by many real-world specialized tasks by retraining the model to a particular task. Resnet is a complex CNN.

A white cube with a black background

Description automatically generated*"Each of the layers follow the same pattern. They perform 3x3 convolution with a fixed feature map dimension (F) respectively, bypassing the input every 2 convolutions. Furthermore, the width (W) and height (H) dimensions remain constant during the entire layer."*

[*Sourced From towardsdatascience.com*](https://towardsdatascience.com/understanding-and-visualizing-resnets-442284831be8)

At the time of the competition, ResNet did not exist, so I will be doing novel work. However, a pre-pre-curser to ResNet, LeNet-5, was effectively applied in a project submission I read. Because both ResNet and LeNet-5 were trained on the same dataset (ImageNet) and are used in the same way (general purpose image categorization CNN that is retrained to specific tasks), I believe this is a reasonable balance between feasible and novel.

I picked ResNet as my main goal because it directly connects to my computer science comps. My group is comparing three different explainable AI techniques on two different datasets. I was responsible for making ML models for one of the datasets, but the other group created a CNN model to identify tumors from brain scans by retraining a ResNet model using PyTorch. I wish to better understand my teammates’ work by using ResNet**+Keras** to classify a different image categorization problem.

My stretch goal is to build my own CNN model to on the image version of the dataset **using Keras**. I will closely model my stretch goal to what we did in class and in our final. I do not see this task as any more work than my main goal, but I wish to priorities the work that will help me with my comps.

# Context

For Data Cleansing I will be referencing [Mabel Villalba Jiménez capstone](https://mabelvj.github.io/capstone_mabelvj/)), the two textbooks from class, and data cleaning examples from the homework. To learn more about the source data I will explore [a paper published by UCSD](http://noiselab.ucsd.edu/ECE228-2020/projects/Report/12Report.pdf) and slides from a Czech lecture on the dataset. For K Nearest Neighbor, I will mostly refer to the homework we did in class and sci-kit-learn documentation. For ResNet I'll mostly be referring to Keras documentation.

# Evaluation

Because there are only two categories, and my data is labeled evaluation will be straight forward. I will also use the evaluation metric proposed by the competition; Area Under the Curve which is created by combining the true positive rate with the false positive rate. I will also create confusion matrixes for each model.

# Concerns

I have no concerns at this point.