DeepWhale: Humpback whales tail-shots classification using Convolutional Neural Networks and Siamese Neural Networks

Marios Andreas Galanis, Vladimir Kozlow

February 19th, 2019

1 Introduction

The identification of animal specimens for population tracking has been a manually intensive task ever since scientists have started to look into the evolution of Earth's biodiversity. In particular, the tracking of aquatic specimens is a very tedious task due to seasonal migrations and unpredictable surface apparition. Photo surveillance systems have been used by scientists to address the difficulties of whale conservation and population monitoring. Conservationists use the shape of whales' tails and unique markings found in footage to identify what species of whale they're analyzing and meticulously log whale pod dynamics and movements. For the past 40 years, most of this work has been done manually by individual scientists, leaving a huge trove of data untapped and underutilized.

Our goal in this project is to help whale conservationists automatize the whale classification task using a deep learning approach. This will allow a fast processing of otherwise extremely tedious tasks. This will also allow the usage of previously unmanageable data such as photographs taken by citizens, and perhaps trigger the development of citizen science in this particular field.

2 Related work

Given that this project is based on a Kaggle competition[1], many approaches specific to this project can be found on the competition website, in the "Kernel" and "Discussion" sections. This competition is similar to a previous competition but has an updated and extended dataset[2]. There is a very large body of literature around few-shots, one-shot and zero-shot learning models which is related to this project. More specifically, our approach is drawn from Siamese Neural Networks models.

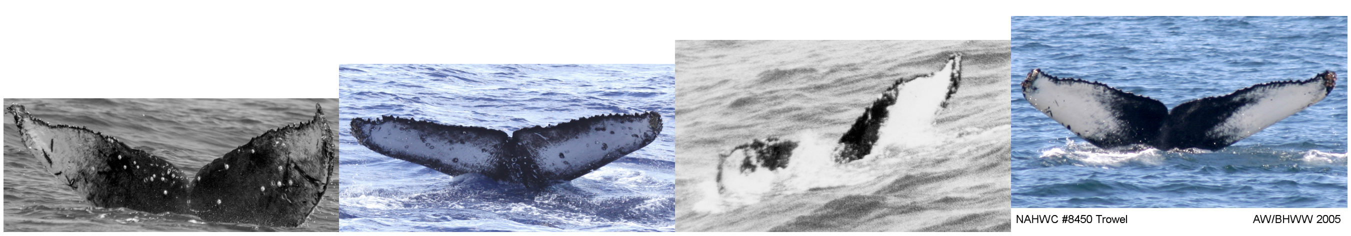
3 Dataset and Features

3.1 Dataset

Our dataset is drawn from a Kaggle competition called "Humpback Whale Identification"[1]. The data for this competition was provided by Happywhale[3], a platform that tracks whale individuals across the world's oceans. The dataset is constituted of 25,361 images of whales' tail. Each training image is identified with one of 5,005 Identification number. One of the Id is "New whale", which means that the image corresponds to a specimen that hasn't been previously recorded.

There is much inconsistency in the original dataset, which implies a fair amount of data pre-processing. The images are of different size/resolution and aspect ratio. The image size ranges from 5959x695 to 77x30 and some images are squared while other are rectangular. Some images have borders which sometime include text. Some images are in RGB and some are in grey-scale. Finally, some images include several whales.

A sample of images from the dataset is presented in Figure 1.



(a) (b) (c) (d)

Figure 1: Four random pictures drawn from the dataset. Aspect ratios: 0.12 for a and b, 0.5 for c and d.

Our approach to solve those issues is presented in the "data pre-processing" section.

3.2 Dataset Split

We split the dataset into a training set containing the 80% of the total examples (picked randomly) and a development set containing the remaining 20% of our examples. The test set was provided for us from the Kaggle competition unlabeled, so we won't be able to evaluate our model's results on the test set until we submit the model to the competition.

3.3 Data Pre-Processing

During the data pre-processing phase we converted every image into gray-scale and 200x200x1 size. The labels set was converted into a matrix with number of rows equal to the total number of images and number of columns equal to the number of different classes. Each row had an entry with a value of 1 at the column corresponding to the ID of this particular whale (depicted in the image corresponding to this row) and zero values at any other entry. Also, we decided to use the mirroring augmentation technique because some whale IDs were appearing only a few times in our training set.

4 Methods

In this section, we present the methods that we plan to implement.

Architecture:

We will implement two different architectures for this identification task, a simple one and a more advanced one:

- Classic Convolutional Neural Network: we will start with a classic CNN to analyze how a basic model can learn this complex few-shots learning task.
- Siamese Neural Network: given that the task we want to learn is very similar to face verification with few-shots/zero-shots learning, we will then try a Siamese Neural Network architecture.

Loss functions:

We will try two different loss functions, each corresponding to one of the architectures:

- Cross Entropy Loss for the classic Convolutional Neural Network
- Triplet Loss for the Siamese Neural Network

Evaluation Metrics:

The metric is a key component of this project because a trivial solution would be for the model to classify each whale as new_whale. Another problem is that, since we plan on using a Siamese Network, not only do we want to maximize the probability of the right class, but also to minimize the probability for all of the other classes. Hence, a relevant metric for this task will be the Mean Average Precision at 5 (MAP@5) presented below:

$$\sum_{u=1}^{U} \sum_{k=1}^{min(n,5)} P(k) \times rel(k) \tag{1}$$

Initialization

We will use pretrained weights extracted from a Kernel[4] shared through the competition website in order to get to make our training faster.

Optimization

We expect overfitting to the training set due to the imbalance of the dataset. Therefore, we plan to use data augmentation, Batch Normalization as well as L2 regularization to prevent it. We will optimize hyper-parameters using a coarse-to-fine random search.

5 Preliminary Results and Discussion

So far, we have tested the classic CNN architecture and despite a the small number of training examples chosen to test it, training has been so computationally intensive that we haven't got any relevant results. We have tried to use Amazon Web Services to solve this issue but we were not able to launch a new instance because our account was blocked to a maximum number of instances of 0. We are now waiting for the customer support to change this limit.

6 Future work

Once the computation issue will be solved, we plan to optimize the Siamese Network architecture that we have found in Martin Piotte's Kernel[4]. This approach seems very promising and as described in the Methods section, we plan to use several optimization methods to increase the Mean Average Precision of our model.

7 Project Code

The project code can be found on `https://github.com/ultimatemalakas/Whales`

8 References

[1] https://www.kaggle.com/c/humpback-whale-identification

[2] https://www.kaggle.com/c/whale-categorization-playground

[3] https://happywhale.com/home

[4] https://www.kaggle.com/martinpiotte/whale-recognition-model-with-score-0-78563/output