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

Marios Andreas Galanis, Vladimir Kozlow

March 19th, 2019

1 Introduction

The identification of animal specimens for population tracking is a tedious task that has been done manually ever since scientists started to investigate the dynamics of Earth's biodiversity. In particular, the tracking of marine mammals is a tricky task due to seasonal migrations and unpredictable surface apparitions. Taking photographs of the specimens is one of the most common technique used by scientists to survey marine mammals' populations. Conservationists then use the shape of whales' tails and unique markings found in footage to identify what species and what specimen of whale they're analyzing and meticulously log whale pod dynamics and movements. This work has been mostly done manually by individual scientists, leading to a huge amount of data being 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 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 Network models.

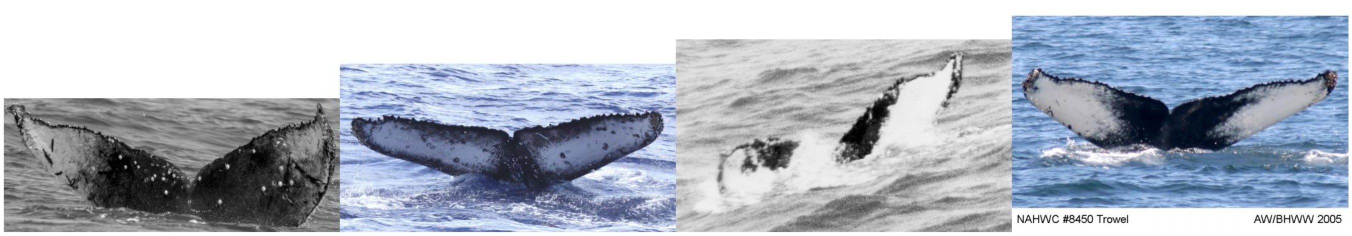
3 Dataset and Features

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 numbers. One of the Ids is "New whale", which means that the image corresponds to an individual 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 sometimes include text. Some images are in RGB and some are in gray-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 decided to 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 the examples. The test set was provided for us from the Kaggle competition unlabeled, so we were not able to evaluate our model's results on the test set until we submitted the model to the competition.

3.3 Data Pre-Processing

In our simple convolutional neural network approach 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 under 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. In the other model (siamese neural network) we start by rotating the images where the whale fluke is pointing down instead of up as usual. The next step is to convert the images to grayscale and then we apply affine transformation maping a rectangular area of the original image to a square image with resolution 384x384x1. Data augmentation is also performed during training by the addition of a random transformation that composes, zoom, rotation, shear and shift. This random transformation is skipped during testing. Finally, the images are normalized to zero mean and unit variance.

4 Methods

In this section, we present the methods that we implemented.

Architecture:

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

- Classic Convolutional Neural Network: A classic CNN was tried to analyze how a basic model can learn this complex few-shots learning task.
- Siamese Neural Network: Our second model is a Siamese NN with a more complex architecture inspired by a Kernel from Martin Piotte posted on Kaggle[4]. Since this was the most successful network let's describe its architecture a little bit more:
  This Siamese NN is made of two models: the head model and the branch model. The head model is a CNN that transforms the input images into vectors of features describing the whales. This head model is inspired from ResNet[5]. The head model is used with the same weights to compute the encoding of two images. The branch model is then used to compare the encodings of the two images to decide whether the pictures are showing the same whales. The branch model is also a CNN, which starts by computing and concatenating the sum, the product, the norm, and the squared difference of the two encodings. The vector obtained is then passed through two convolutional layers, reshaped, and finally passed through a fully connected layer with one output to compute a weighted average. This weighted average is a metric of similarity between two encodings of images. The particular architecture of the head and the branch model can be found in the code.

Training data construction:

This is the trick that increased the overall Siamese NN model's accuracy more than any other architecture or parameter/ hyperparameter choice. The goal is our network to pick the one correct whale among all the whales in the training set. While, it should score the correct whale high, it should also score every other whale lower. In order to force all the other whales to a low probability the training algorithm is fed with pairs of pictures with increasing difficulty, as evaluated by the model at any given time. Essentially, like the case of adversarial examples (presented in class) the training is focusing on the pairs that the model is getting wrong. Given the small number of training examples we also want to make sure that the model is actually looking at the whales and not some random parts of the images like the waves or the clouds in the sky. To ensure that, we make sure that the data presented to the model is unbiased. Indeed, if a picture is used very often in negative examples, the model will might learn to always mismatch this picture without comparing the actual whales. Thus, we are presenting each image an equal amount of times, with 50% positive and 50% negative examples, leading the model to focus on recognizing whales. Finally, to randomize the selection, and control the matching difficult, we used a random matrix. This random matrix has values uniformly distributed between 0 and K. As the value of K gets larger, the matching becomes more random and as K gets smaller, the pairing becomes more difficult for the model.

Training Procedure:

- Classic Convolutional Neural Network: For the classic CNN we chose the Adam optimizer with the default keras settings and the categorical cross entropy loss. We trained this network for 10 epochs and we realized that after 8 epochs the weights converged to some final values. So, we decided to not train it further.
- Siamese Neural Network: For the Siamese NN the training process started without any L2 regularization. After 250 epochs, the training accuracy is very high but the model also starts to overfit the training set. Therefore, at this point L2 regularization was added. The learning rate is kept constant for the first 150 epochs, then it is decreasing until the 250 epoch where it resets to its starting value to decrease again every 50 epochs. Moreover, the constant K started with a very large value (because at the early steps we want the pairs of images to be random and easier for the model to understand the differences between them) and it kept decreasing as the number of epochs was increasing (since we want the pairs of images to become harder and harder for the model to separate as the model is training more and gets better). Finally, the triplet loss was used for the Siamese NN.

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.

5 Results

- Classic Convolutional Neural Network: The trained simple classic CNN was used to predict the labels of our test set but we found that it was always predicting the same class (due to the huge class imbalance between the class "new whale" and the rest of the classes). Therefore, the model's accuracy was always about 30% which is the percentage of "new whale" class examples.

- Siamese Neural Network: For the Siamese Neural Network, we used the trained model and re-ran it for one epoch with different values of K to assess how well it was trained. Again, the larger the value of K, the more randomly the negative example is selected. We tested the model against 4 values of K as presented in Figure 2. We obtained the largest accuracy for a value of K of about 1, which shows that the model was well trained up until this level of difficulty. We also measured the computation time of each epoch and saw that solving the linear assignment problem took much longer when the value of K decreased. There is thus a trade-off between computation time and accuracy to be made between a better accuracy and a lower computational time.

K	Binary cross-entropy loss	Accuracy	Mean score	Computation time (s)
100	0.672	0.575	0.540	2433s 3s/step
10	0.672	0.566	0.462	2424s 3s/step
1	0.455	0.792	0.653	6190s 7s/step
0.5	0.658	0.591	0.579	7629s 9s/step

Figure 2: Results of the model on one epoch for different values of K, with a learning rate of 0.00064s

Based on the MAP@5 evaluation metric our model got a score of 82.297% when we submitted it on the Kaggle competition website. The test labels were not provided on Kaggle which prevented us from analyzing our errors on the test set. The list of the 5 best predictions for 5 whale's images are presented in Figure 3.

File	ffb03ac13.jpg	ffb381363.jpg	ffce3d4a2.jpg	ffd30b649.jpg	ffd73a2ad.jpg
Picture					
Predictions	w_3bf2653 new_whale w_6ebed02 w_bc7de9f w_bfcad53	new_whale w_4d7df64 w_f4e69f1 w_e2372d6 w_f08fb81	w_c5260b8 new_whale w_286d2cc w_72f3685 w_8cee3d3	new_whale w_b99f945 w_c16cb2c w_6b90e54 w_659bdb8	w_8694cb5 new_whale w_5072c08 w_2a7603f w_e4502b3

Figure 3: ID predictions for 5 random images in the test set

6 Analysis of Results

- Classic Convolutional Neural Network: Since the simple CNN approach didn't provide good results we decided that it was pointless to try improving the model's architecture, use data augmentation on training data, or tune the model's hyperparameters. Instead, we decided to focus on the Siamese Neural Networks because the nature of our problem and our dataset makes the Siamese NN approach look much more promising.
- Siamese Neural Network: The results of this model are clearly way more promising. More specifically the overall accuracy (on training set) increased from 30% (classic CNN) to 79.2% and of course this model does not predict always the same class. Furthermore, the Siamese NN gets a very good MAP@5 score which is very important for this particular application. Indeed, even if the model fails to predict the correct class it is important to know that the correct class is one of the top 5 predictions of the model because the researchers can manually evaluate a new picture of a whale and compare it with the top 5 classes predicted until they find the actual true label of this new picture.

7 Insights and Discussion

From our work on this project we realized that a simple CNN is not good enough at performing such a complicated classification task with a small imbalanced dataset. Simple data augmentation techniques are not sufficient to make the performance of a binary-classification CNN adequate. The fact that our dataset contained only a few pictures of each whale made the use of a Siamese NN necessary. But, even after some good data preprocessing and augmentation the use of a Siamese NN with a good architecture is not enough to achieve a satisfactory performance. The problem is that if the image pairing is random and each image is used a random amount of times the network will most likely learn to always mismatch the images of the smaller classes and the new images provided in a test set. Moreover, the model can also focus on random elements of the images instead of the actual whales. That's why the most important task in creating a good model for this project is to ensure that the image pairing will be gradually more difficult as the model becomes better and better and at the same time the model will have the opportunity to train on the images of every class on a similar amount of positive and negative examples.

8 Team Member Contributions

Marios Andreas Galanis: Simple CNN coding and report writing
Vladimir Kozlow: Siamese NN implementation and report writing

9 Project Code

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

10 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

[5] Kaiming He et al., Deep Residual Learning for Image Recognition