Whale Identification using Convolutional Neural Networks

Katrina Zathey  
*School of Continuing Studies,*

*York University,*

Toronto, Canada  
[katrina.zathey@gmail.com](mailto:katrina.zathey@gmail.com)

Crystal Yunan Zhu  
*School of Continuing Studies,*  
*York University,*

Toronto, Canada [yunanmath@gmail.com](mailto:yunanmath@gmail.com)

Queenie Tsang  
School of Continuing Studies,

*York University,*

Toronto, Canada

qpltsang@gmail.com

# Introduction (*Heading 1*)

This paper describes a proposed machine learning model to solve the Humpback Whale Identification problem found at https://www.kaggle.com/c/humpback-whale-identification/overview. The challenge involved identifying the whale based on a picture of its tail. An added layer of complexity was that not all images corresponded to an already known ID. In this case, the image was to be labelled with the *new\_whale* identifier. Common issues included sparse data, images of various resolutions and sizes, and inconsistent image properties (e.g. some black and white and some in colour).

# Methods

## Dataset Description

The dataset used was downloaded from <https://www.kaggle.com/c/humpback-whale-identification/data>. It is a large dataset (5.55 GB) which contains 33, 300 files. The dataset contains images of whales which correspond to more than 3000 whale Ids. The dataset is already split into test and training sets which contain 7960 images, and 25,361 images respectively. The test dataset however did not contain identifiers (as to defeat the purpose of the competition) and as such only the train data was used. The data consists of images of humpback whale photographs in either black-and-white or colour .jpg files. The train folder also contains a train.csv file which maps the training images to the whale ID.

## Methods

The model was build using keras and various convolutional neural networks. The focus was on using prebuilt models to assist in the trial of multiple approaches and to overcome the general unfamiliarity with the creation of new neural net models. Image processing was performed prior to being fed into such models. This preprocessing includes image resizing, experimenting with 1 colour channel (grayscale), or 3 colour channels (RGB), various image transformations (rotation, flips, zoom etc.) but did not include bounding boxes for image input which will be elaborated in the following section.

For the initial model where both new and existing whales are included, we tried four pretrained convolutional neural networks (VGG16, VGG19, ResNet50 and ResNet101) to apply transfer learning to identify the whales in our dataset. We also experimented applying these four CNN to a subset of the whales where only those appear more than 10 times are kept. While adding new layers to the pretrained ResNet50, we also tried GlobalMaxPooling2D, instead of GlobalAveragePooling2D in the pooling latyer, to see whether it would make a difference. Batch size was also increased from 8 to 100 for training data. ResNet101 was also applied to transformed grayscale data.

## Bounding Boxes

Attempts to use bounding boxes to limit the image processing to just the area of interest led to two different approaches, neither of which was successful. The first was using a pretrained model such as YOLO (You Only Look Once). The concept was to reuse an already successful bounding box model and have the output of that model feed into the image recognition model. A roadblock was encountered when looking into the types of images that this model was configured to use. None of the classifications in this model were of whale tails. It would have been possible to look into training a custom configuration and so this was attempted with a library that had a more user friendly configuration, OpenCV. A roadblock was encountered in trying to train a domain specific bounding box model when the input would have required both the images to bound and the coordinates of those bounds as part of the training set. As the data provided in this problem did not have coordinates for any of the over 25 thousand images, it was infeasible to train a domain specific bounding box model.

# Results

An initial model was build using VGG16 which attempted to label both IDs and *new\_whale* for all inputs across 3 colour channels. This approach seemingly produced random results and was not successful. The problem was then approached as 2 models, one being a binary classification of new/existing and the other being a categorical problem of matching those predicted to be an existing whale to the corresponding ID This made intuitive sense as the each model would handle one part of the problem. This was attempted again using 2 VGG16 models and 3 colour channels. This seemingly performed well in training accuracy (around 60%) for the binary classification but did not perform well in categorical accuracy (zero accuracy when accounting for random chance). Neither model performed well in testing accuracy. Color channels were then reduced to 1 (grayscale) to account for black and white images that may have perturbed the model and the number of epochs used in training were increased from 20 to 50 to allow the models a better chance at accurately predicting the input. This resulted in an overall accuracy of 29% with both models combined. Consequently the testing accuracy of this same model however with all 3 colour channels only resulted in an accuracy of 12%.

From experiments on the initial model, we found Loss does not monotonically decrease across epochs either before or after unfreezing the base model when batch size is 8. Accuracy does not monotonically increase across epochs either before or after unfreezing the base model when batch size is 8. Increasing batch size from 8 to 100 leads to significantly lower computational speed, but the training loss is lower. VGG16, VGG19, ResNet50 and ResNet101 have similar performance on this whale dataset. Training using only the subset (whales appear more than 10 times) significantly reduced the training accuracy. Training ResNet101 on grayscale dataset does not improve the accuracy. Adam optimizer works similarly to RMSprop.

All approaches ultimately underperformed the naïve approach of labeling all images as *new\_whale* which produced a testing accuracy of 38%.

# Discussion

Ultimately, this group failed to produce an adequate model for the problem. None of the attempts or combinations outperformed a naïve approach. In looking at some of the top placed entries one of the entrants was able to reach an overall test accuracy of over 90% using ArcFace (Additive Angular Margin Loss for Deep Face Recognition) and senet154. The approach of using convolutional neural nets was likely not the ideal type of model to apply to this problem.