Machine Learning Engineer Nanodegree
Capstone Proposal

Adilbek Madaminov

January 7, 2019

For my capstone project, I propose to use a Kaggle competition titled "Humpback Whale Identification" [1]. This is an ongoing competition to identify individual whales from the images of their tails based on the training set of about 25,000 images and 5,000 labels, each label representing a unique whale.

Domain background

Kaggle was founded in 2010 and is currently most famous for data science and machine learning competitions posted by companies and organizations looking to solve real-world problems like gesture recognition and disease identification. Kaggle also has a community of over 1 million data scientists and machine learning experts who use the platform to share and discuss data and code [2].

The data for this competition is provided by Happywhale.com, a platform for submitting images of whales for whale identification. The winning algorithm of this competition will help whale conservation efforts by allowing scientists to improve the process of identifying whales in new pictures.

Problem statement

The whale identification competition is a multi-class classification problem that challenges to build an algorithm that best identifies individual whales in the given test set of nearly 8,000 images. The performance of an algorithm is evaluated on the submission file that maps each of the test images to a whale identification code (whale Id).

The algorithm is expected to use the training set of about 25,000 images to learn identities of about 5,000 different whales. A particular challenge of this competition is the scarcity of the training images – most of the labels have only one or two images associated with a label (typically, machine learning algorithms are provided hundreds or even thousands of images per label, so this competition will almost certainly require me to research and use an advanced approach).

Datasets and inputs

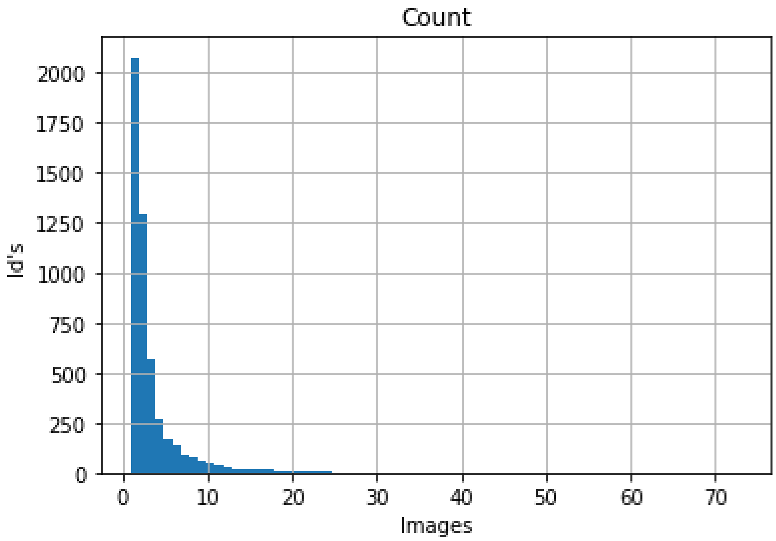
This project’s dataset is publically accessible via the following competition link:

https://www.kaggle.com/c/humpback-whale-identification/data

The dataset contains two folders with images: one with 25,363 training images and one with 7,961 testing images (all .jpg files). Additionally, the dataset comes with the text file "train.csv" that maps each training image to a corresponding whale Id (some of the training images are mapped to "new_whale" instead of an Id to indicate that the label is not known).

The training images and their labels are to be used to train an algorithm to recognize individual whales, and the testing images are for testing the performance of the algorithm. The testing images do not come with labels but they are known to Kaggle's submission system which is how it evaluates submissions.

One notable aspect of the dataset is scarcity of the training images. This scarcity is not uniform – while most labels have 1 or 2 images, there are still hundreds of labels with 3 or more images (up to 73 images for the most prolific label). Also, about 38% (9,664) of the training images are unlabeled (Id = “new_whale”). Ignoring the unlabeled images, here is the full distribution of images per label (Id):



This imbalance in the dataset is likely to cause overfitting towards the more represented labels. Unless addressed, the algorithm will be prone to erroneously predict some whale Ids more frequently than others simply because it saw more images for those Ids during training. I will research the best way to address this problem, but to start, I will try removing some images for over-represented labels [3] (resampling the dataset) and/or generating more images for the under-represented labels using image augmentation.

Another notable aspect of the dataset is the size of the images – they are not all the same size or the same color mode. Because neural networks typically require a fixed input size, I will need to resize the images to a common width and height and to standardize their color mode. I expect the specifics of this process to become more clear once I analyze the images in more detail during the project.

Finally, in order to be able to compute an accuracy score without making a full submission on the competition website, I will need to split the training set into training and validation sets. I will explore the possibility of making the split using keras’ built-in functionality or doing it myself by putting aside some of the training images.

Solution statement

The project’s solution will be an multi-class classification algorithm that can be trained on the training images and labels provided in the dataset, and will be able to make predictions of whale Id's for all of the testing images. This algorithm will output a submission file that maps each of the testing images to a whale Id (more precisely, up to 5 whale Id's per image are allowed). When submitted to Kaggle, the submission will yield an accuracy score and a ranking which will be visible under "Leaderboard".

As a contingency, I will keep track of my own accuracy score for each model. This may become necessary if I have to work with a smaller subset of the dataset due to computational or bandwidth difficulties I may encounter (if the training time starts exceeding practical limits, the full dataset turns out to be too much for evaluating my project by Udacity, etc.)

Benchmark model

For a benchmark model, I will use a simple convolutional neural network (CNN) such as one described in Lesson 2.18 of the Deep Learning section of the Machine Learning Nanodegree (with 3 convolutional layers, 3 max pooling layers, etc.). While this model won’t be expected to make highly accurate predictions, its result will serve as a baseline for any modifications and improvements I will undertake.

Evaluation metrics

My first choice for evaluation metrics is to use the competition's submission score (MAP@5) as detailed in the following competition link:

https://www.kaggle.com/c/humpback-whale-identification#evaluation

The competition allows to make up to 5 predictions per image. This metric is an average of scores for all images, where each image scores 1.0 if the correct label is predicted in the 1st of the five predictions, 1/2 if in the 2nd, 1/3 if in the 3rd, 1/4 if in the 4th, 1/5 if in the 5th, and 0 if none of the five predictions are correct [4].

My second choice is to use my own accuracy score such as keras' accuracy score or F1 score. This may also be necessary if I end up using a smaller subset of the dataset due to computational or bandwidth issues. Here, I will have to account for the imbalance in the dataset that I mentioned earlier. If I rebalance it then keras’ accuracy score may work, but without rebalancing, a better metric may be F1 score. I will research both scenarios.

Project design

I plan to complete the project using the follows steps:

1) Analyze the dataset for overall size, missing data, and dimensionality. In this step, I will try to answer questions such as "How many images do I have to work with?", "What are the dimensions of the images - width, height, and color channels?", and "How many images are there per label?"

2) Preprocess the images in ways that are necessary for training in a later step. Here, I will consider removing any "bad" or unlabeled images, resizing them to a standard size, reducing color mode to grayscale, rescaling each pixel to a value between 0 and 1, splitting into training and validation sets, etc.

3) Build and train a simple convolutional neural network (CNN) as a benchmark model. I will try to train it on the full dataset and create a full submission file with predictions in order to obtain an official competition score. If training time gets too lengthy, I may have to resort to working with a smaller subset of the dataset instead. In that case, it will be important to select a sample correctly given the imbalance of the dataset, as was noted by my first reviewer. I will most likely use stratified sampling [5] in order to avoid losing many whale Ids.

4) Apply image augmentation and transfer learning. I will research ways to generate new training images by applying random transformations on existing images like rotation and shifting. This step will be crucial for improving on the benchmark model given the scarcity of the training images (most labels seem to have only 1 or 2 images). I will also try using a pre-trained network like ResNet50 in order to further improve on the benchmark model and my official competition score.

Citations

[1] https://www.kaggle.com/c/humpback-whale-identification
[2] https://en.wikipedia.org/wiki/Kaggle
[3] https://machinelearningmastery.com/tactics-to-combat-imbalanced-classes-in-your-machine-learning-dataset/
[4] https://www.kaggle.com/pestipeti/explanation-of-map5-scoring-metric
[5] https://en.wikipedia.org/wiki/Stratified_sampling