Humpback Whale Identification

Identifying a whale by its tail

Vitalii Duma
Data Science Master Programme
Faculty of Applied Sciences
Ukrainian Catholic University
Lviv, Ukraine

Serhii Tiutiunnyk
Data Science Master Programme
Faculty of Applied Sciences
Ukrainian Catholic University
Lviv, Ukraine

***Abstract*—This document is a Machine Learning project report dedicated to the humpback whale identification problem which was set as a `kaggle` competition "Humpback Whale Identification". The approach used for solving the problem is a Siamese Neural Network [1] with triplet loss [2] and some additional heuristics.**

***Index Terms*—Siamese Neural Network, image embeddings, triplet loss, cosine similarity, metric learning, image identification, whales identification**

I. Introduction

A. Importance of the problem

The problem of recovering whale populations is very actual nowadays due to the adapting to warming oceans, intense whaling during the last several centuries and competition with the fishing industry for food.

The scientists use the photo from surveillance system to register the whales activity and migration. They use the shape of the whales' tails and special marks there to identify the species of the whale.

The vast majority of this work is being done manually by scientists. So, the goal of this project is automation whale identification which will improve the monitoring process, help to get rid of routine job and increase the scientists' performance.

B. Potential impact

Solution of this problem can be applied for migration monitoring of other animals which might help scientists to take care about endangered species.

Along with it, some new heuristics and approaches applied to Siamese Neural Network (SNN) can improve existed solutions for other problems, such as:

- One-Shot Image Recognition [1]. It is very similar to this case dataset as the vast majority of classes have only one example. Since siamese networks for the first time study discriminatory functions for a large concrete data set, they can be used to summarize this knowledge and for completely new classes and distributions.
- Pedestrian tracking for CCTV [3]. In this project SNN is being used together with size and position features of images to detect several persons in the camera view area. SNN learns associations between several frames and trajectories.
- Matching resumes to jobs [4]. In this use case, SNN tries to match job offers and candidate's resumes. Here can be applied natural language processes (NLP) to retrieve deep contextual information from offer description and resumes, compare its embedding and force to increase distance between unsuitable pairs.

Siamese Neural Networks are widely used for solving problems in different ares. So some enhancements in solution whale identification task can lead to improvements contiguous tasks which were mentioned above.

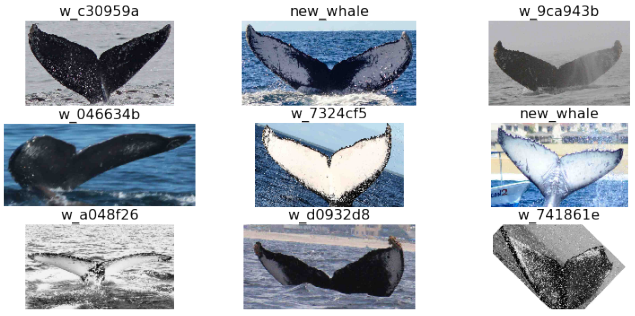

Fig. 1. Whale's tail examples

II. Data collection and preparation

A. Exploratory Data Analysis

The data for the whale identification task was provided by the organizers on the competition platform `kaggle.com` [5]. The dataset contains thousands of images of humpback whale flukes. Individual whales have been identified by researchers and given an `id`. The target is to predict the `id` of images from the test set which will be used for evaluation. What makes this task such a challenge is that there are only a few examples per instance of whale `id` (`3,000+` whale `Ids`).

First of all, it is worth to see the examples of whale's tail images. Ont the Fig. 1 the random images are shown.

On the next step, it was detected being a repeated images which are binary identical or very similar which differ only in terms of brightness or color contrast but the general shapes are

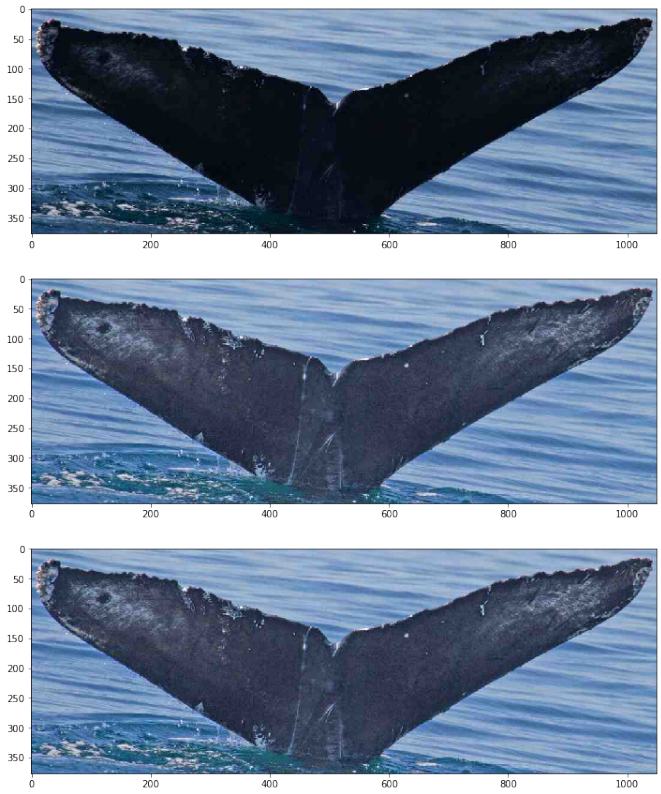

Fig. 2. Similar whale's tail examples

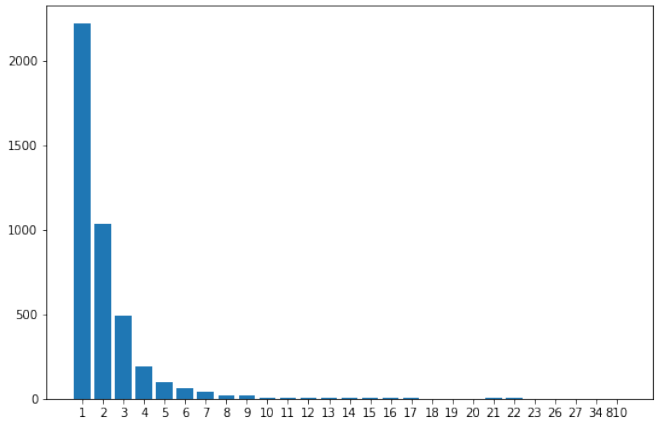

Fig. 3. Data skewness

identical. Fig. 2 is an example of three identical image among training and testing sets.

Moreover, the data is very skewed in terms of the number of images per category. Let's draw the plot of the number of categories which have corresponded number of images. On the Fig. 3 shows the bar plot of the number of categories which have corresponded number of images. It means that vast majority of categories have only one or two examples.

One more interesting thing in the whale's is having identical rotated images. If it was detected correctly there are 10 rotated image copies. It was detected and reverted to the initial image.

Fig. 4. Rotated image example

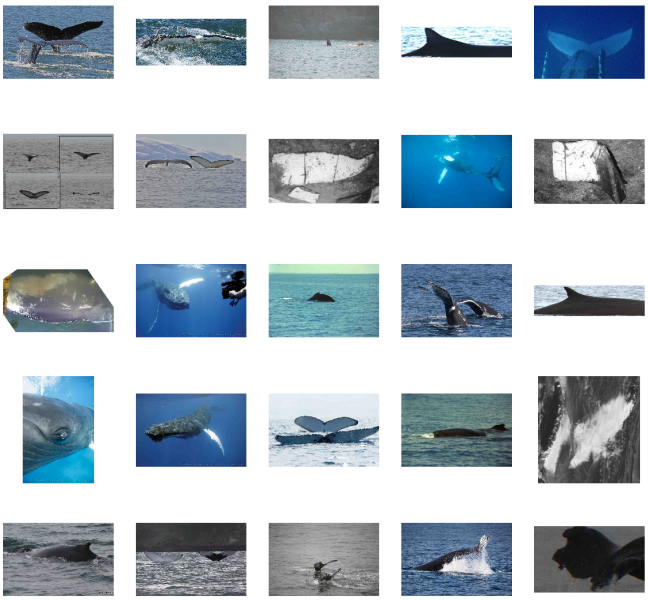

Fig. 5. Manually excluded images

On the Fig. 4 shown an example of rotated whale's tail image.

It was noticed that there are images unhelpful or even harmful to training. It is images with two or more whale's tails per image or images which displays whale's body in addition to the whale's tail. So, such images were marked manually. On the Fig. 5 shown images excluded from the training set.

B. Data preparation

First of all, whole data set should be leaded to the same size. As it is shown on the Fig. 6, image data set has different image sizes. So, each images was transformed to the 256×256 dimension by affine transformation [8] using PIL library. Affine transformation maps a rectangular image dimension to a square.

The biggest improvement in solving this problem was done by the method of generating appropriate pairs of images for the training CNN. Each batch of images consists of pairs in which half of them match the same whale `id` and another half of pairs of different whales. As it was mentioned earlier, there are only a few image instances per whale `id` so each images was used for both matching the same whales and differentiation. Furthermore, the pictures which differs whales i.e. with different `ids`, are selected to be difficult to distinguish. The similar approach is widely used in adversarial training [6].

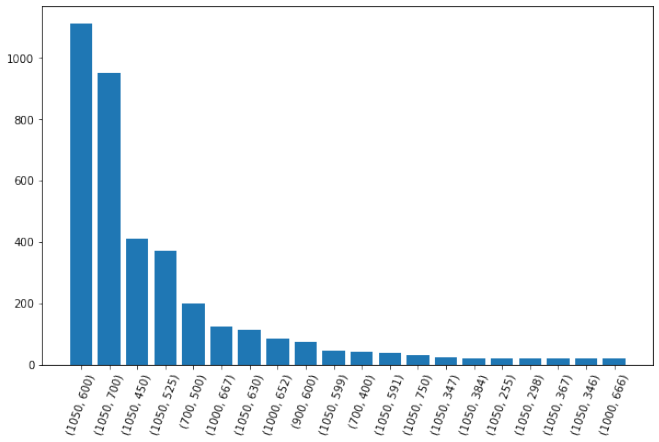

Fig. 6. Image size bar plot

Fig. 7. Image augmentation example

There are several approaches how to evaluate the difference (distance) between the images. The linear sum assignment [7] method was used to find the most difficult pairs of whales images for matching.

At first, as it was mentioned above, there are identical images in the training and test set. Some of them are pixel to pixel identical and the others differ by brightness background color, contrast, size etc. On the step of data preparing all such images are being detected.

At second, some images can contain more than one whale, the whole whale, or beach. These images were deleted from training dataset. There were also a few images of rotated upside-down whale flukes. They were accordingly rotated back. All images were cropped due to bounding-boxes data found in a custom kernel. Bounding boxes were not very accurate, so a cropping margin was introduced to save important parts of flukes.

Along with it, there were several methods applied to images to increase the number of samples for the same whale. During training data augmentation was conducted by applying a random shift, shear and zoom. On the Fig. 7 shown an image augmentation example.

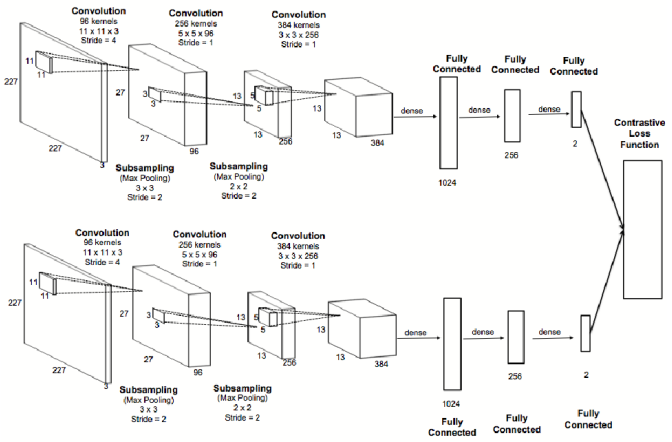

Fig. 8. SNN architecture

III. Modeling

Siamese neural network consists of two identical Convolutional Neural Networks (CNN) which transform the whale images to the embeddings vector. The general architecture of SNN is represented on Fig. 8. The input of the model is two images of whales. The CNN is trained to increase the distance between outcome vectors if the input images have different `ids` i.e. belong to different whales. In the other words, the same CNN with the same weights is being used for both input images.

By comparing each image from the test set with each image from the training set it is possible to determine the most likely whales by sorting them by the probability of a match.

The part of model which compares output vectors is a field for experiments. The simplest way is just cosine distance measure with threshold and a loss function. There are pros and cons of applying the distance measure.

First of all, the distance as a measure of vectors similarity is intuitively understandable. Moreover, swapping the the order of input images does not change the result which is also makes sense.

Along with it, there some drawback of applying the distance measure.

- The distance measure will recognize two objects with a zero values as a perfect match of two images.
- At the same time, two vectors with large but a little bit different values will be considered as good match but not perfect as they are not equal.
- Also distance measure makes applying ReLU (Rectified Linear Unit) activation inappropriate as it zeros negative values which leads to loosing some part of information and force network to work only in the positive part of the space.

There were several experiments conducted to choose the embedding vectors comparison.

- Cosine distance measure with threshold built into loss function.

- Difference between vectors passed to the input of logistic regression.
- More complicated transformations under the vector's features like sum, product, absolute difference, squared difference etc. passed to the logistic regression.
- The same transformation as in the previous point but passed to the neural network.
- Different pretrained models used (`ResNet34`, `VGG16`, `DenseNet`)

The cosine distance configuration showed the best results. `ResNet` showed best accuracy.

The output of the network is a logistic regression.

In addition, there was applied changing learning rate approach during training but it did not give a significant enhancement.

A. CNN model

The branch CNN model is pre trained `ResNet34` and fully connected layer to retrieve an appropriate image embeddings. One fully connected layer is enough to train, as the training data is not big enough to increase the number of parameters and still keep the model expressive enough. Each block processes maps with lower and lower resolutions, with intermediate layers of the pooling layers.

IV. Evaluation

There are two evaluation stages. The first one is `kaggle` competition evaluation and the second one is evaluation after the training to compare models before submitting to the `kaggle` checking.

A. Competition evaluation

Competition submissions are evaluated according to the Mean Average Precision 5. 5 is the number predictions per image.

$$MAP@5 = \frac{1}{U}\sum_{u=1}^{U}\sum_{k=1}^{min(n,5)} P(k).$$

where U is the number of images, $P(k)$ is the precision at cutoff k, n is the number predictions per image. For each `Image` in the test set, model may predict up to 5 labels for the whale `Id`. Whales that are not predicted to be one of the labels in the training data should be labeled as `new_whale`.

`Kaggle` evaluation result is shown on the Fig. 9.

B. Model evaluation

The same evaluation measure is applied for local evaluation i.e. comparison models before submitting. As there is no labeled training set for developers, test set was taken from labeled training set. But there is an issue with splitting train and test sets as there are a lot of classes with only one image instance. Despite of having augmented images, it was decided not to take into account augmented images for evaluation.

Only classes which include more than 3 images were taken for model validation.

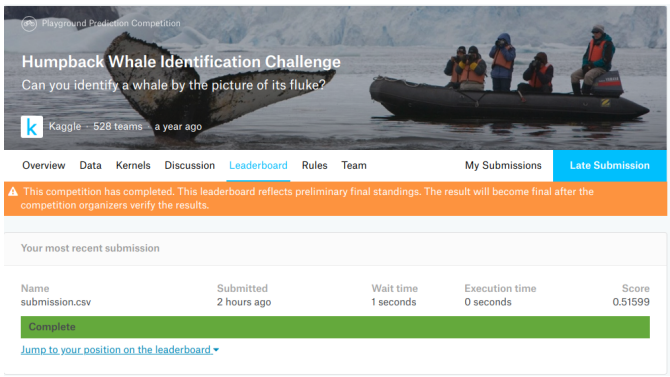

Fig. 9. Competition evaluation

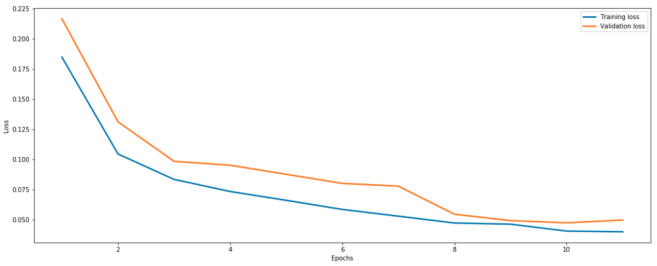

Fig. 10. Model evaluation

There can be applied some heuristics to calculation top 5 classes and including `new_while` class into the response. Top 5 classes are taken as the closest average cosine distance from the test image to the corresponded class.

Along with it, there is an unlabeled `new_whale` class and there are a lot of heuristics how to add this class to the top 5 list of response. The idea was to add `new_whale` to the response if the test image is approximately equidistant from the top 5 classes with some threshold.

Model evaluation result is shown on the Fig. 10.

References

[1] Gregory Koch, Richard Zemel, Ruslan Salakhutdinov. "Siamese neural networks for one-shot image recognition." ICML Deep Learning Workshop. Vol. 2. 2015.

[2] arXiv:1412.6622 [cs.LG]

[3] Laura Leal-Taixe, Cristian Canton-Ferrer, and Konrad Schindler. "Learning by tracking: Siamese CNN for robust target association." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2016.

[4] Maheshwary, Saket, Hemant Misra. "Matching Resumes to Jobs via Deep Siamese Network." Companion of the The Web Conference 2018 on The Web Conference 2018. International World Wide Web Conferences Steering Committee, 2018.

[5] Humpback Whale Identification competition dataset https://www.kaggle.com/c/humpback-whale-identification/data

[6] Takeru Miyato, Shin-ichi Maeda, Masanori Koyama, Ken Nakae1,Shin Ishii "Distributional smoothing with virtual adversarial training" Graduate School of Informatics Kyoto University.

[7] Burkard R.E., Derigs U. (1980) The Linear Sum Assignment Problem. In: Assignment and Matching Problems: Solution Methods with FORTRAN-Programs. Lecture Notes in Economics and Mathematical Systems, vol 184. Springer, Berlin, Heidelberg

[8] Martin G.E. (1982) Affine Transformations. In: Transformation Geometry. Undergraduate Texts in Mathematics. Springer, New York, NY