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

Marios Galanis, Vladimir Kozlow

Department of Civil and Environmental Engineering, Stanford University

Stanford Engineering

Motivation

One of the most common way for scientists to survey marine mammals' populations is to take photographs of the specimens during survey expeditions. The shape of whales' tails and unique markings found in footage can then be used to identify what species and what specimen of whale was encountered. For the past 40 years, 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.

Dataset

Our dataset is drawn from a Kaggle competition called "Humpback Whale Identification[1]. The data for this competition was provided by Happywhale[2], 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. The dataset is extremely imbalance, with 30% of the training examples being of the category "New_whale" and the other classes having 1 to 17 examples in total.

Methods

Our evaluation metric is the Mean Average Precision @ 5 (MAP@5)

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

- U: number of images, P(k) precision at cutoff
- k, n = number predictions per image
- rel(k) indicator function equaling 1 if item k is a correct label, zero otherwise

Simple CNN classifier

Preprocessing:
- RGB → Grayscale
- Remove images with 2 whales manually
- Partition train(80%)/dev(20%) sets

Architecture:
- 3 Convolutional layers:
  - Convolution
  - ReLu activation
  - MaxPooling
- Dense layer with Softmax activation.

Training procedure

The model had a total of ~1.2 million trainable parameters and it was trained for 100 epochs and mini-batch size of 16 with Adam optimizer, cross-entropy loss, and accuracy as a metric.

Model Evaluation

Finally the model was evaluated on the test set but it was found that it was always predicting the same class "new whale" resulting to a 30% accuracy!

Siamese Neural Network[3]

Preprocessing:
- RGB → Grayscale
- Remove images with 2 whales manually
- Bounding box on whale tail and cropping to 384x384 x1.
- Random data augmentation (shear, rotation, mirroring, zoom)

Architecture:
- Encoder: Residual Network having 55 convolutional layers with Batch Normalization, Max Pooling layer, and ReLu activation layer. #parameters: 2,692,096
- Comparator: CNN computing weighted sum of the sum, the product, the norm and the squared difference of two encodings. # parameters: 706 trainable, 0 non-trainable.

Training procedure:
- Load pretrained weights.
- Compute score matrix.
- Creates pairs of examples.
- Start with random pairs
- Increase difficulty solving linear assignment problem and add L2 regularization.

Results

We tested and optimized for different levels of randomness in assigning pairs of images. K represents the level of randomness of the assignment, a value of K=0 representing non-randomized assignment. The prediction were then submitted through Kaggle and our best score was a MAP@5 of 0.82297.

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

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

Discussion

Trying two different types of NN, we were able to compare the strength of complex network architectures such as Siamese NN over simple binary classifiers. Keys to solving the few-shots learning problem of whale identification without over fitting and given the class imbalance were:

- Using pre-trained weights
- Keeping large images
- Increasing training difficulty iteratively
- Balancing training for positive and negative examples

Next Steps and application

Next steps to get better performance at this task could be adversarial training. Although using difficult pairs of whales is already a form of adversarial training, using a GAN could enhance performance and make the model more robust.

However, the best solution is probably to make our model available for public use and collect more data through allowing users to upload their own images. It would provide more test and potentially training data, and also help citizens get more familiar with whale conservation.

References

[1] https://www.kaggle.com/c/humpback-whale-identification\vspace, [2] https://happywhale.com/home , [3] https://www.kaggle.com/martinpiotte

[4] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

Project available at: https://github.com/ultimatemalakas/Whales