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

Marios Galonis, Vladimir Kozlov
Department of Civil and Environmental Engineering, Stanford University

Motivation
One of the most common ways 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 automate 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)
- n: number of images, P(k) precision at cut off
- 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:
- Use pretrained weights.
- Compute score matrix.
- Creates pairs of examples.
- Start with random pairs
- Increase difficulty solving linear assignment problem and add L2 regularisation.

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 0 representing non-randomized assignment. The prediction were then submitted through Kaggle and our best score was a MAP@5 of 0.82297.

Discussion

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 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

Project available at: https://github.com/phillock/HumpWhale