mlnd: Capstone Project

Naruhiko Nakanishi

March 23rd, 2019

[Humpback Whale Identification]

Kaggle Competition is used as a technical domain along with the problem and dataset [1].

1 Definition

Project Overview

After centuries of intense whaling, recovering whale populations still have a hard time adapting to warming oceans and struggle to compete every day with the industrial fishing industry for food.

To aid whale conservation efforts, scientists use photo surveillance systems to monitor ocean activity. They use the shape of whales' tails and unique markings found in footage to identify what species of whale they're analyzing and meticulously log whale pod dynamics and movements. For the past 40 years, most of this work has been done manually by individual scientists, leaving a huge trove of data untapped and underutilized.

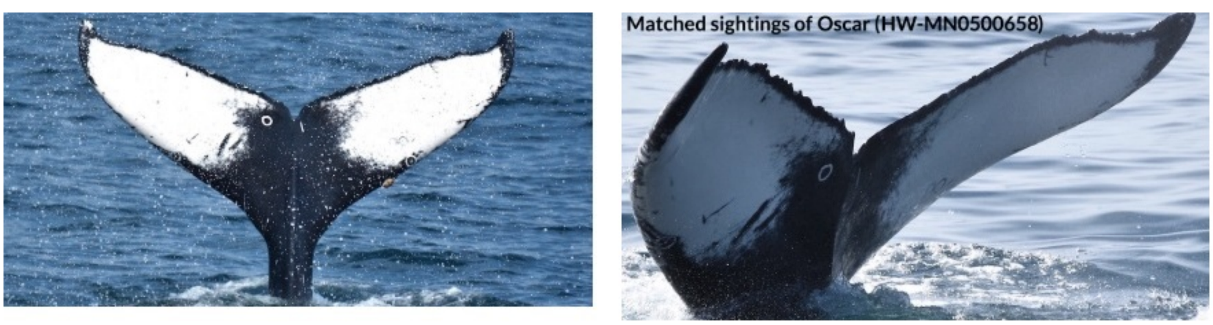

Figure 1: Humpback Whale Identification

Problem Statement

The challenge is to build an algorithm to identify individual whales in images. Happywhale's database is analyzed. Happywhale is a platform that uses image process algorithms to let anyone to submit their whale photo and have it automatically identied [3].

The database is over 25,000 images, gathered from research institutions and public contributors. The contributing is helpful to open rich elds of understanding for marine mammal population dynamics around the globe.

The overall strategy for deriving a model architecture was to use convolutional neural networks (CNNs). CNNs have revolutionized the computational pattern recognition process.

Metrics

The results are are evaluated according to the Mean Average Precision @5 (MAP@5):

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

where U is the number of images, P(k) is the precision at cutoff k, n is the number predictions per image, and rel(k) is an indicator function equaling 1 if the item at rank k is a relevant (correct) label, zero otherwise.

Once a correct label has been scored for an observation, that label is no longer considered relevant for that observation, and additional predictions of that label are skipped in the calculation.

2 Analysis

Data Exploration

This training data contains thousands of images of humpback whale ukes. Individual whales have been identied by researchers and given an Id. The challenge is to predict the whale Id of images in the test set. What makes this such a challenge is that there are only a few examples for each of 3,000+ whale Ids.

File descriptions:

• train.zip - a folder containing the training images

• train.csv - maps the training Image to the appropriate whale Id. Whales that are not predicted to have a label identied in the training data should be labeled as new whale.

• test.zip - a folder containing the test images to predict the whale Id

Exploratory Visualization

As indicated in Figure 2, each Image, has the whale Id. Whales that are not predicted to be one of the labels in the training data is labeled as new whale.

Algorithms and Techniques

Deep learning, specically a convolutional neural network (CNN) which is very effective at finding patterns within images, is used toward the solution.

	Image	Id
0	0000e88ab.jpg	w_f48451c
1	0001f9222.jpg	w_c3d896a
2	00029d126.jpg	w_20df2c5
3	00050a15a.jpg	new_whale
4	0005c1ef8.jpg	new_whale

Figure 2: Humpback Whale data Exploratory

Benchmark

In the benchmark, transfer learning such as MobileNet architecture is used. Transfer learning involves taking a pre-trained neural network and adapting the neural network to a new, dierent data set. MobileNets are based on a streamlined architecture that uses depthwise separable convolutions to build light weight deep neural networks.

3 Methodology

Data Preprocessing

The input data values are normalized to the range [0, 1]. Normalization refers to rescaling real valued numeric attributes into the range 0 and 1. It is useful to scale the input attributes for a model that relies on the magnitude of values.

Implementation

Firstly the dependencies are installed. Keras is used for a deep learning library. Keras is a high level neural networks API.

The second step will be data collection, data exploration and visualisation to understand the fundamental characteristics of the dataset. This training data contains thousands of images of humpback whale ukes. Individual whales have been identied by researchers and given an Id.

The next step is to build the models in Keras for performance and exibility. Deep learning, specically a CNN which is very eective at nding patterns within images, is used toward the solution. A CNN is created to identify individual whales in images.

In the benchmark, transfer learning such as MobileNet architecture is used. Transfer learning involves taking a pre-trained neural network and adapting the neural network to a new, dierent data set.

The next step is Hyperparameter tuning. It is computationally feasible to tune the parameters.

The nal step is to train the model on the entire training set and evaluate the performance. The nal performance will be calculated against the test data set provided by Kaggle. The results are evaluated according to the Mean Average Precision.

Refinement

As mentioned in the Benchmark section, the MobileNet architecture is used as a Benchmark model. Figure 3 shows training accuracy and loss with MobileNet. As indicated in the figure, the MobileNet architecture achieved more than 90 percent accuracy.

In this project, the CNN architecture is used as a model. The initial CNN architecture contains one convolutional layer. Figure 4 shows training accuracy and loss with the initial CNN model. As indicated in the figure, the initial CNN architecture could not achieve more than 40 percent accuracy. The final CNN architecture contains two convolutional layers. Figure 5 shows training accuracy and loss with the final CNN model. As indicated in the figure, the final CNN architecture achieved more than 90 percent accuracy.

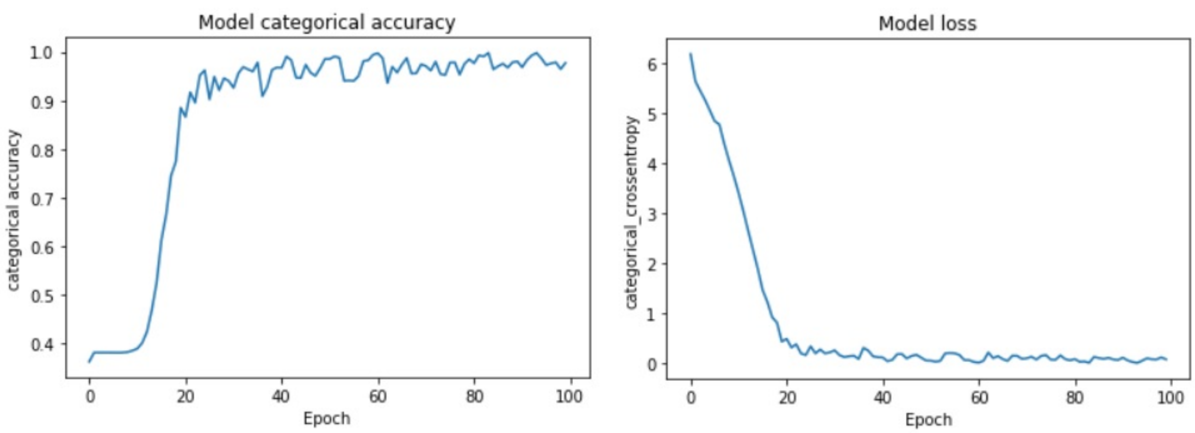

Figure 3: Training accuracy and loss with MobileNet

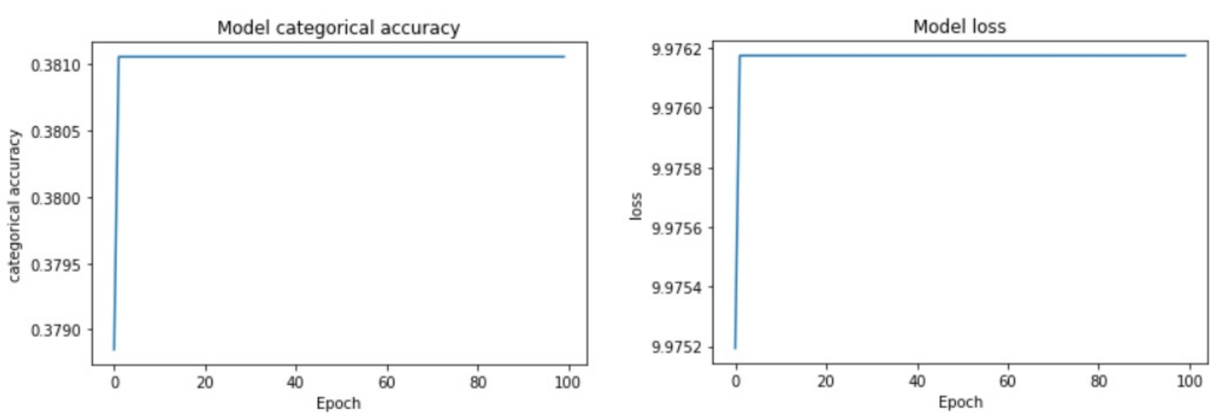

Figure 4: Training accuracy and loss with CNN(1-layer)

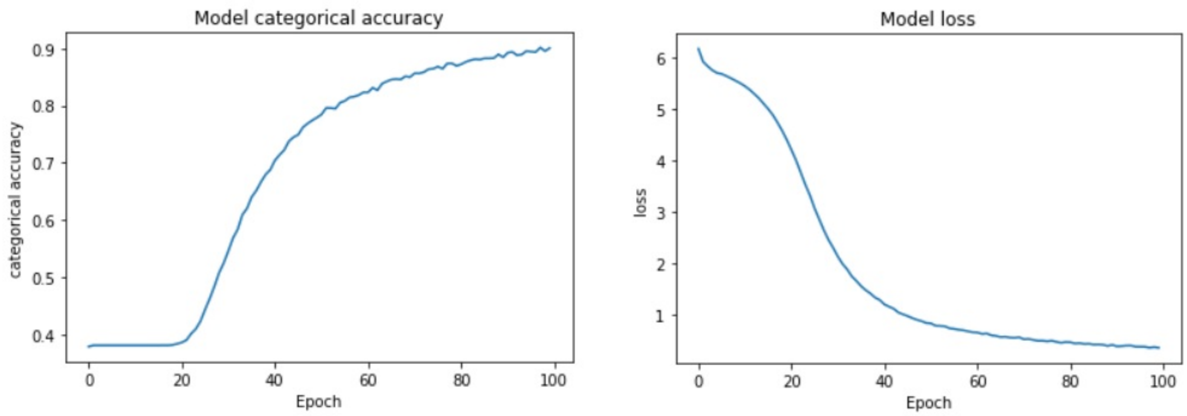

Figure 5: Training accuracy and loss with CNN(2-layers)

4 Results

Model Evaluation and Validation

Figure 6 shows the final CNN model architecture. The architecture consisted of a convolution neural network with the following layers and layer sizes as indicated in the figure.

Table 1 shows hyperparameters and optimizer in the project. As indicated in figure 5, the final CNN architecture achieved more than 90 percent accuracy. It is the same performance as the MobileNet architecture.

Model	Epochs	Batch Size	Optimizer	Learning Rate
CNN	100	128	adam	0.001
MobileNet	100	256	adam	0.001

Table 1: Hyperparameters and Optimizer

Justification

Table 2 shows Kaggle public leaderboard scores. The scores are MAP@5 (Mean Average Precision @5) values as mentioned in the Metrics section. As indicated in the table, the final CNN architecture and hyperparameters performed better compared to the MobileNet architecture.

Model	Public LB Score
CNN	0.29161
MobileNet	0.26544

Table 2: Kaggle Public Leaderboard Score

```
Layer (type)                 Output Shape              Param #
=================================================================
conv0 (Conv2D)               (None, 94, 94, 32)        4736
_________________________________________________________________
bn0 (BatchNormalization)     (None, 94, 94, 32)        128
_________________________________________________________________
activation_1 (Activation)    (None, 94, 94, 32)        0
_________________________________________________________________
max_pool (MaxPooling2D)      (None, 47, 47, 32)        0
_________________________________________________________________
conv1 (Conv2D)               (None, 45, 45, 64)        18496
_________________________________________________________________
activation_2 (Activation)    (None, 45, 45, 64)        0
_________________________________________________________________
avg_pool (AveragePooling2D)  (None, 15, 15, 64)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 14400)             0
_________________________________________________________________
rl (Dense)                   (None, 500)               7200500
_________________________________________________________________
dropout_1 (Dropout)          (None, 500)               0
_________________________________________________________________
sm (Dense)                   (None, 5005)              2507505
=================================================================
Total params: 9,731,365
Trainable params: 9,731,301
Non-trainable params: 64
_________________________________________________________________
```

Figure 6: Final CNN model architecture

5 Conclusion

Free-Form Visualization

As shown in Table 2, Kaggle public leaderboard score with CNN is 0.29161 and the score with MobileNet is 0.26544. It is indicated that the final CNN model performed better than Benchmark model.

Reflection

An algorithm is built to identify individual whales in images. The final model architecture consisted of a convolution neural network with the followinglayers and layer sizes as shown in Figure 6.

Kaggle public leaderboard score is MAP@5 (Mean Average Precision @5) value. The score with the final CNN model architecture is 0.29161 and the score with MobileNet is 0.26544. So it is found that the final CNN model performed better than Benchmark model.

Improvement

MAP@5 value in the project might be improved using following methods.

• Use ensemble learning [2]. An approach to reducing the variance of neural network models is to train multiple models instead of a single model and to combine the predictions from these models. This is called ensemble learning

• Use data augmentation [4]. To get more data, making minor alterations are needed to the existing dataset. Neural network would think that minor changes such as flips or translations or rotations are distinct images.

References

[1] https://www.kaggle.com/c/humpback-whale-identification.

[2] Jason Brownlee. https://machinelearningmastery.com/ensemble-methods-for-deep-learning-neural-networks.

[3] Happywhale. https://happywhale.com/home.

[4] Bharath Raj. https://medium.com/nanonets/how-to-use-deep-learning-when-you-have-limited-data-part-2-data-augmentation-c26971dc8ced.