Machine Learning Engineer Nanodegree

Capstone Project

Prasand Kumar

October 30th, 2017

Domain Background

Nearly half of the world depends on seafood for their main source of protein. In the Western and Central Pacific, where 60% of the world's tuna is caught, illegal, unreported, and unregulated fishing practices are threatening marine ecosystems, global seafood supplies and local livelihoods. The Nature Conservancy is working with local, regional and global partners to preserve this fishery for the future.

Currently, the Conservancy is looking to the future by using cameras to dramatically scale the monitoring of fishing activities to fill critical science and compliance monitoring data gaps. Although these electronic monitoring systems work well and are ready for wider deployment, the amount of raw data produced is cumbersome and expensive to process manually.

The Conservancy is inviting the Kaggle community to develop algorithms to automatically detect and classify species of tunas, sharks and more that fishing boats catch, which will accelerate the video review process. This will have a positive impact on conservation and our planet.

The aim of this project is to build a Convolutional Neural Network (CNN) that can classify the distinct species of fishes. To tackle the computational constraints, we will be using transfer learning technique.

Problem Statement

In this project, The Nature Conservancy asks you to help them detect which species of fish appears on a fishing boat, based on images captured from boat cameras of various angles.

The goal is to predict the likelihood of fish species in each picture.

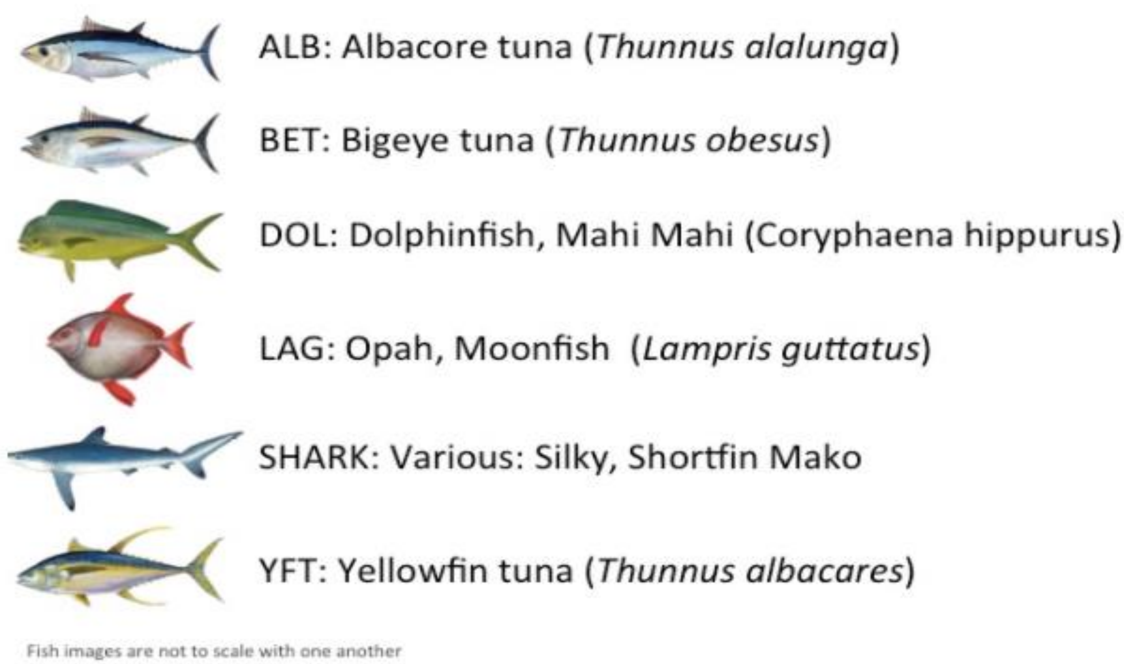
Eight target categories are available in this dataset: Albacore tuna, Bigeye tuna, Yellowfin tuna, Mahi Mahi, Opah, Sharks, Other (meaning that there are fish present but not in the above categories), and No Fish (meaning that no fish is in the picture).

Dataset and Inputs

The dataset was compiled by The Nature Conservancy in partnership with Satlink, Archipelago Marine Research, the Pacific Community, the Solomon Islands Ministry of Fisheries and Marine Resources, the Australia Fisheries Management Authority, and the governments of New Caledonia and Palau.

We used a dataset of 3777 images in training set classified into 8 labels, published by Nature Conservancy on Kaggle. The labels included six species of fish as well as one “No fish” and one “Other” label. The test set has 1000 images.

From the outset, the data presented several key challenges. The images were all varied sizes and had all been taken at various times of day. Some contained more than one species of fish. The dataset was also small, containing only a few thousand images. Each image has only one fish category, except that there are sometimes very small fish in the pictures that are used as bait. The Nature Conservancy has also provided a visualization of labels



Solution Statement

As deep learning is very helpful in image classification, we use transfer learning to train the Convolutional neural network (CNN) which in-turn used to classify the images with fish to their respective labels.

Transfer Learning - A neural network is trained on data. This network gains knowledge from this data, which is compiled as “weights” of the network. These weights can be extracted and then transferred to any other neural network. Instead of training the other neural network from scratch, we “transfer” the learned features. Few such networks are RESNET, InceptionV3, VGG16.

Benchmark model

K-nearest neighbor classification - The model was trained on the color histogram of the images with Euclidean distance as a distance metric. The model yielded 1.70 log loss in the submission leaderboard.

With CNN implementation, our model would perform better than the KNN model. Also, due to computational constraints we can aim for a log loss value less than 1.70 log loss.

Evaluation Metrics

The model is evaluated using the multi-class logarithmic loss. Each image has been labeled with one true class. For each image, you must submit a set of predicted probabilities (one for every image). The formula is then,

$$logloss = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{ij}\log(p_{ij}),$$

where N is the number of images in the test set, M is the number of image class labels, log is the natural logarithm, yij is 1 if observation i belongs to class j and 0 otherwise, and pij is the predicted probability that observation i belongs to class j.

The submitted probabilities for a given image are not required to sum to one because they are rescaled prior to being scored (each row is divided by the row sum). To avoid the extremes of the log function, predicted probabilities are replaced with

$$max(min(p, 1-10^{-15}), 10^{-15}).$$

Problem Design

Programming Language - Python 3.5
Libraries – Keras, Tensorflow, OpenCv, Scikitlearn

Workflow

- Fixing a base line model using KNN.
- Building a CNN model from scratch to compare it with the models created using Transfer Learning Techniques.
- From the images the features are extracted using the pre-trained network. And, 8 output neurons are present in the final layer to get predictions.

- The CNN model can be tuned to improve its performance by multi-folds. The two primary classes of refinements to improve the CNN model performance are
    - Neural Network Architecture – Number of Filters, Shape of Filter, Depth of Output Volume and Types of pooling.
    - Training Hyper-Parameters – Length of Training (Number of Epochs), Initial Learning Rate, Decay Rate and Dropout Keep Rate.

---

https://www.kaggle.com/c/the-nature-conservancy-fisheries-monitoring

https://www.analyticsvidhya.com/blog/2017/06/transfer-learning-the-art-of-fine-tuning-a-pre-trained-model/

http://cs229.stanford.edu/proj2016/report/FrostGeislerMahajan-MonitoringIllegalFishingThroughImageClassification-report.pdf