Introduction

The Nature Conservancy, an environmental non-profit, is working with several Pacific Island nations and a big tuna fishing company to more easily count and identify fish caught at sea using cutting edge technology. The goal is to use deep learning to help fishermen reduce the number of protected animals like species of sharks and tunas that are accidentally caught along with the tuna. The Nature Conservancy hopes that the program could prevent overfishing and help threatened and endangered sea life recover without putting fishermen out of work…

The Nature Conservancy has installed nearly a dozen boats with electronic monitoring systems that include a set of cameras, sensors, and GPS devices. The system can record most of what takes place on board to prove that the operators did nothing illegal and to back up any compliance data that they must present to officials when they deliver their catch.

Project overview

In this project, we will be training a deep learning model to automatically detect and classify species of tunas, sharks and more that fishing boats catch, which will accelerate the recording review process. The approach is to have an algorithm embedded into a fully automated software tool that workers could use in their fishing operations. The algorithm will help fishermen detect which species of fish appears on a fishing boat, based on images captured from boat cameras of various angles.

Problem Statement

In this project, we are going to train a machine learning model 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). Each image has only one fish category, except that there are sometimes very small fish in the pictures that are used as bait.

We’ll first create a synthetic dataset and use that to train our model. We’ll then validate the model synthetic test data. We will create multiple models using grid search for epochs, batch size, and optimizer.

Data Acquisition

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.

The train and test datasets is downloaded locally from Kaggle website. Then, the data is uploaded into Docker container. The train data is then loaded into the /DATA/TRAIN/ folder. It will be stored accordingly:

Albacore tuna: /DATA/TRAIN/ALB/

Bigeye tuna:/DATA/TRAIN/BET/

Opah:/DATA/TRAIN/DOL/

Mahi Mahi: /DATA/TRAIN/LAG/

Sharks: /DATA/TRAIN/SHARK/

Yellowfin tuna: /DATA/TRAIN/YFT/

Other fish than the above: /DATA/TRAIN/OTHER/

No fish in the picture: /DATA/TRAIN/Nof/

The test data is loaded in /DATA/TEST/

Please refer the dataload.txt script in the documentation.

The dataset has the following:

1719 photos of ALB, Albacore tuna:

200 photos of BET, Bigeye tuna

117 photos of DOL, Opah:

67 photos of LAG, Mahi Mahi

176 photos of SHARK, Sharks

734 photos of YFT, Yellowfin tuna

299 photos of OTHER, Other fish than the above

465 photos of Nof, No fish in the picture:

Faster review and more reliable data will enable countries to reallocate human capital to management and enforcement activities which will have a positive impact on conservation and our planet.

digits within a natural image.

Your 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). Each image has only one fish category, except that there are sometimes very small fish in the pictures that are used as bait.

The science here can be applied to a large scale problem, such

as recognising house numbers from google street view images. We noticed that a

housing number is a sequence of digits. To help understand this, we can build a

classifier that understands how to classify digits and use a convolution neural networks

to scan through an image, returning only the parts that it believes contains a digit. Our

primary dataset for this study is public street view dataset available

develop algorithms to automatically detect and classify species of tunas, sharks and more that fishing boats catch, which will accelerate the video review process. Faster review and more reliable data will enable countries to reallocate human capital to management and enforcement activities which will have a positive impact on conservation and our planet.

Machine learning has the ability to transform what we know about our oceans and how we manage them. You can be part of the solution.

Model Creation:

I will be creating two CNN models. A base model with the following major components CNN model:

1- Convolutional layer with 60 feature maps of size 5x5.

2- Pooling layer taking the max over 2x2 patches.

3- Convolutional layer with 12 feature maps of size 3x3.

4- Pooling layer taking the max over 2x2 patches.

5- Dropout layer with a probability of 50%.

6- Flatten layer.

7- Fully connected layer with 128 neurons and rectifier activation.

8- Dropout layer with a probability of 50%.

9- Fully connected layer with 50 neurons and rectifier activation.

10- Dropout layer with a probability of 50%.

11- Output layer.

Model Selection:

To improve the algorithm and select the best perming model, I will be tuning the below areas by Grid-Searching the hyper parameters when applicable:

1- Network Topology: I will be changing the network topology by using different number of layers and neurons.

2- Learning Rate: I will experiment with very small learning rates and large rates. I will try adding momentum term; then change the learning rate.

3- Activation Functions: I will be experimenting using the different activation function sigmoid,tanh, relu, then a softmax, linear or sigmoid on the output layer

4- Batches and Epochs: The batch size defines the gradient and how often to update weights. An epoch is the entire training data exposed to the network, batch-by-batch. I will be experimenting running small batch sizes with large epoch size.

5- Regularization: regularization is a great approach to curb overfitting the training data. I will be using the dropout regularization technique. Dropout randomly skips neurons during training, forcing others in the layer to pick up the slack. I will be experimenting with dropout in the input, hidden, and output layers.

6- Optimization and Loss: There are many optimization methods that offer more parameters, more complexity and faster convergence. However, for our classification problem, we will be mainly experimenting with the followings: adam and RMSprop

Based on the above tuning strategy, the model with the best performance is model One with the following characteristics:

Performing

Using model One, the below is the results of Grid-searching the hyper parameters: