Lab note

Yuree Oh

Pomfret School, Connecticut, USA

Alex Oh	Date	12/19

Development of Oil Spill Detection Technology using Video Image Processing by Deep Neural Network

Used image augmentation to increase the number of data in the dataset. Image augmentation alters data so that the model is able to detect oil spills no matter what angle, color etc and increases the number of data without repeats so that it ensures the model is accurate.

To use image augmentation, ImageDataGenerator from tensorflow is used and the image is resized from 64,64 to 150,150.

To reiterate, the structure of the model is the following:
Using augmented data divided into oil spill and no spill, the model does fine tuning on GoogleNet, pre-trained by the data in ImageNet (which has multiple categories while we only need two categories- oil spill and no spill). Then, using a sigmoid activation function (RELU is used inside GoogleNet) the weight values are changed.

Epoch 16/50
30/30 [==============================] - 200s 7s/step - loss: 0.1906 - accuracy: 0.9460
Epoch 17/50
30/30 [==============================] - 200s 7s/step - loss: 0.1760 - accuracy: 0.9386
Epoch 18/50
30/30 [==============================] - 200s 7s/step - loss: 0.3487 - accuracy: 0.8644
Epoch 19/50
30/30 [==============================] - 203s 7s/step - loss: 0.3189 - accuracy: 0.8781
Epoch 20/50
30/30 [==============================] - 204s 7s/step - loss: 0.2266 - accuracy: 0.9208

Tracking epoch with the highest accuracy
Finishing the method part of the research paper

Alex Oh	Date	12/21

Development of Oil Spill Detection Technology using Video Image Processing by Deep Neural Network

I learned about multiple optimizers, their functions, and how they operate. Optimizers evaluate multiple positrons and re-calculate weight values. It has the general equation of:
$w_{t+1} = w_t - \eta \times \nabla L(w_t)$,
where w is the weight (gradient) of a positron,
η is the learning rate (how much will the program learn)
$L(w_t)$ is what direction the program will learn.

Gradient descent is an optimizer that calculates the loss function of all data sets and eventually the w (weight) value. This is basically how a batch gradient descent works. However, it is incredibly slow and very likely to fall into local minimas.

Stochastic gradient descent, a method derived from batch gradient descent, calculates the loss function of selectively random data sets. The number of data sets the program randomly selects to evaluate weight values must be set by the operator. Despite the fact that it relies heavily on the number of data sets chosen and is sometimes inaccurate, it is relatively fast compared to BGD as it is able to change its weight value more by going over more steps at a time. Also, it has a probability of finding the global minimum even when it comes across local minimas.

The momentum optimizer, like its namesake, utilizes previously calculated gradient (weight) values to calculate current gradient. The v_{t-1}value is calculated by an equation that uses previous gradient values and a momentum coefficient (hyperparameter set by operator). Basically, $v_{t-1}$is how much the previous step moved.

$$v_t = \gamma v_{t-1} + \eta \nabla_\theta J(\theta)$$

$$\theta = \theta - v_t$$

$$v_t = \eta \nabla_\theta J(\theta)_t + \gamma \eta \nabla_\theta J(\theta)_{t-1} + \gamma^2 \eta \nabla_\theta J(\theta)_{t-2} + \dots$$

The adagrad optimizer removes fixed learning rate and replaces it with a formula of variables. It is unique as it gives change not only to the gradient but also to the learning rate, thus rendering the learning rate no longer constant. $G_t$contains the sum of previous gradient values until t.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt{G_t + \epsilon}} \cdot \nabla_\theta J(\theta_t)$$

CNN, the danger of oil leaks and its effect on marine ecosystem

Alex Oh	Date	12/23

Development of Oil Spill Detection Technology using Video Image Processing by Deep Neural Network

I learned about CNN, convoluted neural networks, which are used to extract datasets into numerical values. Common examples that use CNN are Pinterest, Adobe DeepFont (which compares user's handwriting to fonts), and WePod. In order to use CNN, there must be two or more datasets. In our case, there has to be a dataset (image) of a clear ocean (without oil) and one with an oil leak.

A channel is an area or a dimension. Original images contain 3 channels- red, green, and blue. However, the data that the channels contain are not pixel values. Pixel values are too limited and contain little data.

CNN performs matrix convolution with filters on these channels. Matrix convolution is a method of multiplying matrices. A filter acts like a stamp. It initially multiplies its values in (0,0) with that in filter (0,0). Once all values in the channel between the parameters of the filter are multiplied, the products are added. The result becomes the (0,0) value of the feature map. As the values within the filter are weights, they are frequently altered for accuracy.

Using the MNIST dataset and CoLab, I made a deep learning program that identifies handwritten numbers with approximately 97% accuracy. Then, I repeated using CNN to extract data. However, as the MNIST dataset contains very little ambiguous data, there was almost no difference in the accuracy of the two programs.

Using datasets that contain ambiguous and non-numerical data

Alex Oh	Date	12/26

Development of Oil Spill Detection Technology using Video Image Processing by Deep Neural Network

Coded the model for the deep learning project. I learned about different libraries in python (in colab) such as os, glob, PIL, etc that are used to extract data from datasets containing images.

Using colab, the program imports images, resizes them into (64,64), and converts them into data that shows the RGB value for each pixel. The model then uses CNN, an ADAM optimizer and a binary crossentropy (as there are only 2 values possible - 0 or 1) loss function. The model uses 5 epochs.

The dataset used today came from google datasets, which redirected me to kaggle, a website where users share datasets. I used Damingo Lavi's *Oil Spill vs No Spill* dataset. However, because the dataset had very little images and had data for training and testing overlapping, the model was not very accurate (75%). While it did show a 100% accuracy for testing data, it was later found that the data overlapped and thus caused an abnormal accuracy.

going deeper with convolutions – read and take notes
Plan what to write in the Method part of the research paper

Alex Oh	Date	12/28

Development of Oil Spill Detection Technology using Video Image Processing by Deep Neural Network

Once Deep Learning program(s) gained attention by having an error rate less than that of a human being, Google began to invest in Deep learning tools. Initially, Google's idea was to create a program with layers as deep as possible. However, this caused multiple issues as it is an NP Hard problem, with computations increasing not linearly but exponentially. This caused the program to have a colossal amount of computation. Also, there was an overfitting problem.

Therefore, Google sought out ways to make the program more efficient. They attempted to utilize the sparsely connected architecture, proposed in 2014. The Sparsely Connected Architecture works with nodes that have the least amount of common output nodes connected to in two neighboring pair of nodes. However, this proved to be extremely inefficient, as Densely connected architecture are efficient.

This hence caused the development of the Google net architecture, comprised of multiple Bottleneck features. The architecture resizes images not in series but in parallel. In series, as the layers deepen, the input would be from features, causing a problem. Also, as Google found out that two 3x3 layers are much more efficient than a 5x5 layer, the architecture uses two 3x3 instead of 5x5.

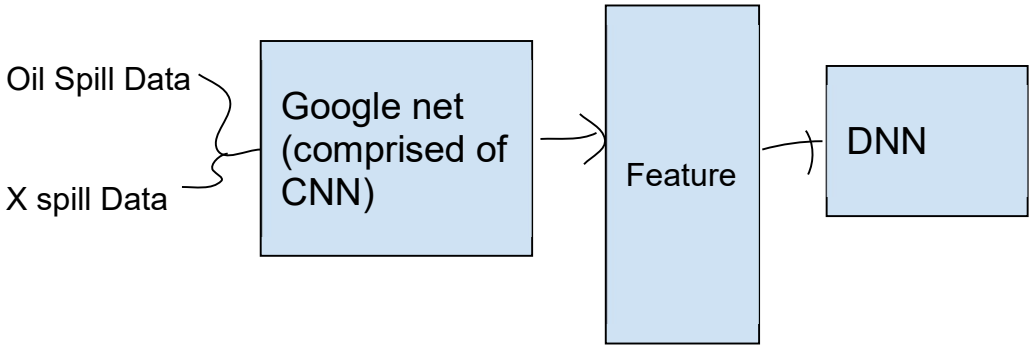
Used the ImagerGeek chrome extension to crawl Nospill and Oilspill data in Google, then manually removed unneeded data (images that does not contain the sea) and divided them into oil spill and no spill data.

Alex Oh	Date	12/30

Development of Oil Spill Detection Technology using Video Image Processing by Deep Neural Network

Learned about fine tuning. Fine tuning is a process where the parameter is delicately adjusted. It is used when an external/pre learned model is used to extract features from data sets. An example of a pre learned model would be Google net. Google net has thousands of categories and a colossal data set. However, for our DNN, we have only 2 categories and very little data (about 700-800), which calls for a need for adjustment in parameters.

Here is a drawing to clarify how fine tuning and pre learned models are used.



(activation function, loss functions, and optimization functions are omitted in the diagram)

Used google net in the program, fine tuned it and edited the code to fit our purpose.
Reached
1s 65ms/step - loss: 0.2412 - accuracy: 0.9184 - val_loss: 0.3089 - val_accuracy: 0.8750
in 50 epochs

creating more image datasets via augmentation (need about 2000 total)

Alex Oh	Date	1/2
Development of Oil Spill Detection Technology using Video Image Processing by Deep Neural Network		
Collected image data by using various keywords (eg. Mingbulak oil spill) Also attempted using different languages - resulted in a success! Turns out, there are different images/cases that people refer to 'oil spills' in their language. Hopefully this makes way for more collectable images.		
Wrote a small portion of the research paper and got feedback. As the model is not fully complete (due to lack of image data), some parts could not be written (eg. results)		
Collect data		