Machine Learning Engineer Nanodegree

Capstone Proposal

Identifying ships from Space Borne Photography

Kommaddi Tharun

February 23rd, 2019

Definition

Domain Background

Space Borne Photography (Satellite Imagery) provides various aspects including agriculture, meteorology, energy, and cartography. New commercial imagery providers, such as Planet are using of small satellites to capture Earth images every day.

This flood of new imagery is outgrowing the ability for organizations to manually look at each image that gets captured, and there is a need for machine learning and computer vision algorithms to help automate the analysis process.

The aim of this project is to identify the task of detecting the position of large ships in satellite images. Automating this process can be applied to many issues including monitoring port activities and supply chain analysis.

Satellite imagery used to build this dataset is made available through Planet's Open California dataset, which is openly licensed. As such, this dataset is also available under the same CC-BY-SA license. Users can sign up for a free Planet account to search, view, and download their imagery and gain access to their API.

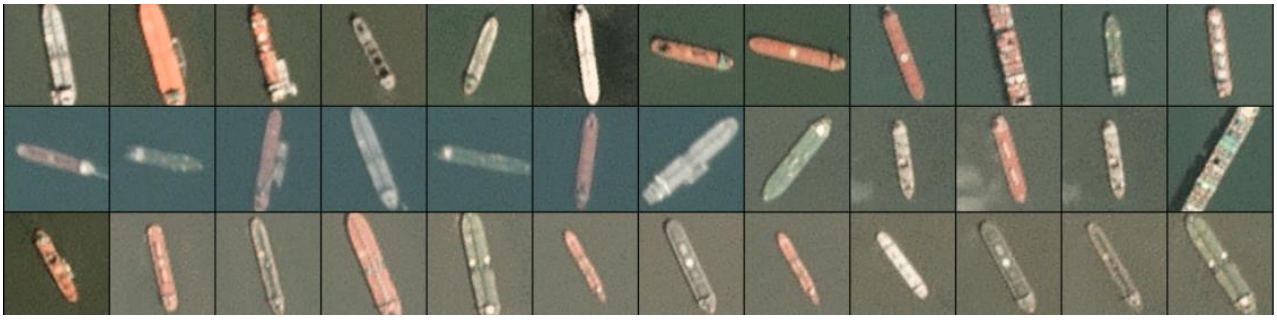
Problem Statement

The aim of the project is to identify ships in a satellite image. The satellite image contains huge amount of information, I would like to identify where the ships are located in the image. In order to do this I would like to use Convolution Neural Networks(CNN) to classify the visualized data and create a model for detecting ships.

Datasets and Inputs

The dataset consists of image chips extracted from Planet satellite imagery collected over the San Francisco Bay and San Pedro Bay areas of California. It includes 4000 80x80 RGB images labeled with either a "ship" or "no-ship" classification. Image chips were derived from PlanetScope full-frame visual scene products, which are orthorectified to a 3 meter pixel size. The data set is downloaded from kaggle.

The dataset is distributed as a JSON formatted text file. The loaded object contains data, label, scene_ids, and location lists.The "ship" class includes 1000 images. Images in this class are near-centered on the body of a single ship. Ships of different sizes, orientations, and atmospheric collection conditions are included. Example images from this class are shown below.



The "no-ship" class includes 3000 images. A third of these are a random sampling of different landcover features - water, vegetion, bare earth, buildings, etc. - that do not include any portion of an ship. The next third are "partial ships" that contain only a portion of an ship, but not enough to meet the full definition of the "ship" class. The last third are images that have previously been mislabeled by machine learning models, typically caused by bright pixels or strong linear features. Example images from this class are shown below.

ref: https://www.kaggle.com/rhammell/ships-in-satellite-imagery/home.

Solution Statement

The solution I would like to propose is usage of Convolution Neural Networks. I will use "Softmax" function for predicting the probabilities of each classes. "Relu" activation for each layer for increasing accuracy and as well as overcome overfitting of data. I would like to use "sgd" optimizer for the validation and fitting of data. I the process I would like to tune some features like:

- Learning rate
- No. of layers
- Various Optimizers
- Different features in data augmentation

Benchmark model

For this benchmark model I would like to create a basic Convolution Neural Network model with minimum layers available.

METRICS:

As the dataset contains unbalanced classes accuracy might not be a good choice. So I would take f1 score as my metric. Which is harmonic mean of precision and recall.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$= \frac{True\ Positive}{Total\ Predicted\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

$$= \frac{True\ Positive}{Total\ Actual\ Positive}$$

$$F1 = 2 \times \frac{Precision * Recall}{Precision + Recall}$$

During development, a validation set is used to validate the model from the data.

The dataset is augmented so that we can predict more accurately. And finally "categorical_cross validation" is used for loss metric. I would like to use 'Adam' optimizer for optimizing the model and accuracy as metric.

Project Design:

The process I would like to do is like this:

Pre-processing data:

The given data is in the form of JSON format. First I would like to extract the data from the JSON file. The pixel value data for each 80x80 RGB image is stored as a list of 19200 integers within the data list. So I have to convert the pixel value data into image and store all the images in to a list. As every image is of same size there is no need for further altering of the images.

I would like to check whether the image conversion took place correctly or not. Then I would like to split the whole data into training and testing using 'train_test_split' and normalize the data.

Creating Benchmark model:

Then I will create a benchmark model with minimal layers and train the data and check for the accuracy. This is the accuracy I would like to surpass.

Final model:

Here I will create a deeper model with different layers of CNN. Augmentation of data is also done in here for greater accuracy. I will tune and check with some of the hyper parameters for extracting best model.

Testing:

Testing the model for accuracy and evaluating the model.

Sample Image from the dataset: