Machine Learning Engineer Nanodegree

Capstone Proposal

Jeongwhan Choi

January 15th, 2018

Proposal

Domain Background

Deep learning belongs to neural model paradigm in three paradigms of machine learning. It is the latest version of [1]connectionism that has been going on since the 1950s. In connectionism, problems that were previously considered obstacles have been solved one by one, and it has recently become the most destructive technology field in artificial intelligence.[^ 2]In 1979, Kunihiko Fukushima first published Neocogitron model applying neurophysiological theory to the pearl neural network. This model, inspired by Torsten Wiesel's award-winning Nobel Prize in neurophysiology, was the beginning of Convolutional Neural Network(CNN). CNN, which imitates human visual cognition processes, is inherently used in a unique way in the field of computer vision. Convolution technology, which shows excellent performance in extracting desired features from various types of data, is used in various fields such as image processing and speech recognition.

The reason for using the convolution technique in image processing and signal analysis is to separate and extract features contained in signals such as original image or sound wave. So, I need to access via CNN to identify whether the monitored object is a ship or an iceberg.

Problem Statement

Drifting icebergs present threats to navigation and activities in areas such as offshore of the Arctic Ocean. As shown in the picture below, the iceberg of this appearance may not be distinguishable from the ship, and it can give a great impact to the passing ship.

[2]

[2]

Currently, many institutions and companies use aerial reconnaissance and shore-based support to monitor environmental conditions and assess risks from icebergs. However, in remote areas with particularly harsh weather, these methods are not feasible, and the only viable monitoring option is via satellite.

They are responsible for maritime safety using SAR for ship and iceberg monitoring. Ships and icebergs can be detected because the response of backscattering is stronger than the surrounding open water.[3] However, the distinction between ship and iceberg is still difficult. Machine learning can be used to solve this problem. The goal is to create an image classification model that finds icebergs among SAR images collected by satellites. We will use the deep learning with CNN for this. As mentioned in the domain background, CNN is excellent for separating and extracting the characteristics of images.

The CNN model receives the flattened glacier image data as input and the resulting value will indicate whether the image is a iceberg. The result will indicate that the iceberg in case of 1 in a range of from 0 to 1.

Datasets and Inputs

Statoil, a global energy company that operates globally, has worked closely with companies such as C-CORE. C-CORE has been using satellite data for over 30 years and has built a computer vision based surveillance system. To keep the operations safe and efficient, Statoil provided the data sets to Kaggle in order to gain a fresh perspective on how to use machine learning to more accurately detect and differentiate threat icebergs as early as possible.

train.json, test.json

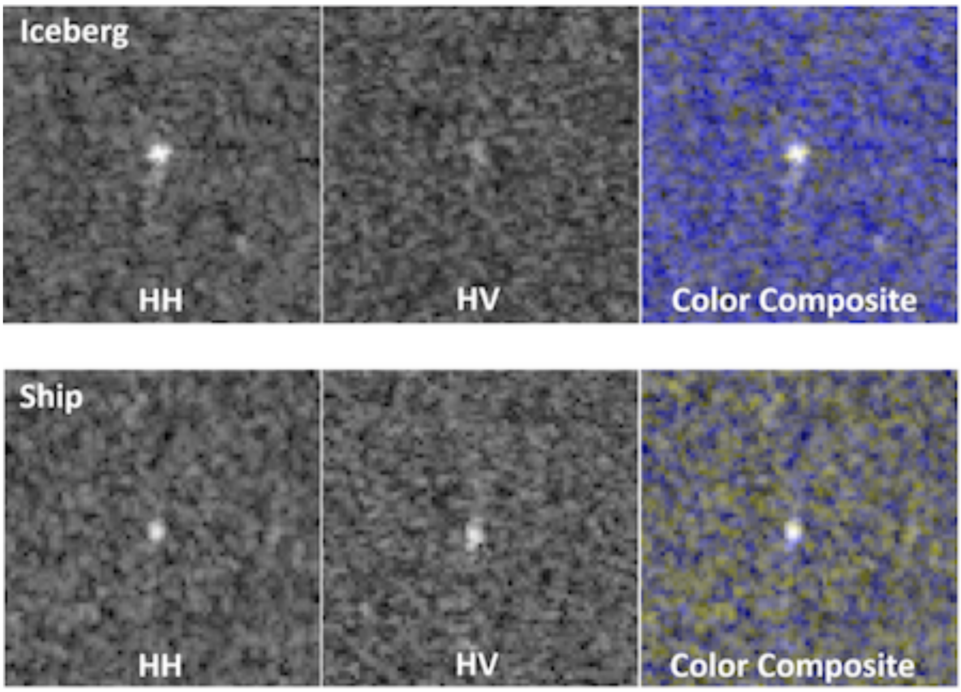
The data (`train.json`, `test.json`) is presented in `json` format. The files consist of a list of images, and for each image, you can find the following fields:

- **id** - the id of the image
- **band_1, band_2** - the flattened image data. Each band has 75x75 pixel values in the list, so the list has 5625 elements. Note that these values are not the normal non-negative integers in image files since they have physical meanings - these are float numbers with unit being dB. Band 1 and Band 2 are signals characterized by radar backscatter produced from different polarizations at a particular incidence angle. The polarizations correspond to HH (transmit/receive horizontally) and HV (transmit horizontally and receive vertically).
- **inc_angle** - the incidence angle of which the image was taken. Note that this field has missing data marked as "na", and those images with "na" incidence angles are all in the training data to prevent leakage.
- **is_iceberg** - the target variable, set to 1 if it is an iceberg, and 0 if it is a ship. This field only exists in `train.json`.

The satellite used to detect icebergs is located 600 km from the Earth using a radar that records the echo and returns the signal of the object, and the data is converted into an image. The C-band radar operates at frequencies that can be seen through darkness, rain, clouds and fog. The echoes of different objects are recorded and converted into images. Objects appear to be bright places because they require more radar energy than they do, but strong echoes can occur in solid lands, islands, and sea ice as well as in icebergs and boats. The energy returned to the radar is called backscattering. Many things include wind and rear spawning. High winds produce a bright background and low winds produce a darker background. The Sentinel-1 satellite is a radar looking at the sides. In other words, you see the image area at an angle (incident angle). In general, the ocean background darkens at high incidence. You also need to consider radar polarization, the way radar transmits and receives energy. Advanced radars like Sentinel-1 can transmit and receive in horizontal and vertical planes. You can use this to get a double-polarized image.

Here we have data in two channels: HH (Transmit / Receive) and HV (Transmit and Receive Horizontally). This can play an important role in categorizing other targets as they tend to use energy in different ways. Every image is a 75x75 image with two bands, and the angle at which the image is taken is indicated by inc_angel.

Here is the sample images.[4]



Solution Statement

To solve the problem presented in the Problem Statement, I will create an image classifier that can identify the ship and the iceberg by deep learning using Convolutional Neural Network. To do this, we need to train CNN by inputting data from `train.json` file. The result will then be represented by a value between 0 and 1, which is the probability that the image is iceberg.

Some of the incident angle from the satellite are unknown and marked as "na". Replace these na with 0 and find the indices where the incident angle is >0. In other words, I do not plan to use `inc_angle`. However, this field can be used if a better performance model is needed.

Benchmark Model

This problem is provided by the Kaggle contest, so there are no benchmark models yet. After carefully review, I chose the model proposed by [DeveshMaheshwari] as a benchmark model. For the first time, this model was generated by training the average of two band_1 and band_2 fields before CNN training. In addition, the model added several dropout layers between the convolutional layers and reduced the number of data using the MaxPooling layers. This model uses the sigmoid layer as the last layer, and it is appropriate to use the sigmoid layer because it is a binary classification that identifies whether the input image is a ship or an iceberg.

Evaluation Metrics

The evaluation metric used in accuracy.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{CGT}}$$

- SUT—System Under Test
- TP—True positive, an object present in the GT and the SUT (also called Correct Detection or

one-to-one match)

- TN—True negative, an element present in neither the GT nor the SUT
- CGT—Complete Ground Truth is the total number of GT objects.

Accuracy = (TP+TN)/CGT, the sum of the true positives and the true negatives relative to the total number of GT objects. This is a measure of the actual performance of the system with regard to both correctly detecting and correctly rejecting targets.[5]

The simple accuracy metrics used as the training dataset has 53.05% "ships" and 46.94% "icebergs ". The classes are close to balanced.

If the training data classes are imbalanced, then F1-score metric is better but, this dataset is almost balanced, so using Accuracy as the metrics.

Project Design

Environment

- macOS High Sierra
- Python 3.54
- Anaconda 4.44
- Jupyter Notebook
- Keras
- TensorFlow

Workflow

1. Understand domain knowledge.
    - SAR image?
    - The two channels: HH and HV?
2. Analyze the input data.
    - Visualize the input data as the image in 2D or 3D.
    - Since the data is preseted in `json` format, it needs to be converted back to 75x75 pixels.
3. Generate the training data.
    - Extract all two bands and take average of them as 3rd channel to create a 3-channel RGB equivalent.
4. CNN model.
    - Use the benchmark model.
    - Use transfer learning with VGG16.

- Compare the case of using transfer learning and the case of using the benchmark model.

5. Write the report.

Reference

1. Connectionism. In Wikipedia. Retrieved January 15, 2018, from https://en.wikipedia.org/wiki/Connectionism↩

2. Alexandrov, V., Volkov, V., Sandven, S., & Kloster, K. (2008). Detection of Arctic icebergs on the basis of satellite SAR. In *The 2 intern Workshop on advances in SAR Oceanography from Envisat and ERS missions. ESA-ESRIN. 21-25 January*.↩↩

3. C. Bentes et al. (2016). Ship-Iceberg Discrimination with Convolutional Neural Network in High Resolution SAR Images.↩

4. Background, Statoil/C-CORE Iceberg Classifier Challenge. In Kaggle. Retrieved January 15, 2018, from https://www.kaggle.com/c/statoil-iceberg-classifier-challenge#Background↩

5. Godil, A., Bostleman, R., Shackleford, W., Hong, T., & Shneier, M. (2014). *Performance Metrics for Evaluationg Object and Human Detection and Tracking Systems*. US Department of Commerce, National Institute of Standards and Technology.↩