Machine Learning Engineer Nanodegree

Capstone Proposal

Kaiyuan Hu October 11, 2017

Kaggle competition: Statoil/C-CORE Iceberg Classifier Challenge

Domain Background

Statoil/C-CORE Iceberg Classifier Challenge is an ongoing competition state on Kaggle.

Statoil worked closely with *C-CORE* that use remote sensing system to detect icebergs. The remote sensing system is housed on satellites over 600 kilometers above the earth. The satellites provide two channels image information: HH(transmit/receive horizontally) and HV(transmit horizontally and receive vertically). They want to use those image data to find a model can classify those icebergs from ships.

Image classification is a difficult problem in computer vision area. Convolutional neural network(CNN) is a successful method in dealing with image classification issues. In 2015, K.He, X.Zhang, etc proposed a famous CNN architecture named ResNet-152 that achieve 95.51% accuracy in image classification on ImageNet which is a superhuman result. In this icebergs and ships image classification problem. I want to propose a CNN architecture to solve this problem.

Problem Statement

Since this is an ongoing competition state on Kaggle, this problem does not be solved. This is a binary image classification problem, the best result up-to-date on Kaggle have 0.1029 log score and the 20th best result on Kaggle have 0.1463 log score. So the performance of the model can be evaluated by log score.

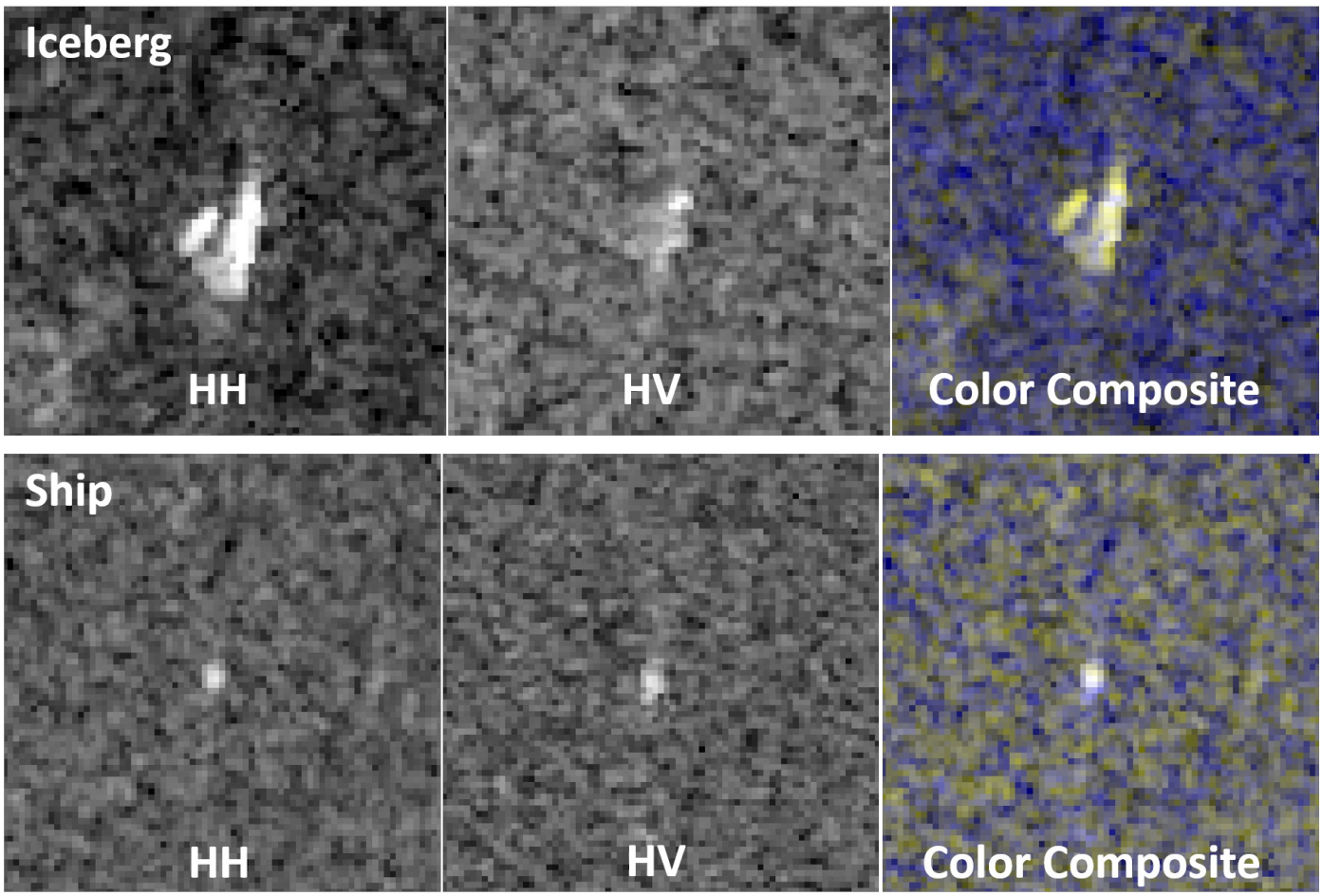
Datasets and Inputs

Statoil and *C-CORE* provide a training dataset with size 42.85mb and a test dataset with size 245.22mb. The format of the data is in JSON. For each row, the data have 5 different name fields.

- id: the id of the Image
- band_1, band_2: the flattened image data. Each band has 75*75 pixel values in the list, so in each band there is a list of 75*75=5625 elements. Each element is a float number.
- inc_angle: the incidence angle of which the image was taken. The field has missing data marked 'na'.
- is_iceberg: this is the target variable, set to 1 if there is an iceberg and set to 0 if it is a ship.

Based on the information above, we know that the image is black-white scale and I will mainly use band_1 and band_2 those two variables.

Here is the sample image.



Solution Statement

Since this is a binary image classification problem, CNN seems suitable for this problem. I will go to design a transfer learning model that uses two pre_trained ResNet50 model first. After that mix the result from the pre_trained model and add some other layers like dropout layer max-pooling layer and fully connected layer. Based on my knowledge, I don' t think the incidence angle will matter but I will try to add this feature if I have enough time.

Here is the detail:

- I will use two pre_trained Resnet50 models first. One is to handle band_1 and another is to handle band_2.
- I may add some pooling layers and dropout layers after each pre_trained Resnet50 model,
- I will use a Concatenate layer to merge two model.
- After the Concatenate layer, I may add few more layers, like fully connected layer, dropout layer, and pooling layer.
- Then I will add a fully connected layer with sigmoid activation function.

Basically, I will follow the idea state above, but this is just a statement. I may modify the structure of the model during the implementation.

Benchmark Model

noobhound proposed a CNN model on Kaggle. This CNN model reaches 0.21174 log score. The model *noobhound* used seems to be a simplified VGG-16 net. He adds some dropout layer between each block and he changes the final fully connected layer to GlobalMaxPooling layer. He also adds a Concatenate step that mix band_1 and band_2. You can find the detail implementation of he model at Here. The model is in code block 4.

Evaluation Metrics

The performance of this model can be evaluated by accuracy score.

Here is a sample table:

	Iceberg	Ship
Prediction accuracy	X%	X%

Project Design

Here is the workflow for this project.

1. Clean and transfer the data.
    - Since the image data in the JSON file is flattened, we need to convert it back to 75*75 pixels black-white scale image.
    - The input dimension for the pre_trained model is (height, weight, channel) which is 3 dimensions, but the training data is only (height, weight) which is 2 dimensions. I need to define a method to reconstruct the channel information.
    - I may also rescale the value in each image.
2. Visualize the image.
    - Visualize the image in 2D.
    - Visualize the image in 3D. Actually, we can figure out some features during the visualization analysis.
3. Design the CNN model.
    - Design the CNN model as stated in the *Solution Statement* part.
    - Try to use different optimization function to optimize the model.
4. Train the CNN model.
    - Adjust parameters to optimize the model.
5. Write report.
    - Write the final report as required.

Reference

- He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
- https://www.kaggle.com/knowledgegrappler/a-keras-prototype-0-21174-on-pl
- https://www.kaggle.com/c/statoil-iceberg-classifier-challenge