Small Vessel Detection from Synthetic Aperture Radar (SAR) Imagery using Improved Deep Learning

Preshma Linet Pereira
ppereira@umass.edu

Amulya Chetlapalli
achetlapalli@umass.edu

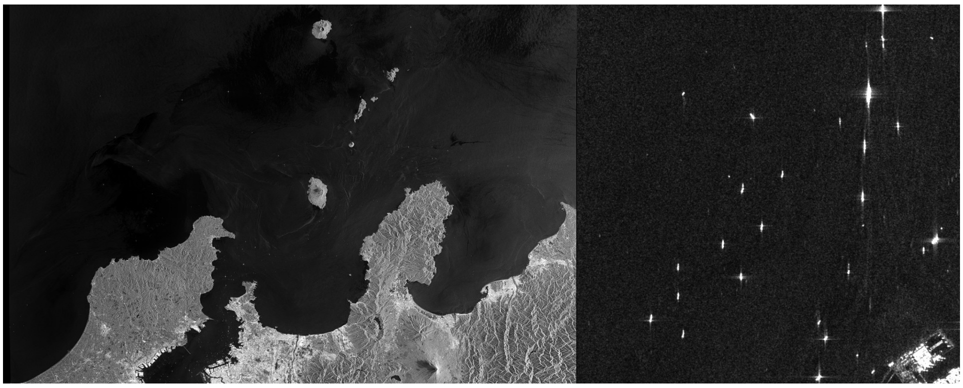
1 Introduction

Image classification on large objects is a widely explored area compared to low resolution small objects on large imagesets. One such problem is detecting small vessels from large Satellite Imagery such as SAR. Synthetic aperture radar (SAR) is an active microwave imaging sensor whose all-day and all-weather working capacity gives it an important place in marine exploration. Since the United States launched the first SAR satellite, SAR has received much attention in marine remote sensing, e.g., geological exploration, topographic mapping, disaster forecast, traffic monitoring, etc. As a valuable ocean mission, SAR ship detection is playing a critical role in helping regulatory affairs for maritime management, so it is becoming a research hotspot [6]. We propose using the LS-SSDD-v1.0 open source SAR dataset [6] to build and train a model which automatically detects maritime vessels. This type of automation would allow regulatory agencies to better conduct shipwreck rescue, fishery enforcement, and vessel traffic management. Our implementation is publicly available on Github [1].

2 Problem Statement

Ship Detection or Small vessel detection involves detecting edges around ships or vessels that are in the ocean, but what makes it interesting is detecting inshore ships (ships and utility vessels closer to the shore) due to its close proximity to other vessels and to the shore. These inshore vessels get blended with its surrounding making it difficult to draw a bounding box around it. Some ship targets and non-ship targets such as waves, dams, islands, icebergs, or reefs, have approximate back-scattering intensity in SAR that makes ship detection in SAR imagery difficult, making these problems as one of the many unsolved problems in computer vision literature [2].

[1] https://github.com/preshmalinetpereira/SSDDShipDetection



(a) Large Scale Image (24000 X 16000 pixels) (b) Clipped Training Image

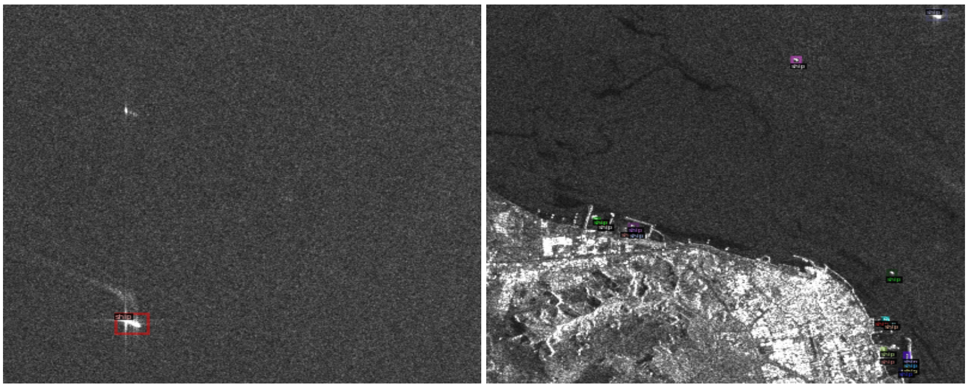
Figure 1: Comparison of original image size compared to clipped image after Data Pre-processing

2.1 Data Set

We are using the Large-scale Ship Detection Dataset-v1.0 (LS-SSDD-v1.0) from Sentinel-1 [6] as our dataset. It contains 15 large scale SAR images of size 24000 x 16000 pixels. These large images are further divided into 600 sub-images for each SAR image. So it contains a total of 9000 sub-images that are generated for network training and testing in the LSSSDD-v1.0 dataset [6]. We used the first 10 of the 15, 24,000 x 16,000 pixel images as the training set (train) and the last 5, 24,000 x 16,000 pixel images as the test set (test). The test set further contains 234 offshore images (test offshore) and 766 inshore images (test inshore). Differences between the datasets are shown below in Table 1 and Figure 2

Dataset	#Imags	#Ships	%Imags w/ Ships	Ship/ *Img	Ships Pixel/*Img Pixel
train	6000	3637	0.18	3.23	0.0016
test	3000	2378	0.24	3.23	0.0023
test offshore	2234	1495	0.27	2.41	0.0021
test inshore	766	883	0.15	7.54	0.0030

Table 1: LS-SSDD-v1.0 datasets. *Img denotes an image that has at least one ship



(a) Offshore Image sample (b) Inshore Image Sample

Figure 2: Offshore and Inshore Image Samples

2.2 Expected Results

Our problem primarily acts as a binary classifier with only 1 class ”ship”. A score threshold of 0.5 will be used to identify if the object is a ship or not. To perform quantitative analysis, we will evaluate the unseen test set to gauge the quality of the model output using Detection Probability (Pd), False Alarm (Pf), Missed Detection (Pm), Recall, Precision, Mean Average Precision (mAP), and F1 score.

3 Technical Approach

3.1 Baseline Models

We have decided on two baseline models from Facebook’s Detectron2 API that are described in [5] implemented on two neural network architectures which are (1) Faster Region-Based Convolutional Neural Network (Faster R-CNN) [3] and (2) RetinaNet [1]. These baselines models are pre-trained on the Microsoft COCO dataset. For the backbone, we have use ResNet-101 Feature Pyramid Network (FPN). We have trained the model on train data using mini batch gradient descent (MGD) with 12 epochs for 36K iterations similar to models in the original paper [6]. We also maintained the same hyper-parameters as well.

3.2 Improvements

In order to better fit our model, we are introducing a validation dataset which will allow us to better tune our hyper-parameters and improve the out-of-sample performance on the training data. We keep the test data unchanged and split the original train dataset as follows : 85 % of random images as new train data and the rest 15

% as the validation dataset.

To further improve our model performance we will use data augmentation. This approach will help us avoid over-fitting the model to the train dataset. The techniques we intend to use include copy-pasting strategy, color augmentation and geometric augmentation.

Further, we plan to test our model on the High-Resolution SAR Images Dataset (HRSID) which has been created from SAR satellite images. This will allow us to better evaluate the performance of our model[4].

4 Intermediate/Preliminary Results

We implemented the baselines as mentioned above and trained the model on the training dataset to maintain consistency with the author's in [6]. We trained using the MGD (Mini-batch Gradient descent) with 12 epochs and matched the hyperparameters where possible, as we are using Facebook's Detectron API. Following is the results for the implementation in Table 2

	Model	P_d	P_f	P_m	Recall	Precision	mAP	F1
Our Baseline Model	Faster R-CNN	74.76	26.01	25.22	74.77	73.98	71.30	0.74
Our Baseline Model	RetinaNet	58.05	19.62	41.95	58.01	80.30	53.04	0.65
Baseline Model	Faster R-CNN	77.71	26.26	22.29	77.71	73.74	74.80	0.76
Baseline Model	RetinaNet	55.51	5.38	44.49	55.51	94.62	54.31	0.70

Table 2: Test Dataset results comparison between baseline and Original Paper's baseline model

	mAP @ 36k	mAP @ 72k	mAP @ 108k	mAP @ 144k
Training	52.02	54.02	62.05	63.81
Test	70.02	69.88	69.40	67.96

Table 3: Training vs test performance at different iterations

Based on the above results, we infer that our baseline model got much lower performance than the original paper. So we calculated mAP for a number of specified iterations to understand the performance on the train and test set and have tabulated the results in Table 3. At every iteration it seems like the train set underperformed and the model underfit the train data resulting in better performances on the test data. This means it was unable to capture the relationship between both offshore and inshore scenes. We know from [6] that there is a huge accuracy gap between the inshore and offshore scenes. This phenomenon is consistent with common sense, because ships in the inshore scenes are harder to detect than the offshore, due to the severe interference of land, harbor facilities, etc. This shows that even though we've used a powerful baseline model we need to further improve the dataset by augmenting the data, adding a new validation dataset and improve the model parameters to get better performance for the training as well as test set. We also plan to test our model using a test dataset from the HRSID dataset [4] to evaluate our model performance.

References

[1] Tsung-Yi Lin et al. "Focal loss for dense object detection". In: *Proceedings of the IEEE international conference on computer vision*. 2017, pp. 2980–2988.

[2] Nhat-Duy Nguyen et al. "An evaluation of deep learning methods for small object detection". In: *Journal of Electrical and Computer Engineering* 2020 (2020).

[3] Shaoqing Ren et al. "Faster r-cnn: Towards real-time object detection with region proposal networks". In: *Advances in neural information processing systems* 28 (2015), pp. 91–99.

[4] Shunjun Wei et al. "HRSID: A High-Resolution SAR Images Dataset for Ship Detection and Instance Segmentation". In: *IEEE Access* 8 (2020), pp. 120234–120254. DOI: 10.1109/ACCESS.2020.3005861.

[5] Y.Wu et al. *Detectron2*. https://github.com/facebookresearch/detectron2. 2019.

[6] Tianwen Zhang et al. "LS-SSDD-v1. 0: A deep learning dataset dedicated to small ship detection from large-scale Sentinel-1 SAR images". In: *Remote Sensing* 12.18 (2020), p. 2997.