Small Vessel Detection from Synthetic Aperture Radar (SAR) Imagery using Deep Learning
(Application Project - Computer Vision)

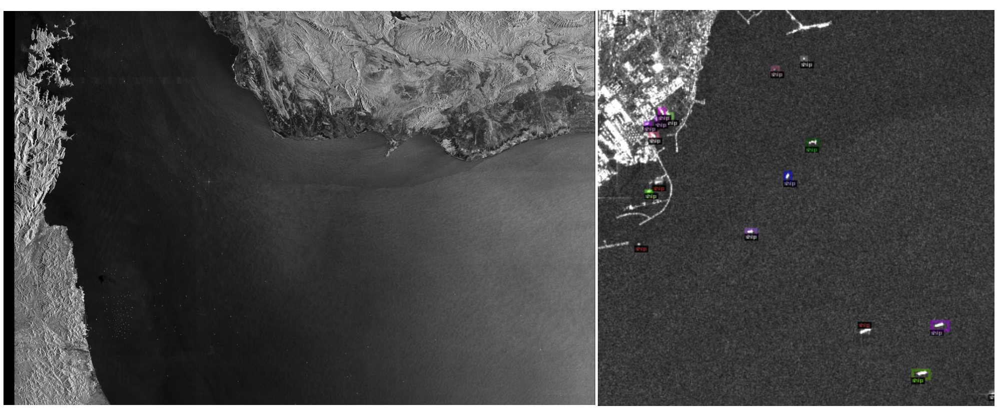
Jake Taylor
jakee417@stanford.edu

Toktam Mohammadnejad
toktammd@stanford.edu

1 Introduction

1.1 Description

Synthetic aperture radar (SAR) provides high-resolution, all-day, all-weather satellite imagery, which has become one of the most important means for high-resolution ocean observation and is well suited to better understand the maritime domain. We propose using the LS-SSDD-v1.0 open source SAR dataset to build and train a computer vision small vessel detection model which automatically generates bounding boxes around maritime vessels [1]. 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]



(a) Large Scale Image (24,000 × 16,000 pixels) (b) Clipped Training Image (800 x 800 pixels)

Figure 1: Comparison of large scale and clipped LS-SSDD-v1.0 SAR Imagery.

1.2 Challenges

Object detection from SAR imagery is challenging for a variety of reasons. Some ship targets and inshore non-ship targets such as waves, dams, islands, or reefs, have approximate backscattering intensity in SAR imagery that makes ship detection in SAR imagery difficult. Moreover, many low-level and mid-level image features that have been widely used in object detection and classification applications cannot be introduced directly into ship detection via SAR imagery, which imposes an additional challenge. Lastly, detecting small objects in large-scale remote sensing images remains an unsolved problem within the computer vision literature.

[1] `https://github.com/jakee417/LS-SSDD-v1.0-ShipDetectionComputerVision`

2 Dataset

We use the Large-Scale SAR Ship Detection Dataset-v1.0 (LS-SSDD-v1.0) from Sentinel-1 [1]. LS-SSDD-v1.0 contains 15 raw, large-scale, space-borne, 24,000 × 16,000 pixel, SAR images. To facilitate network training, the original large-scale SAR images are directly cut into 9000, 800×800 pixel sub-images. As a result, 600 sub-images are obtained from each large-scale SAR image. Cumulatively, a total of 9000 sub-images are generated for network training and testing in the LS-SSDD-v1.0 dataset [1]. The authors use the first 10 of the 15, 24,000 x 16,000 pixel images as the training set (`train`). The last 5, 24,000 x 16,000 pixel images are used as the test set (`test`). `test` is further broken down into 2234 offshore images (`test offshore`) and 766 inshore images (`test inshore`). Differences between the datasets are shown below in Table 1 and Figure 2:

Dataset	# Imgs	# Ships	% Imgs 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

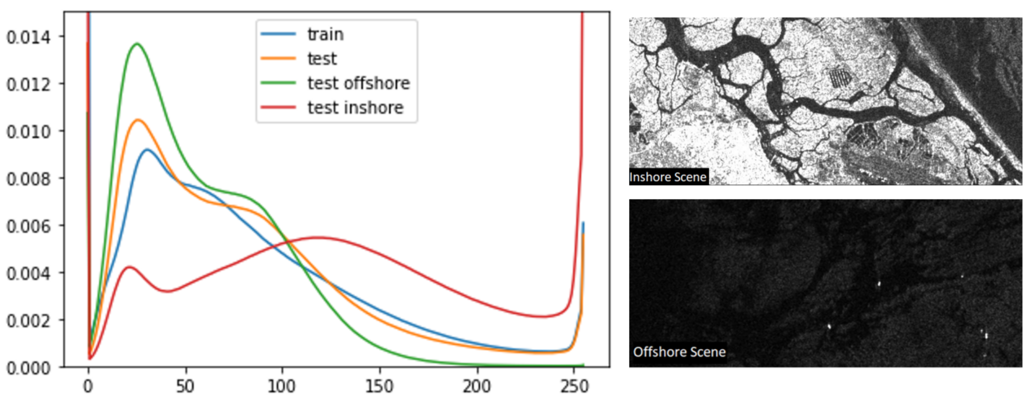


Figure 2: (Left) Sample of pixel intensity frequencies. (Right) Inshore and Offshore scenes.

3 Evaluation Metrics

To discern between vessel and non-vessel, a score threshold of 0.5 will be used. To compare model output against ground truth bounding boxes, a 50% intersection over union (IOU) will be enforced. Finally, a variety of evaluation metrics will be used on an unseen test set to gauge the quality of the model output: Detection Probability (P_d), False Alarm (P_f), Missed Detection (P_m), Recall, Precision, Mean Average Precision (mAP), and F_1 score.

4 Learning Methodology

4.1 Model Baselines

We implemented two different baseline models from Facebook's Detectron2 API, an open source detection toolbox [2; 3]. The two neural network architectures we used were (1) Faster Region-Based Convolutional Neural Network (Faster R-CNN) [4] and (2) RetinaNet [5] both pre-trained on the COCO dataset. For both of these models, we used either ResNet-50 or ResNet-101 with Feature Pyramid Network (FPN) pre-trained on the ImageNet dataset as the backbone [6]. To ensure consistency with the authors in [1], we trained on all of `train` (no validation dataset) using mini-batch gradient descent (MGD) with momentum for 12 epochs (36k iterations) and matching hyperparameters where possible given the difference in archtectures.

The comparison between baseline and our results is shown in Table 2.

	Model	P_d	P_f	P_m	Recall	Precision	mAP	F_1
Baseline	Faster R-CNN	77.71	26.26	22.29	77.71	73.74	74.80	0.76
Baseline	RetinaNet	55.51	5.38	44.49	55.51	94.62	54.31	0.70
Ours	Faster R-CNN	74.76	26.00	25.23	74.76	73.99	71.32	0.74
Ours	RetinaNet	58.03	19.67	41.97	58.03	80.33	53.02	0.67

Table 2: Preliminary results on the Test dataset

Interestingly, when inference was conducted on `train`, a much lower level of performance was observed, which is summarized in Table 3. This seems to suggest that the trained model had underfit `train` and that `test` might be an easier task to begin with. As reported in [1], we also observed that when inference was conducted on `test offshore` and `test inshore`, the `test offshore` always has a much higher mAP likely due to the absence of near-shore backscattering. Referring back to Table 1 and Figure 2, we have evidence to confirm that `train` and `test` come from different distributions and that `train` distribution may be more similar with `test inshore` distribution. Thus, there is a need to introduce a new strategy to better estimate out-of-sample performance on `train`.

	mAP @ 36k	mAP @ 72k	mAP @ 108k	mAP @ 144k
Training	52.32	55.02	62.83	65.81
Test	71.32	70.90	70.40	67.96

Table 3: Training versus Test performance at specified amounts of training iterations

4.2 Introducing a Validation Dataset

To better estimate out of sample performance, we plan to depart from [1] and adopt `train w/ val`, `validation`, and `test` datasets. We randomly shuffle the `train` dataset into `train w/ val` and `validation` datasets, while the `test` dataset will remain unchanged so that we can compare the obtained results with baseline. We use 15% of the `train` dataset for `validation`.

4.3 Data Augmentation

We plan to use data augmentation to help with performance improvement and potential overfitting to the training dataset. Other than using commonly-used data augmentation techniques, such as random flipping, random rotation, and color augmentation (e.g., changing brightness and contrast), we plan to implement an augmentation algorithm based on copy-pasting ships [7].

5 Related Work

A survey of works related to deep learning-based object detection and specifically small object detection can be found in [8] and [9], respectively. This project specifically relies on the dataset and benchmark results given by [1], which has related research in [10; 11; 12].

6 Conclusion

We observed a large accuracy difference between the inshore scenes and offshore scenes. This observation is due to the fact that ships in the inshore scenes are harder to detect than those in offshore scenes because of the severe interference of land, harbor facilities, etc. The land masking strategy (e.g. sea-land segmentation) may help improve the accuracy of ship detection in inshore scenes [13].

7 Contribution

Both group member contributed equally to the project. Jake implemented Faster R-CNN and the evaluation logic, while Toktam implemented RetinaNet and worked on the training procedure.

References

[1] T. Zhang, X. Zhang, X. Ke, X. Zhan, J. Shi, S. Wei, D. Pan, J. Li, H. Su, Y. Zhou, and D. Kumar, "Ls-ssdd-v1.0: A deep learning dataset dedicated to small ship detection from large-scale sentinel-1 sar images," *Remote Sensing*, 2020.

[2] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2." `https://github.com/facebookresearch/detectron2`, 2019.

[3] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," 2020.

[4] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015.

[5] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," 2018.

[6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," 2015.

[7] M. Kisantal, Z. Wojna, J. Murawski, J. Naruniec, and K. Cho, "Augmentation for small object detection," *CoRR*, vol. abs/1902.07296, 2019.

[8] L. Jiao, F. Zhang, F. Liu, S. Yang, L. Li, Z. Feng, and R. Qu, "A survey of deep learning-based object detection," *CoRR*, vol. abs/1907.09408, 2019.

[9] K. Tong, Y. Wu, and F. Zhou, "Recent advances in small object detection based on deep learning: A review," *Image and Vision Computing*, vol. 97, p. 103910, 2020.

[10] T. Zhang and X. Zhang, "High-speed ship detection in sar images based on a grid convolutional neural network," *Remote Sensing*, vol. 11(10), 2019.

[11] T. Zhang, X. Zhang, J. Shi, and S. Wei, "Hyperli-net: A hyper-light deep learning network for high-accurate and high-speed ship detection from synthetic aperture radar imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 167, pp. 123 – 153, 2020.

[12] T. Zhang and X. Zhang, "Shipdenet-20: An only 20 convolution layers and <1-mb lightweight sar ship detector," *IEEE Geoscience and Remote Sensing Letters*, pp. 1–5, 2020.

[13] Q. An, Z. Pan, and H. You, "Ship detection in gaofen-3 sar images based on sea clutter distribution analysis and deep convolutional neural network," *Sensors*, vol. 18, no. 2, 2018.

[14] Z. Cai and N. Vasconcelos, "Cascade R-CNN: delving into high quality object detection," *CoRR*, vol. abs/1712.00726, 2017.

[15] J. S. Z. X. H. F. Jiangmiao Pang, Cong Li, "R2-cnn: Fast tiny object detection in large-scale remote sensing images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, pp. 5512 – 5524, 2019.

[16] K. Chen, J. Wang, J. Pang, Y. Cao, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Xu, Z. Zhang, D. Cheng, C. Zhu, T. Cheng, Q. Zhao, B. Li, X. Lu, R. Zhu, Y. Wu, J. Dai, J. Wang, J. Shi, W. Ouyang, C. C. Loy, and D. Lin, "Mmdetection: Open mmlab detection toolbox and benchmark," *CoRR*, vol. abs/1906.07155, 2019.