Ship detection using Region-Based Convolution Neural Network (R-CNN) from SAR remote sensing data

Abdalmenem Owda, Balazs Krupinszki
Supervisor: Ole Winther

DTU Compute, Technical University of Denmark

Introduction

Synthetic Aperture Radar (SAR) sensors are considered a valid alternative to the coastal-based means and are particularly suitable for the detection of ships in open sea scenarios thanks to their capability to acuqire images independently from daylight and weather conditions. Traditionally, SAR ship detection algorithms rely on Constant False Alarm Rate (CFAR) methods: the sea clutter background is modelled according to a suitable distribution and a threshold is set to achieve a given probability of false alarm[5].

The increasing popularity of Machine Learning promoted the use of different Deep Learning algorithms in this field as well. Multiple object detection models have been proposed in the past few years (e.g. Region-Based CNN (R-CNN)[3], Fast R-CNN[2], Faster R-CNN[8], Single Shot Detector(SSD)[7], etc.) which can be utilized for SAR target detection. In this study we are going to explore and investigate the performance of one of the most successful models - Faster R-CNN - on a public SAR ship dataset [9] in order to find and prepare a well performing architecture for our future works.

Key points

- **Obtain** a large, unified and labeled SAR ship dataset.
- **Preprocess** the dataset for use - extract image sizes; extract and calculate bounding boxes.
- **Explore** the possible models and architectures to use and decide which ones to utilize.
- Implement and optimize the model.

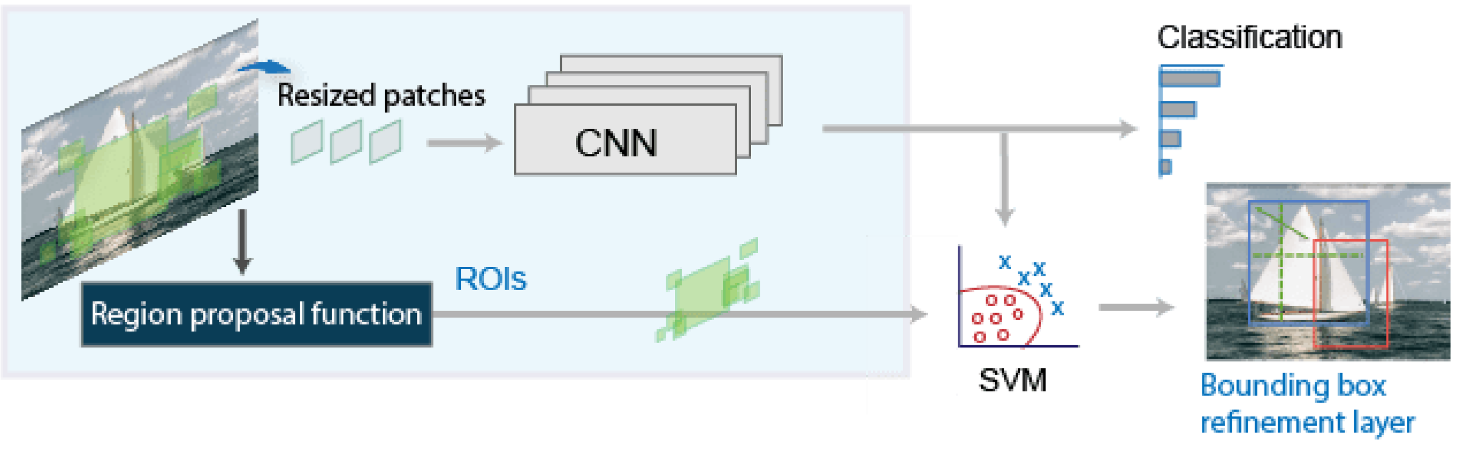
Dataset

Several SAR ship datasets have been published and collected[1] for many different uses with a variety of image sizes and additional data. One of the biggest datasets available - Sar-Ship-Dataset - seemed promising for our application. Not only does it contain large amounts of ship instances (59535), but the relatively small image sizes are also well suited for our use without any transformations needed. To make the data applicable for the model, the dataset had to be preprocessed first. Image names, image sizes, and bounding boxes had to be extracted from the individual text files and collected in a common csv for further use.

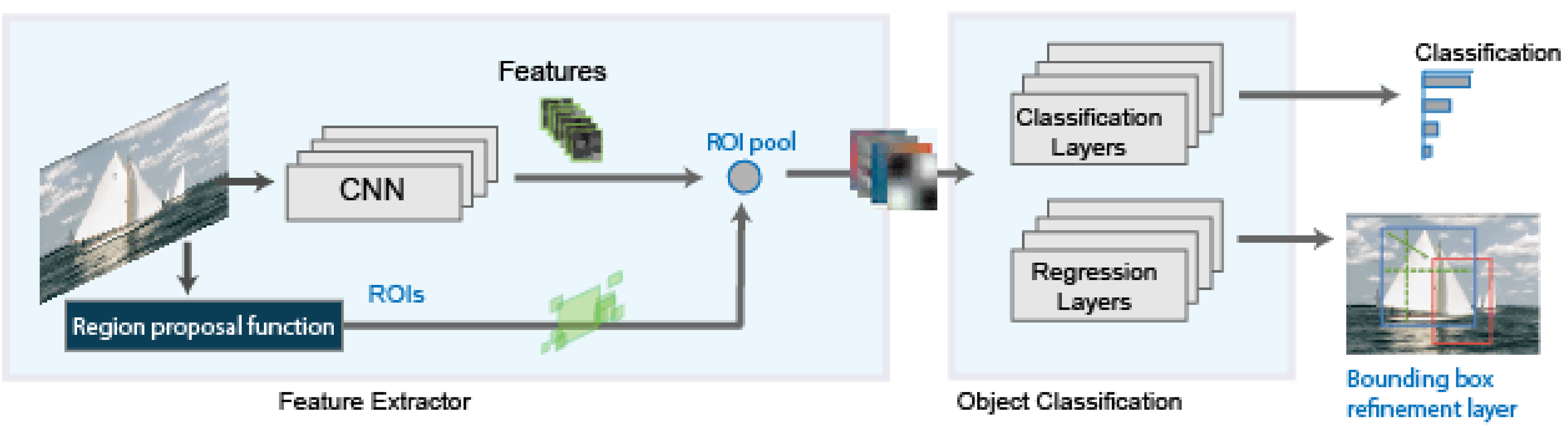
Breakdown of R-CNN models

Region-Based CNN models have been widely used for object detection purposes for a while and several improvements have been made to them to make them more precise.

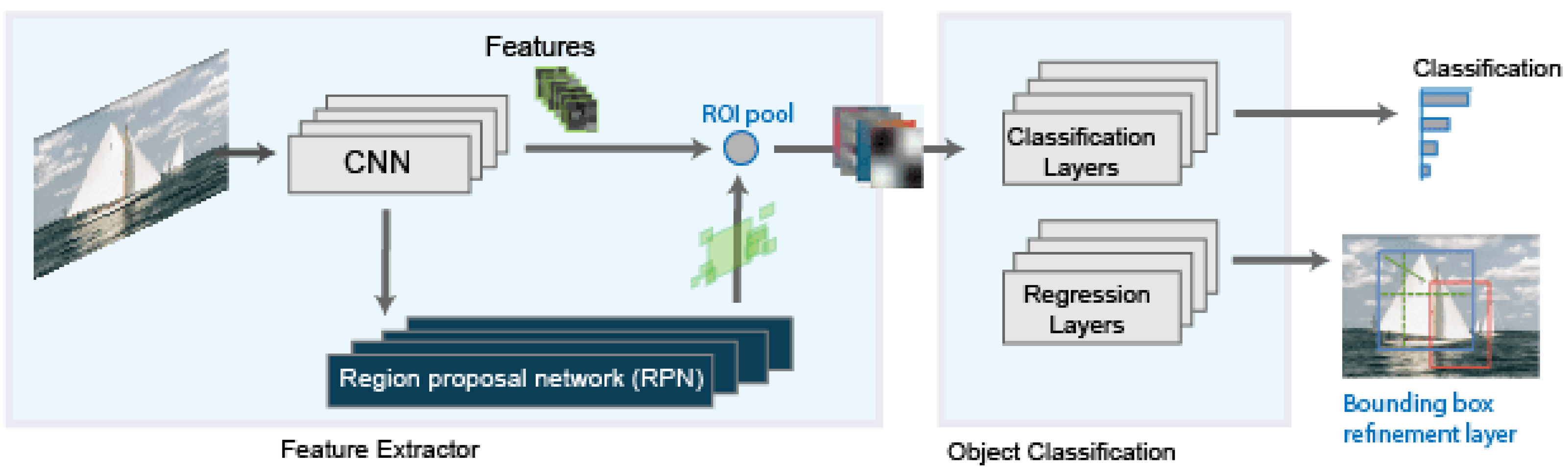
The basic **R-CNN** detector first generates region proposals (ROIs - Region of Interests) using an algorithm, such as Edge Boxes [10]. These region proposals are then cropped out of the original image and resized, then classified by the CNN. Finally, the proposed bounding boxes are refined by a support vector machine (SVM) that is trained by the CNN features.



The **Fast R-CNN** also uses a region proposal algoritm to generate ROIs, but unlike the R-CNN detector, it first generates the shared feature map on the entire image and then pools CNN features from this corresponding to each region proposal, thus it is much more efficient as computations for overlapping regions are shared.



The **Faster R-CNN** shares a lot of it's arhitecture with the Fast R-CNN detector. The main difference is that it uses a region proposal network (RPN) to generate ROIs. The RPN utilizes the shared feature map generated by the CNN to make the region proposals, thus it is faster than using an external RP algorithm.



Backbone CNN

We settled on utilizing the **Faster R-CNN** detector due to the advantages it provides over the other two models. A typical Faster R-CNN model can be broken down into four main parts:

- Backbone CNN that creates the shared feature map
- Region Proposal Network
- ROI pooling
- Fully connected classifier layer

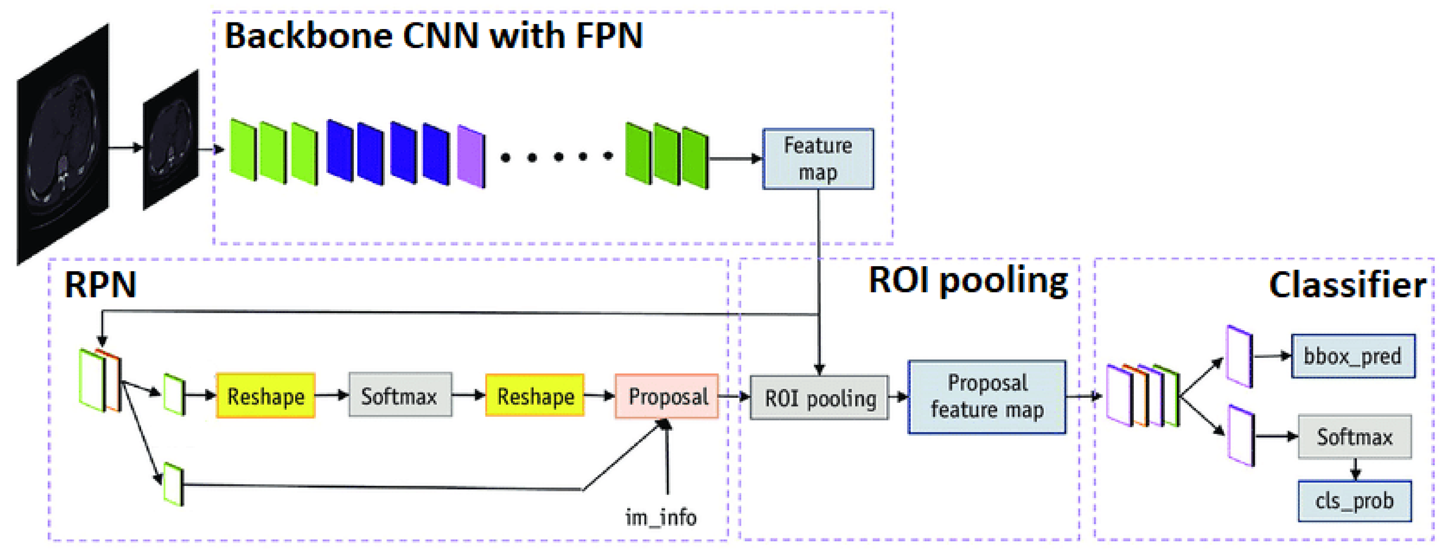


Figure 1: Typical Faster R-CNN Arhitecture

However, several implementations of the Faster R-CNN model are possible depending on the CNN architecture used as the backbone. We utilized 3 different (pre-trained) implementations of the Faster R-CNN detector:

- ResNet50 FPN: A 50 layer deep CNN with a Feature Pyramid Network (FPN) [6]
- MobileNetV3 Large [4] FPN: A deep and fast CNN with an FPN, optimized for slower hardware. Competitive accuracy compared to ResNet50, but much faster execution.
- MobileNetV3 Large 320 FPN: An iteration of the MobileNetV3-Large FPN that uses reduced resolution, thus sacrifices accuracy for speed.

Results

Region proposals of the three different architectures (in green) after 3 epochs of training, and the target bounding boxes (in red).

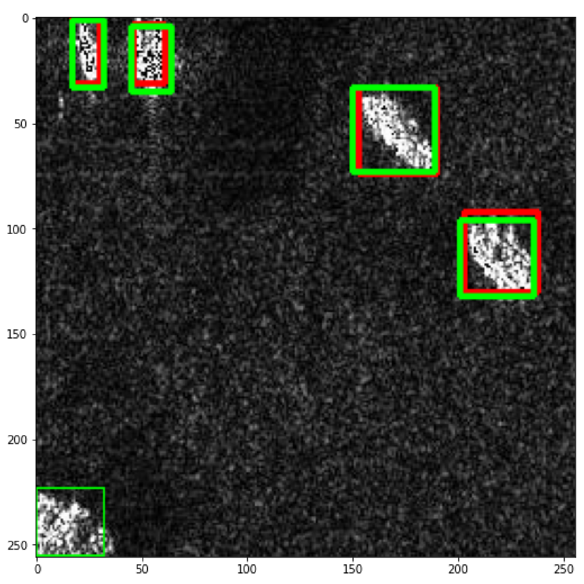


Figure 2: ResNet50

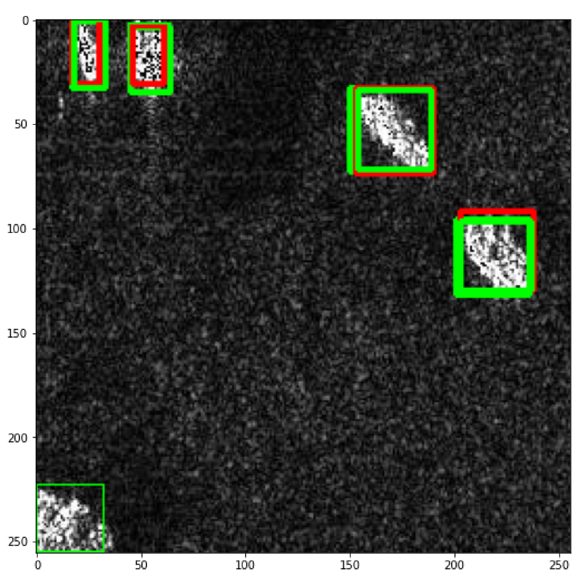


Figure 3: MobileNetV3 Large

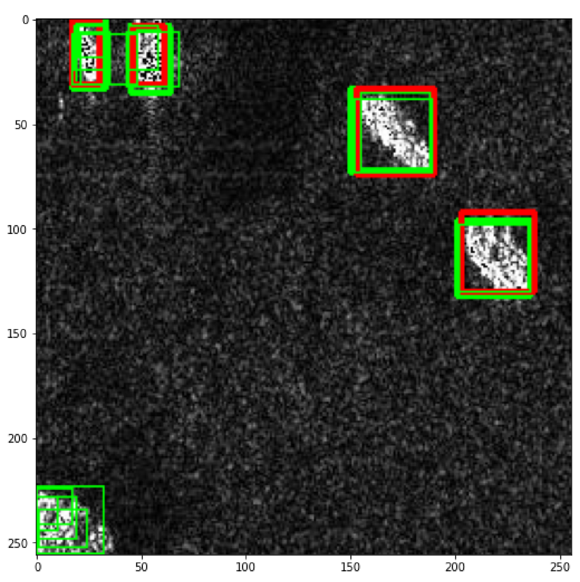


Figure 4: MobileNetV3 Large 320

Table 1: Performance of the three architectures.
mAP@[.5:.95] = average mean average precision (mAP) over different IoU (intersection over union) thresholds, from 0.5 to 0.95, step 0.05. AR = Average Recall.

Model	mAP@[.5:.95]	AR
ResNet50 FPN	0.872	0.810
MobileNetV3 Large FPN	0.898	0.762
MobileNetV3 Large 320 FPN	0.785	0.680

All three architectures yield great results and should be considered for further optimization. While the ResNet might perform slightly better, the execution speed of the MobileNetV3 can prove valuable in a few applications.

References

[1] Satellite imagery datasets containing ships. URL https://github.com/JasonManesis/Satellite-Imagery-Datasets-Containing-Ships.

[2] R. Girshick. Fast r-cnn, 2015.

[3] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. pages 580–587, 2014. doi: 10.1109/CVPR.2014.81.

[4] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam. Searching for mobilenetv3, 2019.

[5] P. Iervolino, R. Guida, P. Lumsdon, J. Janoth, M. Clift, A. Minchella, and P. Bianco. Ship detection in sar imagery: A comparison study. 07 2017. doi: 10.1109/IGARSS.2017.8127384.

[6] T.-Y. Lin, P. DollÃ¡r, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature pyramid networks for object detection, 2017.

[7] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. Ssd: Single shot multibox detector. *Lecture Notes in Computer Science*, page 21â37, 2016. ISSN 1611-3349. doi: 10.1007/978-3-319-46448-0_2. URL http://dx.doi.org/10.1007/978-3-319-46448-0_2.

[8] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.

[9] Y. Wang, C. Wang, H. Zhang, Y. Dong, and S. Wei. A sar dataset of ship detection for deep learning under complex backgrounds. *Remote Sensing*, 11(7), 2019. ISSN 2072-4292. doi: 10.3390/rs11070765. URL http://www.mdpi.com/2072-4292/11/7/765.

[10] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 391–405, Cham, 2014. Springer International Publishing. ISBN 978-3-319-10602-1.