Maritime Ship Detection Using Synthetic Aperture Radar Satellite Imagery

Alexander Makhratchev and Sean Ng

University of California: San Diego

December 5, 2022

Abstract

In the past, synthetic aperture radar (SAR) images did not have a high resolution, and required immense computational power to create. However, due to recent advances in remote sensing, SAR image quality has significantly increased and is becoming a hot topic in research. One application of these images is to detect container ships traveling through bodies of water. This paper uses the Large-Scale SAR Ship Detection Dataset-v1.0 (LS-SSDD-v1.0) which contains large scale SAR images from Sentinel 1 satellites that contain ships and their respective labels. These large images are cut into smaller pieces in order to train a variety of object detection models to locate vessels of different shapes and sizes. We trained our own deep learning models based of the *You Only Look Once* (YOLOv7) framework. When comparing the models to other deep learning frameworks and older an version of YOLO, we found its performance in terms of accuracy to be on par or better than some of the models. The results from this paper can be applied to fields such as maritime rescue and economic modeling.

1 Introduction

With recent advances in algorithms and technology within the realm of computer vision, satellite imagery has become an applicable field of study. Roughly 1,700 commercial satellites were launched in 2021, bringing the total to 5,000 satellites orbiting Earth by the end of 2021 [5]. These satellites can complete an orbit around Earth in under 2 hours and take 5m resolution pictures from over 500 km away providing vast amounts of image data. Using computer vision techniques, we are able to obtain necessary information from the images. Specifically, analyzing satellite imagery for ship detection has been subject to a lot of research where different satellites and detection models have been created and improved on for better speed and accuracy. One of the most widely used image data within this sphere has been the SAR images. Previously, optical satellites have been known to produce some of the highest resolution images, however with the drawback that it can only produce high resolution images under certain time of day and weather conditions [3, 1]. SAR's strengths lie in the fact that it can produce high quality images even if it is cloudy or at night time.

Ship detection has been a popular subject for research within the fields of object detection. And with the use of SAR satellite imagery, machine learning and deep learning models are able to more accurately detect vessels across the globe. With uses in various economic tasks, search and rescue operations, aiding in determining military decisions, managing ship traffic, and monitoring possible illegal activity, not only do models have to be accurate, but they also need to be quick as well. Previously, other researchers were only to do classification or extract the region of interest from an image. Now, with the emergence of deep learning, we are able to automatically detect vessels without the need of handcrafted features or large amounts of computational resources. The *You Only Look Once* (YOLO) framework has been the most popular and main subject of research within the deep learning object detection sphere in the past few years. However, much literature regarding ship detection and specifically small ship detection, have yet to assess the latest version of YOLO with many papers often using older YOLO versions when either testing the model or comparing it to different deep learning architectures. Utilizing the current latest YOLO framework (YOLOv7), we were able to produce accurate and rapid results for the task of detecting ships in SAR images.

2 Literature Review

A research group in China [10] released their paper on ship detection and the accompanying dataset: LS-SSDD-v1.0. Their goal was to create the most accurate and deployable dataset for cargo ship detection from SAR images. In order to provide the most accurate ground truth annotations, the team used Automatic Identification System (AIS) to locate ships in a specific image and label them. Additionally, the images were overlaid on top of an optical Google Earth engine map to make sure no small islands were annotated as ships. The images were collected in large swaths up to 250 km in size from 15 different locations around the globe, mimicking the data the algorithm will be deployed on. Ships of various sizes were annotated including large cargo container ships and smaller fishing vessels. The large images were divided into smaller ones so they can be easily fit into a training network. None of the images were discarded even with no ships, because we want our classifier to be able to differentiate between backgrounds and ships, reducing false positives. Overall, this dataset attempts to improve every aspect of previously released datasets for SAR ship detection and that is why we will be using it in this paper.

There are a few other datasets for this task, but we did not choose them due to a variety of reasons. The first one being SSDD [4] dataset, Figure 1 Row 1. It contains 1160 SAR images that are 500x500 pixels and obtained from a mix of satellites with varying resolutions. Two major drawbacks are that annotations are not double checked with Google Earth or AIS and there are numerous repeated scenes, which does not benefit our model. Another dataset we considered was SAR-Ship-Dataset, [9] which contains 43k SAR images of size 256x256 pixels and contains 60k ships, Figure 1 Row 2. Once again the images were collected from many different satellites so their resolutions differ. One change from the previous dataset was that all images without ships were discarded, which increases false positive rate. The labeling on this dataset was done by SAR experts, which still leaves room for error. To combat some of these issues, AIR-SARShip-1.0 [8] a dataset was proposed, which contained 31 SAR images of size 3000x3000 pixels, Figure 1 Row 3. However, these images contained too much noise which reduced the effectiveness of the ship detector. Also, the ground truth was determined by experts solely. The last dataset we considered was the High-Resolution SAR Images Dataset [7] which contained 5604 SAR images of size 800x800 pixels of various resolutions, Figure 1 Row 4. In this dataset, the ground truth was determined using Google Earth, but images with pure backgrounds were abandoned. Thererfore, we chose the LS-SSD-v1.0 dataset to use for our experiment.

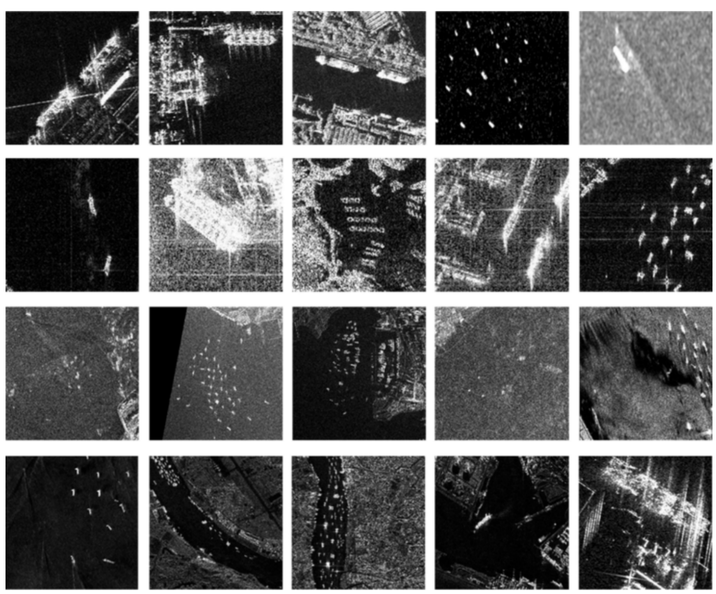

Figure 1: Each row contains sample images from each of the datasets in the order: SSDD, SAR-Ship-Dataset, AIR-SARShip-1.0, High-Resolution SAR Images Dataset. Image Source: [10]

Kanjir et al. released a paper reviewing 119 papers on ship detection using optical satellite images. In addition, they also compared and contrasted the different imaging systems used for vessel detection such as optical and reflected infrared, hyperspectral, thermal infrared, and radar [3].

Each of the imaging systems has their pros and cons, but radar and specifically Synthetic Aperture Radar (SAR) satellites are the best sensor for ship detection. Hyperspectral sensors and thermal infrared both have too low of a spatial resolution to be used for ship detection. While optical satellites have a high spatial resolution, bad weather conditions may make it hard for an optical satellite to get clear images. For example, Figure 2 below showcases an example of an optical image that is covered mainly by clouds and can make classification quite difficult. In addition, it is also dependent on the reflection of sunlight off the Earth's surface, so if it's an overcast day or even at nighttime, the optical satellite can't get clear images. SAR on the other hand is able to consistently collect high resolution images over a wide area due to its use of radar signals that are bounced off the Earth's surface. In addition, radar signals allow the SAR satellite to be able to collect data regardless of time of day as well as most weather conditions. Not only that, but larger ships often appear as bright objects in SAR images because many are made of metal and contain sharp edges that reflect the radar signals very strongly and allow for better ship detection. There are some drawbacks with SAR images being noisy, having a tough time detecting small ships, being sensitive to high winds and certain sea conditions, and classification of vessels is difficult [3]. Despite these drawbacks SAR satellite imagery is the best system we have for collecting consistent high quality images.

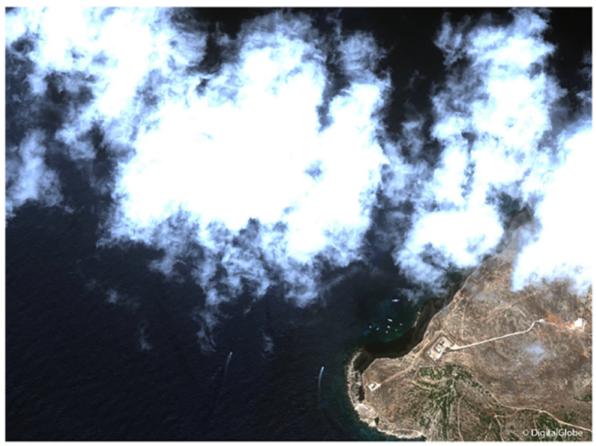

Figure 2: Clouds are a major nuance when dealing with optical data. Image Source: [3]

Chang et al. firstly explores older object detection methods and models to give background, but more importantly compares and contrasts the more recent sophisticated deep learning models that have been popular. They compare the R-CNN, Fast R-CNN, Faster R-CNN and finally the YOLOv2 models in the order of how each model improves on the other. R-CNN is the slowest model out of all of them which led to the development of the Fast R-CNN model that improves runtime by utilizing a much faster softmax function instead of support vector machines. In both the R-CNN and Fast R-CNN models, they use a selective search algorithm to identify a small subset of regions that might contain a ship, however this method is quite slow. Faster R-CNN improves on this by allowing a neural network to predict the proposed areas that might contain a ship. While Faster R-CNN does perform quickly, its accuracy suffers [1].

Chang et al. finally references the You Only Look Once (YOLO) model which performs even faster than Faster R-CNN due to the fact that the whole model is contained in a single network which allows for easier optimization. In addition, since the Faster R-CNN model only looks at regions in isolation, it often misidentifies parts of the background as objects. On the other hand, the YOLO model sees the whole image during training and is able to learn contextual information which leads to better classification. In their results, they show that YOLOv2 had around 20% better accuracy on two different datasets than Faster R-CNN and also performed around 5.8 times faster than Faster R-CNN [1]. These performance results made it quite clear to us that the YOLO model was the better choice for our research.

3 Data

In this paper we will be using the LS-SSDD-v1.0 dataset which is obtained from the C-band from the Sentinel-1 satellite. Because vessels on water experience higher backscattering values, only the co-polarization channels (VV) are kept from these images. There are 15 large scale images in this dataset of size 24,000 x 16,000 pixels. The first ten images are the training set and the last five images are the test set. In order to be able to load the data onto a GPU during training, each image is cut into 600 equal sized pieces of 800 by 800 pixels. This data resembles our data in deployment the most due to the large size of the images as well as the quality and resolution of them.

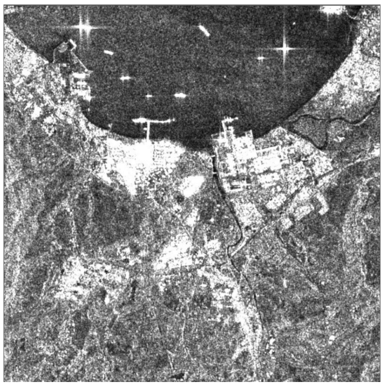

Figure 3: Inshore image with ships

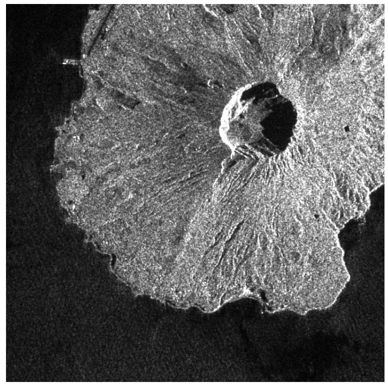

Figure 4: Inshore image without a ship

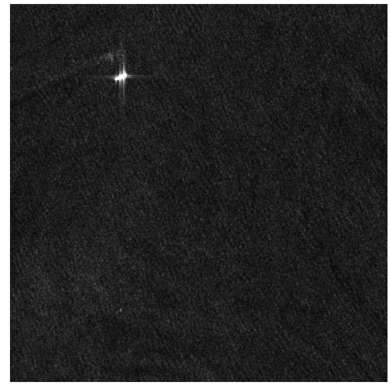

Figure 5: Offshore image with a ship

4 Methods

The raw LS-SSDD-v1.0 dataset contains a lot of redundant information, such as both SAR sub images and the large raw SAR images, as well as images in the cross-polarization channel (VH). For reproducibility reasons, we decided to extract only the relevant information: sub-images and images and upload them to a separate Google Drive folder. Once we had downloaded the edited dataset, we had to convert the XML labels to YOLO formatting to be able to use them in our YOLO models. XML labels specify the absolute coordinates of the top left and bottom right for each bounding box. But YOLO requires us to specify the middle point of the bounding box as well as the width and height of it, all of which needs to be normalized between 0 and 1. After the new labels are created, we separate the dataset into train and test from the text files provided by the authors. The authors of the LS-SSDD-v1.0 dataset wanted it to be a standardized research baseline, so all of the image preprocessing is already done for us.

As mentioned in the previous section, we chose YOLO object detection models due to their high computational speed and high accuracy when compared to the Faster R-CNN model. Their ability to see the whole image during training and testing allows them to take into account contextual information and better detect ships whether in the middle of the sea, or near land and ports. In our experiment, we decided to do transfer learning using two of the pre-trained models within the YOLOv7 Github repository.

For our first model we chose the YOLOv7-w6 as our lighter weight model, using around 70.4 million parameters [6]. When tested on the MS COCO dataset, the YOLOv7-w6 model performs 8th overall when compared with other models with an average precision of 54.9 [2]. Because of its lighter weight, the model was also able to run at 84 frames per second. During training, we found the best parameters to run with the resources we had to be an image size of 640 pixels for faster but still accurate computations, as well as a batch size of 8 so that we don't overload our computational resources. We trained for 200 epochs and utilizing 4 cores and a GeForce RTX 2080 Ti, it was able to run in roughly 10 and a half hours. We also used the optimizer's default values for momentum, learning rate, learning rate decay. In addition, after analyzing the prediction outputs after training, we chose a value of 0.02 for our object detection probability and a value of 0.5 for our Intersection over Union threshold when testing our model on the test set. To test our models, we used to provided test.py script from the YOLOv7 repository and after analyzing the outputs from initial testing, we chose an improved confidence score of 0.097 to maximize the F1 score when testing our model.

The second model we experimented with was the YOLOv7-d6 model and is twice the size of the previous model [6]. It contains 154.7 million parameters and currently holds the second best score for Average Precision on the COCO dataset [2]. Additionally, the model can output predictions at a rate of 44 frames per second, making it extremely quick at inferencing. The following values are the model parameters we chose for our best model of this type. We made the image size 640 pixels, which still retains all the necessary information in the image and makes it less computational expensive. For batch size, we chose 8, which is the maximum amount of images we could fit on our GPU. We trained for 300 epochs as suggested by the YOLOv7 creators [6]. We used the default momentum, learning rate, and learning rate decay that is implemented in YOLOv7 in the optimizer. After the model was trained, we had to select a confidence threshold of 0.239, because it maximized our F1 score. Lastly, we chose 0.5 as our Intersection over Union threshold.

5 Results

5.1 YOLOv7-W6

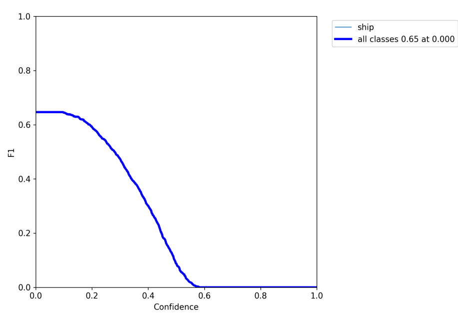

Figure 6: F1 Curve vs Confidence Graph

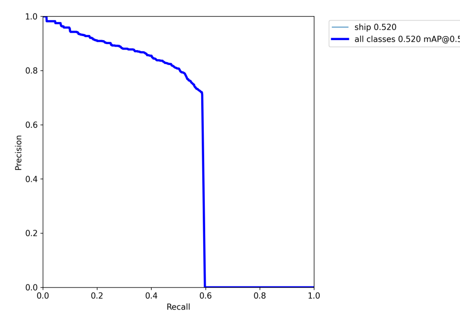

Figure 7: Precision vs Recall Graph

For this model, we chose a confidence threshold of 0.097 to maximize our F1 score. Looking at Figure 6, we can see our F1 score starts out at around 0.65 and stays consistent until a confidence level of around 0.097, then slowly decreases as the confidence increases. Figure 7 showcases the precision recall curve, where the precision score drops significantly when the recall value is roughly 0.6.

5.2 YOLOv7-D6

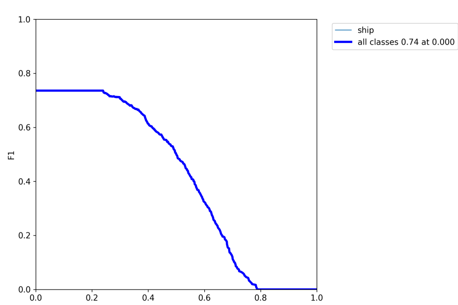

Figure 8: F1 vs Confidence Graph

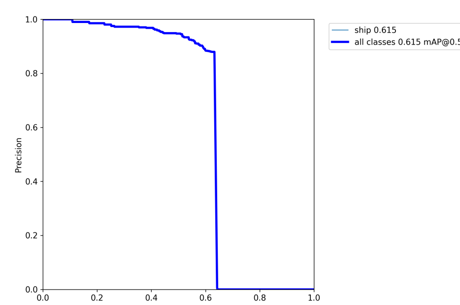

Figure 9: Precision vs Recall Graph

For the confidence threshold, we chose 0.239, because it is the value that would maximize the F1 score. From Figure 8, we can see that the F1 score stays constant until it reaches the confidence threshold value and begins to decrease significantly. Figure 9 shows us that our precision score is much higher than recall, so our model tries to minimize the false positive rate.

Model	Confidence T	Recall	Precision	F1	Size (mb)	mAP@0.5	Time(ms)
YOLOv7-w6	0.097	0.587	0.719	0.65	140.1	0.52	16.4
YOLOv7-d6	0.239	0.63	0.879	0.74	266	0.615	32.5

Table 1: Comparison of Models. Confidence T is the threshold previously mentioned. Time is the inference time per image

If we use Recall, Precision, or F1 score as the metric for selecting the best model, YOLOv7-d6 is the best one. However, it is much larger and inference time is double that of the W6 model.

6 Discussion

Looking at Table 1 above, we can see that our W6 model is about 47% smaller than the D6 model in terms of physical memory. This smaller overall size leads to a much faster run-time overall where the w6 model also runs about 49% faster as well. This decrease in size and increase in run-time seems to tell us that there is a proportional increase in run-time. However, this increased run-time and smaller size leads to a very apparent trade-off in how well the model performs. We can see that the difference between d6 and w6 model's precision score is 0.16 and the F1 score has a difference of 0.09. This intuitively makes sense because a deeper network often yields more accurate results while running slower. However, in our specific case, since we are looking to detect objects from static images and doesn't require real-time predictions and we want to ensure better predictions from our model, the d6 model proves to be the better model for us.

The F1 score balances the precision and recall, but we want to focus more on precision, because we want to penalize false negative more. In our data, the inshore images have a lot more noise origination from the land. This can easily cause our model to mis-classify large chunks of land as ships. In Figure 10, we can see the very low confidence that is assigned to the ships near the port, and even how some objects on land have a higher confidence than the ground truth ships on water. In order to keep our precision high, we had to maintain a high confidence threshold.

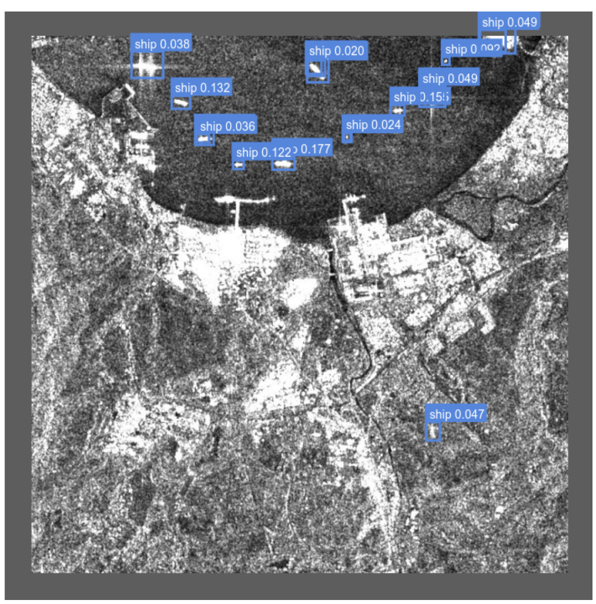

Figure 10: Model Predictions with Confidence

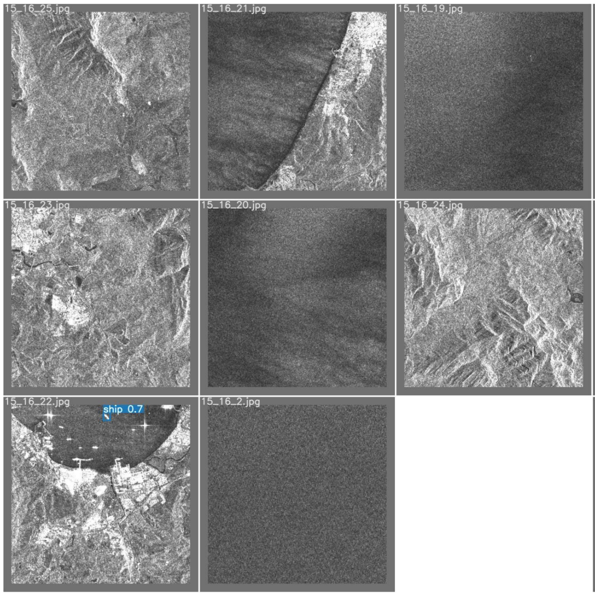

Figure 11: Model Prediction and Confidence

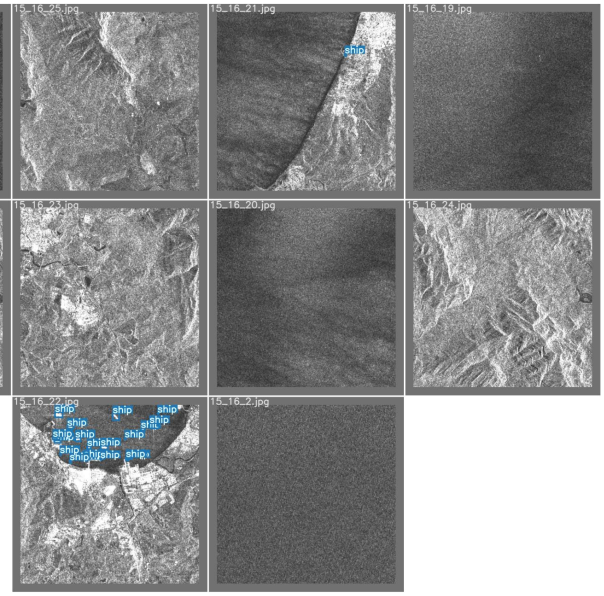

Figure 12: Ground Truth

In Figure 11, we can see that a high confidence threshold prevents a lot of the boats from being detected, but it also helps prevent detection of land. Figure 12 shows the ground truth labels, and how our model missed many of them near the land.

The second batch of images mostly contain offshore images as see in Figure 14. Images with only water in them look similar to static, and ships leave a very bright mark, which is simple to recognize. In Figure 13, we can see the classifier predicted both ships correctly, with a confidence value above the threshold. With very little noise in an image, the classifier is able to accurately detect the ships.

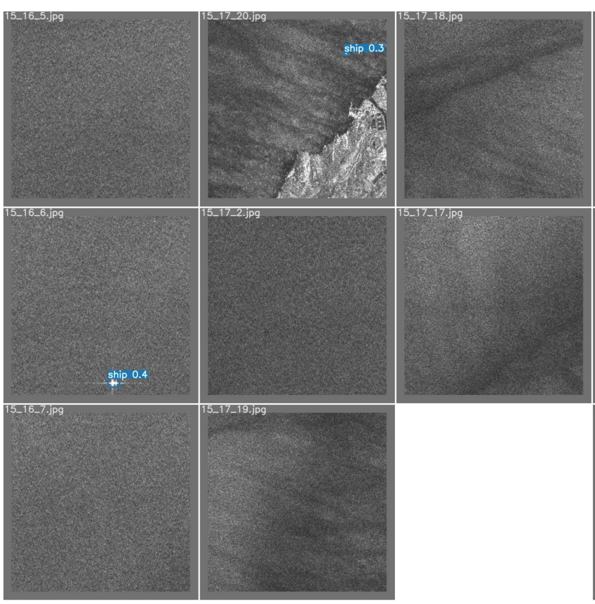

Figure 13: Model Prediction and Confidence

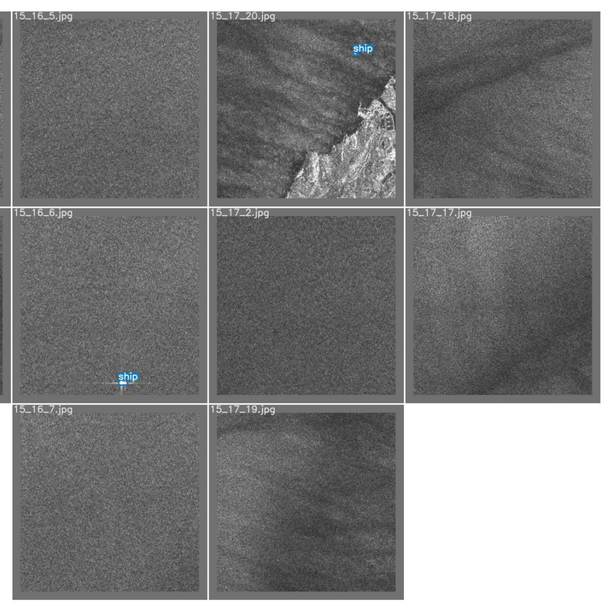

Figure 14: Ground Truth

We compare our model to the models presented in the [10] paper, which released this data set. The first row in Table 2 is our model and the subsequent ones are from [10]. When comparing our YOLOv7 model to YOLOv3, the YOLOv3 model ran in 46 ms[10] which is a bit slower than our YOLOv7 model

Model	Recall	Precision	F1
YOLOv7-D6	0.63	0.879	0.74
YOLOv3	0.26	0.86	0.4
RetinaNet	0.56	0.95	0.7
Cascade R-CNN	0.73	0.84	0.78
Faster R-CNN	0.78	0.74	0.76

Table 2: Comparison of Models.

that ran in 32.5 ms. But this is to be expected because our YOLOv7 model is about 48 mb smaller. However, what's surprising is that despite the smaller size the YOLOv7 model performs better with a 0.37 increase in recall, a 0.19 increase in precision, and a 0.34 increase in the F1 score.

The first three models in Table 2 are Single Shot Detectors (SSD) and are generally less accurate when it comes to detecting smaller objects, but are much more efficient at inference time. This can be seen in the lower recall and F1 score. The last two models in Table 2, Cascade and Faster R-CNN include a Region Proposed Network in their architecture for feature extraction, which allows them to detect objects at a higher accuracy in exchange for inference speed. In future works, we hope to able to train such networks for this task, in order to increase our F1 score.

Our goal was to be able to deploy this model on new SAR satellite image data. We hope to be able to accurately count the number of ships in a specific region. The uses of this can range anywhere from economic predictions to rescue missions. However, some limitations do exist for this model. It is unable to accurately detect ships near a port, which makes us rethink our approach for this problem. In further work, we hope to classify an image first whether it's offshore or inshore, and then use a specific object detection algorithm trained for that scenario to detect ships. This would allow for a more lightweight model to be used for offshore images while using a model with more layers for the inshore ship detection.

7 Individual Contributions

My partner (Sean Ng) and I (Alex) both contributed equally to this project. We both did research into which datasets to use and compared our findings with one another. When creating the test scripts, we split the work in half, with Sean focusing more on the storage and download aspect, while I was tasked with having the correct formatting. Both of us had to set up and run object detection models on DSMLP and use software to monitor training metrics. For the written report, we split the work in half and edited each others half after.

References

[1] Yang-Lang Chang, Amare Anagaw, Lena Chang, Yi Chun Wang, Chih-Yu Hsiao, and Wei-Hong Lee. Ship detection based on YOLOv2 for SAR imagery. *Remote Sensing*, 11(7):786, 2019.

[2] Papers With Code. Real-Time Object Detection on COCO. https://paperswithcode.com/sota/real-time-object-detection-on-coco?p=yolov7-trainable-bag-of-freebies-sets-new.

[3] Urška Kanjir, Harm Greidanus, and Krištof Oštir. Vessel detection and classification from spaceborne optical images: A literature survey. *Remote sensing of environment*, 207:1–26, 2018.

[4] Jianwei Li, Changwen Qu, and Jiaqi Shao. Ship detection in SAR images based on an improved faster R-CNN. In *2017 SAR in Big Data Era: Models, Methods and Applications (BIGSARDATA)*, pages 1–6. IEEE, 2017.

[5] SIA.org. Commercial satellite industry growing as it continues to dominate expanding global space business – sia releases 25th annual state of the satellite industry report. https://sia.org/commercial-satellite-industry-growing-as-it-continues-to-dominate-expanding-global-space-busines #:~:text=A%20total%20of%201%2C713%20commercial,over%20the%20past%20five%20years.

[6] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. *arXiv preprint arXiv:2207.02696*, 2022.

[7] Shunjun Wei, Xiangfeng Zeng, Qizhe Qu, Mou Wang, Hao Su, and Jun Shi. HRSID: A high-resolution SAR images dataset for ship detection and instance segmentation. *Ieee Access*, 8: 120234–120254, 2020.

[8] SUN Xian, WANG Zhirui, SUN Yuanrui, DIAO Wenhui, ZHANG Yue, and FU Kun. AIR-SARShip-1.0: High-resolution SAR ship detection dataset. , 8(6):852–863, 2019.

[9] Rong Yang, Gui Wang, Zhenru Pan, Hongliang Lu, Heng Zhang, and Xiaoxue Jia. A novel false alarm suppression method for CNN-based SAR ship detector. *IEEE Geoscience and Remote Sensing Letters*, 18(8):1401–1405, 2020.

[10] Tianwen Zhang, Xiaoling Zhang, Xiao Ke, Xu Zhan, Jun Shi, Shunjun Wei, Dece Pan, Jianwei Li, Hao Su, Yue Zhou, et al. LS-SSDD-v1. 0: A deep learning dataset dedicated to small ship detection from large-scale Sentinel-1 SAR images. *Remote Sensing*, 12(18):2997, 2020.