Maritime Ship Detection Using Synthetic Aperture Radar Satellite Imagery

Alexander Makhratchev and Sean Ng

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 and deploy them on Sentinel-1 image data from Google Earth Engine.

Introduction

With recent advances in algorithms and technology within the realm of computer vision, satellite imagery has become an applicable field of study. Roughly 1700 commercial satellites were launched in 2021, bringing the total to 5000 satellites orbiting Earth by the end of 2021. 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 Synthetic Aperture Radar (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. 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. Utilizing deep learning models like YOLOv7 and Faster R-CNN, we were able to produce accurate and rapid results for the task of detecting ships in SAR images.

Related Works

Methods

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 production the most due to the large size of the images as well as the quality and resolution of them.

The data we intend to deploy our model on is the Sentinel-1 SAR image collection on Google Earth Engine. To access this data, we built a helper function that takes in the coordinates of the desired area as well as the start and end dates that the user wants to pull from. The coordinates for this function can be found on the Google Earth Engine Code editor (https://code.earthengine.google.com/) where you create a rectangular bounding box over the desired area of the map and within the console, it will show the coordinates of the bounding box. The helper function will download the set of images locally as .tif files. From there, we use another helper function to resize and then split the image into subimages. The helper function resizes the full images to the next biggest multiple of 800x800 so that each image will cleanly be split up into sub images of size 800x800. The reason why we split the images into subimages of 800x800 is because our inshore-offshore classifier and object detection models predict on images of that size. The helper function finally returns a numpy array of m x n x 800 x 800, where m x n are the grid of sub images generated with each sub image being 800x800.

To build our inshore-offshore classifier, we utilized the Large-Scale SAR Ship Detection Dataset-v1.0 (LS-SSDD-v1.0) dataset to train our model. For training, we used a standard 70/30 train to test split ratio. To find the best model for classification, gathered a few different machine learning models from scikit-learn (K-Nearest Neighbors, SVM, Decision Tree Classifier, and Random Forest Classifier) into a list and looped through each model to test its accuracy. For each individual model, it was fitted and tested on the training and test sets 10 times and the scores for those 10 runs were averaged to give the score for that model. From this, we found that the K-Nearest Neighbors worked the best. Then to find the best parameters for our model, we used 5-fold grid search cross validation. We tested values 1, 2, 5, 10, 15, 20 for the number of neighbors, uniform and distance for the weights parameter, and different algorithms such as ball_tree, kd_tree, and brute. From our cross validation, we found that the best parameters for our KNN model were 5 nearest neighbors, auto for algorithm, and uniform for the weights.

The first class of object detection model we decided to train was YOLOv7. The two models we tried were YOLOv7-W6 and YOLOv7-D6, with 70.4 million and 154.7 million parameters respectively and they are able to do real time inference at 84 and 44 frames per second respectively. These are relatively light-weight models that trade accuracy for speed. During training, we used the train/test split as specified by the author of the dataset. Additionally, for both models we downsized the images to 640 by 640 pixels, used a batch size of 8, and trained for 200 epochs. For the hyper parameters, we used the default ones given in the training script for momentum, learning rate, and optimization function. After training the mode, we experimented with a variety of thresholding values and found that 0.25 worked the best for both models.

The next types of models we trained were Faster R-CNN and RetinaNet. Because we used Pytorch, we had to discard a majority of the images, because they had no objects in them. This brought the number of images from 9k to 1.9k, so we attempted to introduce augmentations, but ran into numerous difficulties when trying to implement them. For both models, we used a pre-trained backbone on the COCO dataset and changed the model head to have two classes (ship and background). The models were trained for 300 epochs with a learning rate of 0.005. A thresholding value of 0.5 was used in order to classify a prediction as positive.

Results

Conclusion

Appendix

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. In quarter 1, we mainly explored the The *You Only Look Once* (YOLO) framework on the Large-Scale SAR Ship Detection Dataset-v1.0 (LS-SSDD-v1.0) dataset. Within the dataset, there was a good mix of images offshore and more inshore near land masses and ports and because of this, the classifiers we trained are able to somewhat generalize to our new task. To increase accuracy, we plan on creating a separate model for inshore and offshore images. To separate the two, we intend to first clean images and then from there we can take the sum or the mean pixel value to classify it. Some other types of object detection models we will be testing include Retinanet and Faster R-CNNs. After experimenting with the models, we will select the best one in order to detect ships in an area of interest. We plan to make a report visualizing our findings over a period of time in a specific geographic area, similarly to this.

Contributions

Alex worked on the object detection part of the project. Everything from organizing the data, researching models, and creating full training and evaluation scripts.

Sean worked on the inshore-offshore classifier part of the project as well as creating the functions to download the Google Earth Engine images.