Automating Offshore Infrastructure & Vessel Identifications Using Synthetic Aperture Radar & Distributive Geoprocessing

by

Brian A. Wong
Dr. Patrick Halpin, Advisor
SkyTruth, Client
Global Fishing Watch, Client
April 27, 2018

Masters project submitted in partial fulfillment of the
requirements for the Master of Environmental Management degree in
the Nicholas School of the Environment
of Duke University

TABLE OF CONTENTS

ABSTRACT

Global Fishing Watch (GFW) recently published the first worldwide industrial fishing effort data set learned from processing 22 billion Automatic Identification System (AIS) observations. Despite quantifying 40 million hours of fishing activity that extended to over 55% of the ocean's surface area in 2016, GFW now aims to quantify fishing effort not captured by current analyses through multimodal remotely-sensed imagery. Such imagery-based vessel identifications are commonly confounded with offshore infrastructure, though, so a global offshore infrastructure data set is first required to disentangle the two. This study first establishes robust and scalable methods for automating offshore infrastructure identifications using synthetic aperture radar in the Gulf of Mexico, and then evaluates the feasibility to adopt these methods for vessel identifications. Results indicate our model identifies offshore infrastructure with a probability of detection of 96.3%, an overall accuracy of 91.9%, a commission error rate of 4.7%, and an omission error rate 3.7%. Additionally, a cloud-native geoprocessing framework using the Google Earth Engine Python API was implemented to automate vessel identifications globally. Over 45,000 SAR images or approximately 100TB of data were processed to build a new database overlaying both SAR-derived and AIS-derived vessel locations.

EXECUTIVE SUMMARY

This study aims to fill critical data gaps required by the client, Global Fishing Watch (GFW), in its work to quantify the ocean's fishing effort. GFW recently published the first global fishing effort dataset learned from processing 22 billion automatic identification system (AIS) observations. Despite quantifying greater than 40 million hours of fishing activity that extended to over 55% of the ocean's surface area in 2016, GFW now aims to quantify fishing effort not captured by current analyses attributable first to low AIS adoption rates, and second, poor AIS coverage, a satellite-side issue.

GFW has explored using three imagery-based solutions to identify fishing vessels and activity from space, including nighttime lights detectable from NOAA's VIIRS satellite, high-resolution 3-meter spectral imagery, and synthetic aperture radar (SAR) imagery from the European Space Agency's Sentinel-1 mission. In all three, however, vessel identifications are commonly confounded with offshore infrastructure, so a global offshore infrastructure dataset is first required to disentangle the two. Moreover, this fits the second client's goals of publishing an annual global offshore infrastructure dataset. The second client, SkyTruth, is an environmental NGO that specializes in using scaled remote sensing for environmental monitoring.

This study first establishes robust and scalable methods for automating offshore infrastructure identification using radar imagery, and then evaluates the feasibility to adopt these methods for vessel identification. More specifically, the study directed its efforts to develop the model in the Gulf of Mexico due to ground truth availability, making for more robust validations. The final model includes five distinct steps for identifying offshore infrastructure. Input data include all Sentinel-1 SAR imagery for 2017 that overlapped the study area, from which the model first produced a Median Composite; second, calculated a Differenced Image between a temporary mean filtered image and the Median Composite; third, applied one of two Thresholding techniques (global and dynamic methods); fourth, cleaned the binary image with an Erosion and Dilation technique; and fifth, reduced connected clusters to its Centroid. Results indicate our top performing model identifies offshore infrastructure with an overall accuracy of 91.9%, a commission error rate of 4.73%, and an omission error rate 3.70%.

A secondary analysis was conducted to characterize model sensitivity to image count availability for the top performing model in each thresholding category (global and dynamic thresholding methods). Constrained to an area of consistent image count throughout 2017, the inputs for the Median Composite stage of the model were systematically reduced one month at a time until only three months of imagery were used. Results indicated that the global thresholding method performed with at least 90% overall accuracy with as little as five months of imagery, whereas the dynamic thresholding method required at least eight months of imagery. This translates to 12 and 19 images, respectively. Most importantly, though, these secondary results provide the client with critical model characteristics when deployed beyond the study site.

With methods established and model sensitivity characterized, the clients requested an evaluation of the methods for offshore infrastructure seen outside the Gulf of Mexico and vessel identifications. First, a wind farm off the coast of Shanghai, China was used to test the model, which detected 161 total wind turbines, 158 of which were spatial matches to a manual ground

truth dataset, leaving 3 false positives, and 0 false negatives. Next, for vessel identifications, a sample SAR image in Indonesian waters was used to for testing purposes. The model identified 107 total vessels, 104 of which were spatial matches to a manual ground truth dataset, leaving 2 false positives, and 3 false negatives. These results were positive enough indicators for the clients to begin scaling the methods, so the Google Earth Engine Python API was used to automate the vessel identification processes. GFW appended the workflow and added AIS-derived vessel locations matched to each individual SAR image extent and timestamp. To date, over 45,000 SAR images or approximately 100TB of data were processed to build a new database that overlays both SAR-derived and AIS-derived vessel locations.

The impetus of this study was initiated by a GFW need of a global offshore infrastructure dataset for internal uses. In the early stages of model development, extracting offshore platform locations for a remarkably small subset region containing only two dozen platforms proved more than challenging, so the final model performance that produced a 91.91% overall accuracy for all the Gulf of Mexico validated to 2,135 ground truth points was considered a significant success by both clients. Furthermore, additional gains were achieved by testing the model's flexibility and implementing cloud-native geoprocessing workflows.

The concluding recommendations for the clients all surround model validation. A strong foundation was established, however, most were geographically constrained. Conducting similar robust validations for both offshore infrastructure outside the Gulf of Mexico and vessel identifications will provide the clients with heightened confidence to deploy the model globally. Additionally, reaching these future milestones will enable broader scientific inquiry, especially with respect to spatiotemporal trends characterizing the legacy of anthropogenic structures throughout the oceans, and characterizing an appended landscape of fishing activity currently undetected.

INTRODUCTION

Fisheries provide both major sources economic value and food security throughout the world, however, its detached nature from land has made consistent monitoring efforts a significant challenge (FAO, 2016). This master's project first client, Global Fishing Watch (GFW), altered this insufficient monitoring paradigm with a novel beta product: the first global fishing effort dataset (Kroodsma et al., 2018). Moreover, GFW, recently published this work in Science where it discovered that fishing activity occurs in over 55% of ocean surface area that equates to four times the extent of terrestrial agriculture (Kroodsma et al., 2018). Temporally, GFW also quantified over 40 million hours of fishing activity for the 2016 calendar year alone.

AIS (Automatic Identification System)

The aforementioned study relies on a technology called automatic identification system (AIS). AIS is a maritime safety mechanism where vessels broadcast a message that identifies the vessel, its geographic coordinates, and approximate travel speed (Tetreault, 2005). These AIS messages are received by satellites in orbit or terrestrial receivers when in close proximity to coastlines. The intent of AIS is to reduce vessel collisions by increasing maritime domain awareness (MDA), however, GFW took advantage of the temporal documentation produced for each vessel in its study (Kocak and Browning, 2015). In particular, intermittent AIS broadcasts by an individual vessel create a track log over time and space for each vessel. When aggregated over a full calendar year that includes every individual vessel's track log, a global AIS message aggregation can be interpreted as a global vessel density map (Figure 1). In GFW's study, the client trained a convolution neural network to learn the temporal behaviors of fishing vessels for both times of active fishing and non-fishing signals (Kroodsma et al., 2018). This machine learning technique was applied to the global vessel density map to classify when each fishing vessel was either actively fishing or not.

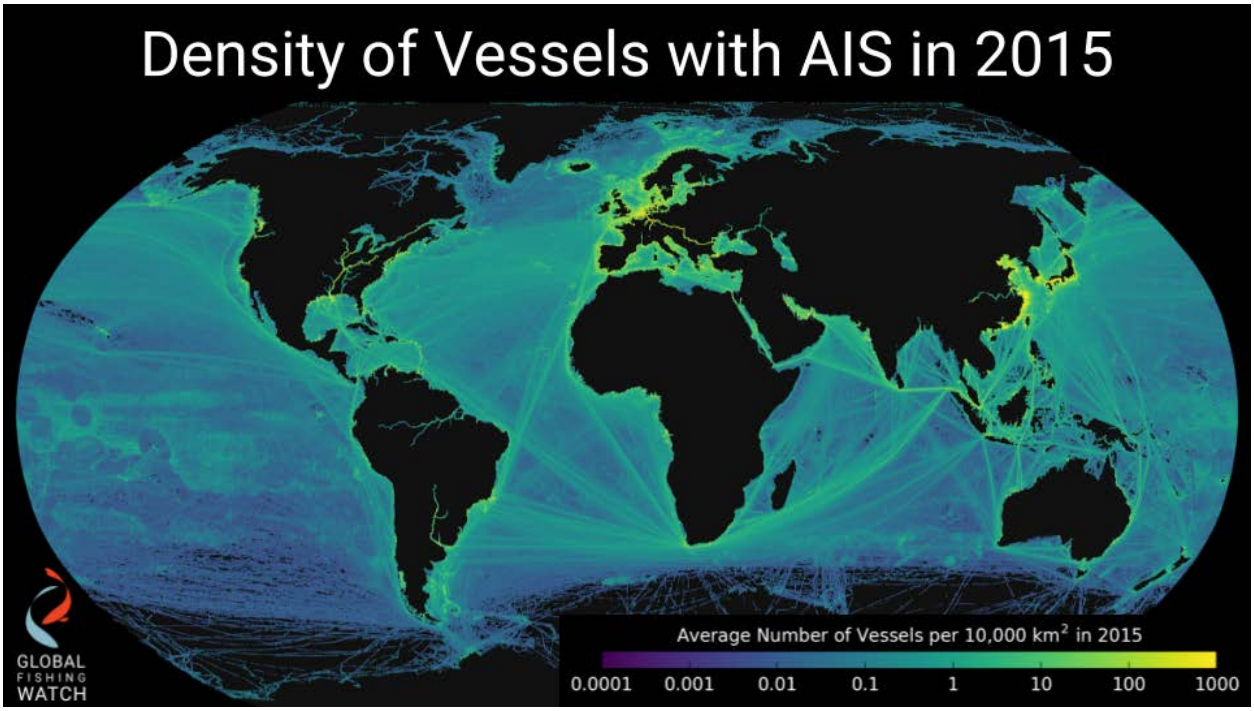

Figure 1. Cumulative global vessel density map for 2015 derived from aggregated AIS messages.

AIS Limitation: Adoption & Coverage

GFW's AIS usage for its study is novel in terms of both scale and accessibility as a fisheries monitoring platform, however, GFW has expressed its limitation and caveats to the current version of its global fishing effort dataset. In particular, GFW highlights two issues, both of which are attributable the AIS technology that the analysis depends on.

First, although AIS is not considered a cutting-edge technology, low AIS adoption rates still persist in many countries, including nations that have considerably large fishing fleets (Kroodsma et al., 2018). GFW highlights each country's exclusive economic zones (EEZ) where less than half of its fishing fleet actively uses AIS technology (Figure 2a). These highlighted EEZ's represent the flag state for which a vessel belongs to even though these vessels are distributed globally in their spatiotemporal activity. And second, GFW highlights a satellite side issue where signal interference and AIS receiving channel limitations cause poor AIS coverage areas throughout the world, most notably in Southeast Asian waters where large fishing vessel fleets are known to be actively fishing (Figure 2b). In particular, the channel limitations of AIS

receiving satellites are due to a maximum total number of AIS messages the satellite can receive at any given time, thus areas of high vessel traffic oftentimes broadcast more AIS messages than the satellite can record (Cervera and Ginesi, 2008; Clazzer et al., 2014).

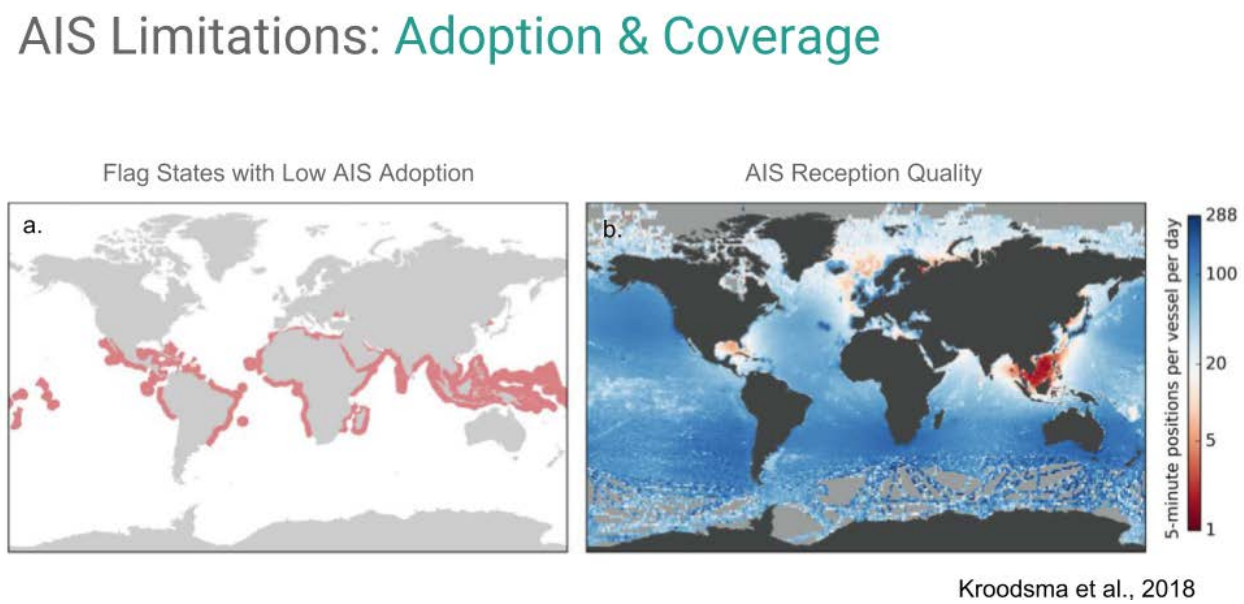

Figure 2. GFW identified limitations to its fishing effort dataset attributable to low AIS adoption rates and insufficient AIS coverage.

The low AIS adoption rate issue highlights particular instances where GFW is confident that its model confidently predicts fishing activity when AIS data is available, however, cannot make any prediction when a vessel does not broadcast an AIS message. On the other hand, the satellite side issue highlights particular instances where GFW is once again confident that its model confidently predicts fishing activity when AIS data is available, however, cannot make any prediction when these AIS messages are not properly recorded by the AIS receiving satellite.

Imagery-Based Solutions

Given these limitation, GFW has pursued three imagery-based solutions to fill critical data gaps for future work. In particular, GFW aims to identify fishing vessels from satellite imagery and find the differences between imagery-derived fishing vessel counts and AIS-derived fishing vessel counts. This inquiry aims to better understand the difference between the two fishing vessel identification pathways and characterize fishing vessel density not currently captured by AIS data.

For the first of three imagery-based solutions, GFW partnered with National Oceanic and Atmospheric Administration (NOAA) to correlate fishing activity to nighttime lights irradiance data derived from its Visible Infrared Imaging Radiometer Suite (VIIRS) satellite (Figure 3a; Global Fishing Watch, 2018). Oftentimes, fishing vessels use bright lights to attract fish at night insomuch that these vessels are visible from space (Elvidge et al., 2015; Liu et al., 2015). Second, GFW also uses high resolution imagery data from a silicon-valley startup whose mission is to image the earth daily (Figure 3b). In particular, Planet specializes in high-cadence and high-resolution optical imagery that can be used for vessel identifications. Lastly, GFW researched using synthetic aperture radar (SAR) imagery from the European Space Agency's (ESA) Sentinel-1 satellite, also for vessel identifications (Figure 3c).

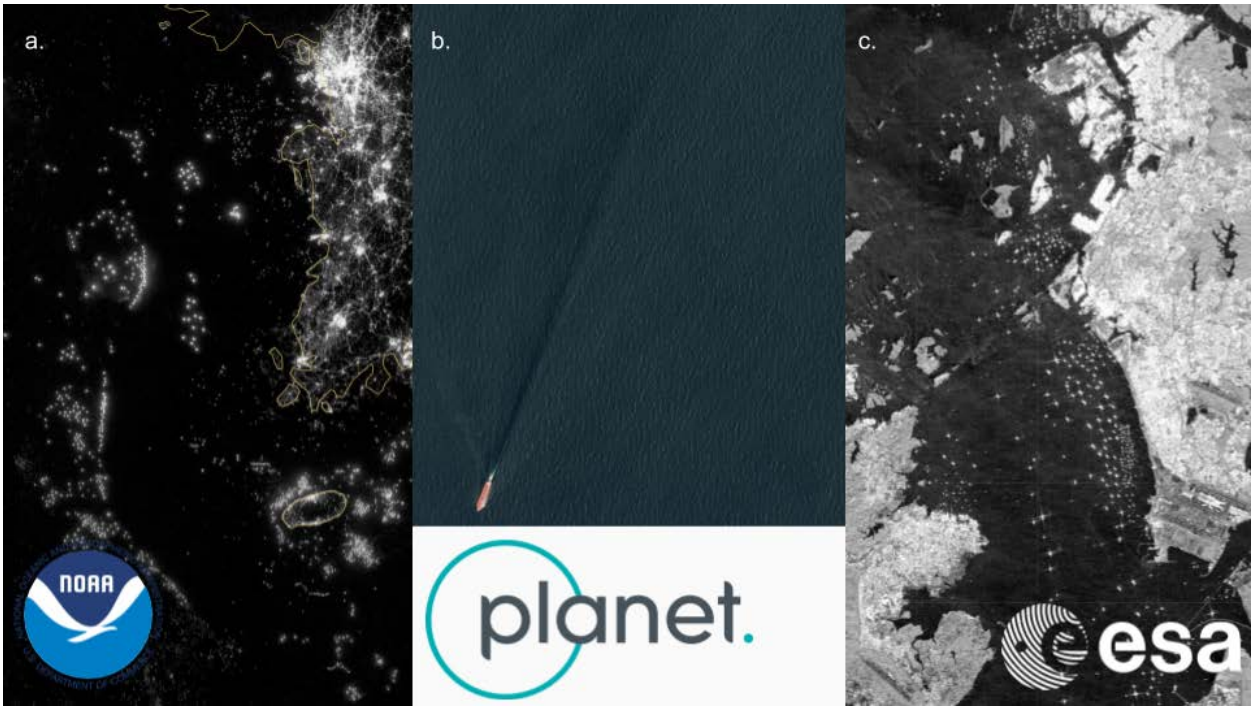

Figure 3. The three imagery-based solutions that GFW is pursuing to identify vessels from space.

Objectives

The client has pursued the aforementioned imagery-based solutions to characterize fishing vessels not captured by AIS, however, the client has expressed a particular challenge in

these pursuits. In all three imagery-based pathways, GFW is confident fishing vessel locations can be extracted, however, confounding identifications with non-fishing vessels and offshore infrastructure (e.g. oil platforms, wind turbines, aquaculture) can distort future analyses. To address this, GFW identified the immediate need of a global offshore infrastructure dataset. This would allow for GFW to use these data as a global mask when identifying vessels from the three imagery-based sources.

Producing a global offshore infrastructure dataset is beyond the scope of this master's project, however, establishing foundational methods employable by GFW is not. Moreover, the client identified the Gulf of Mexico as a suitable case study site due to ground truth data availability provided by Bureau of Ocean Energy Management (BOEM) that identifies all oil platforms within U.S. waters in the Gulf of Mexico (BOEM, 2018).

GFW's interest is for internal use of a global offshore infrastructure dataset, however, one of its founding partners and the second client for this master's project, SkyTruth, is interested to publish an annual global offshore infrastructure dataset. SkyTruth is an environmental NGO that focuses on scaled environmental monitoring through remote sensing and oftentimes produces cleaned datasets consumed by other NGOs, advocacy groups, and academics for research related to environmental impact and human health effects.

Additionally, GFW has made progress using VIIRS data from NOAA and high-resolution spectral imagery for vessel identification projects, however, has not made significant gains using SAR imagery, thus have requested this master's project to focus on leveraging the SAR imagery pathway. As such, the primary objectives of this study are to:

1. Establish scalable and robust methods for offshore infrastructure identification using SAR imagery from ESA's Sentinel-1 satellite.
2. Evaluate feasibility to adopt methods for open ocean vessel identification.

GFW & SkyTruth define scalable as deploying the concluding methods globally and robust as an overall accuracy of greater than 90% for this particular project phase.

MATERIALS & METHODS

Data Sources: Imagery and Ground Truth

For imagery, Sentinel-1 SAR images were used for the Gulf of Mexico study site. We filtered the imagery to include 1) all imagery for the 2017 calendar year, 2) overlapped with extent of ground truth data from BOEM, and 3) contained the VH-band. In total, 541 SAR images or approximately 1.2TB of compressed data were used for the study. Ground truth used to validate the model was access from BOEM. The "Platforms" dataset was downloaded from BOEM's Mapping Data Center online (BOEM, 2018).

Data used for the feasibility study (Objective 2) also used Sentinel-1 SAR imagery, however, was not constrained to the Gulf of Mexico. Instead, it was geographically constrained to all the oceans and temporally constrained from December 15, 2017 to March 13, 2018 as of the time of this study. This totaled approximately 45,000 SAR images and 100TB of data.

Tools & Software

The primary tool to conduct the geospatial analyses and geoprocessing was Google Earth Engine. Traditional pathways for desktop geoprocessing like ArcGIS were not feasible due to the significant data requirements and heavy computational costs. Moreover, downloading the data would not only have been arduous, but not scalable for future client needs, so constructing a cloud-native workflow was all but required from the project's initiation. Google Earth Engine (GEE) is a cloud-based open source geoprocessing platform that combines remote sensing with distributive computing (Gorelick et al., 2017). One of the most significant benefits to using GEE is its intermittent downloading of all major open source satellite imagery to its servers, which allows for instant access to these imagery in near real-time. This benefit translates to removing the imagery download step that is oftentimes a significant burden for scaled remote sensing applications (Gorelick et al., 2017).

A secondary toolkit was used to evaluate vessel identification feasibility and scaled geoprocessing efforts. In particular, a Python script was developed to automate distributive geoprocessing, which also accessed Google BigQuery and Google Cloud Storage. The former is a cloud-based data hosting service that specializes in scaled analytics for massive datasets (e.g. CSV files too large to fit on desktop or too slow to access from servers). The latter is the hosting service where the new dataset is being constructed.

Ground Truth Preprocessing

Although BOEM freely provides one of the most detailed and large offshore infrastructure datasets publically available, cleaning of this ground truth was required. The dataset explicitly includes both active and inactive platforms, so the latter needed removal considering these were identified as no longer in existence and completely removed. The complete dataset included 7,291 total oil platforms with 4,753 inactive sites, leaving 2,538 active sites. Of the active sites, many instances occurred where multiple connected platforms were counted as separate platform sites (Figure 4), however, BOEM informed my client and I that these connected platforms were actually a part of the same complex. As such, and given GFW and SkyTruth's needs, the 2,538 active sites were aggregated by common Complex ID values, leaving 2,110 ground truth locations.

Figure 4. Example oil platform that BOEM unique ID labels 4 separate platforms (blue dots) whereas SkyTruth and GFW needs aggregated these by common Complex ID values (red dot).

Model Methods

The geospatial model to identify offshore infrastructure in the Gulf of Mexico study site contains five distinct stages described below. The full model scripts to execute these analyses are found in the three Google Earth Engine scripts (Appendix A).

Median Composite

First, all imagery meeting the data filtering requirements were aggregated into a Google Earth Engine Image Collection computable object. This essentially stacks every individual image into a collection of layers that allows for GEE function calls to be applied to the Image Collection object, all the while maintaining geographic properties. Additionally, commonplace radar imagery preprocessing techniques were applied in this stage. This included applying ocean-parameterized angle corrections and transforming the image units from decibels to the radar unitless "natural units" (Topouzelis et al., 2016). Angle corrections are necessary for radar imagery due to its off-nadir signal sent from the satellite (Mladenova et al., 2013; O'Grady et al., 2013). This side-swath produces large ranges in pixel sizes, where pixels closer to the satellite are smaller and pixels farthest from the satellite are largest. Applying the angle correction normalizes all pixels of an image to the same pixel size. And transforming the pixel data back to "natural units" is also a radar commonplace practice when planning to manipulate image data for analyses. After applying these two preprocessing steps each individual image, the median value of each pixel across the entire Image Collection was selected to replace and create a new single layer for the study site. This is the Median Composite, which first acts as a temporal filter, removing all mobile objects, especially vessels as seen in Figure 5. Additionally, a secondary benefit was the optically visible reduction in image noise.

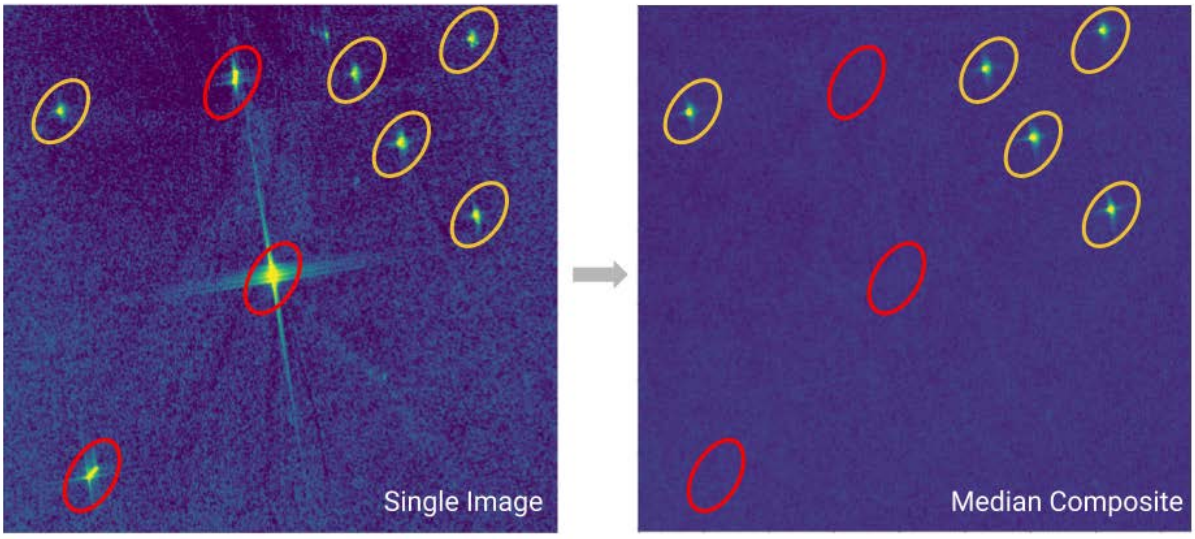

Figure 5. Demonstration of how a median composite also acts as a temporal filter, removing mobile objects (vessels) in the red ovals, whereas stable objects (platforms) remain in both the Single Image and Median Composite seen in the yellow ovals.

Image Differencing

The Median Composite provided great benefits in terms of acting as a temporal filter and reducing image noise, however, it also introduced image artifacts into the model. In particular, sharp artifacts are seen across the Median Composite in areas where images overlap spatially, creating start streaks visually (Figure 6a). These visual streaks, however, represent sudden changes in the data that need correction, so an image differencing technique was applied to adjust the Median Composite for these image artifacts.

To do so, a temporary Mean Filtered Image was first created using a circular 500m moving kernel (Figure 6b). This kernel moves a 500m circle across the Median Composite one pixel at a time, calculates the mean values of all pixels within this kernel, and creates a new image replacing the old value with this new local mean value, thus creating the Mean Filtered Image (Figure 6b). This Mean Filtered Image was then subtracted from the Median Composite, pixel for pixel, which then output the Differenced Image (Figure 6c). Visually, the stark differences between each side of the image artifact left in the Median Composite is now all but removed. That said, a residual artifact still remains visible, although far less an issue.

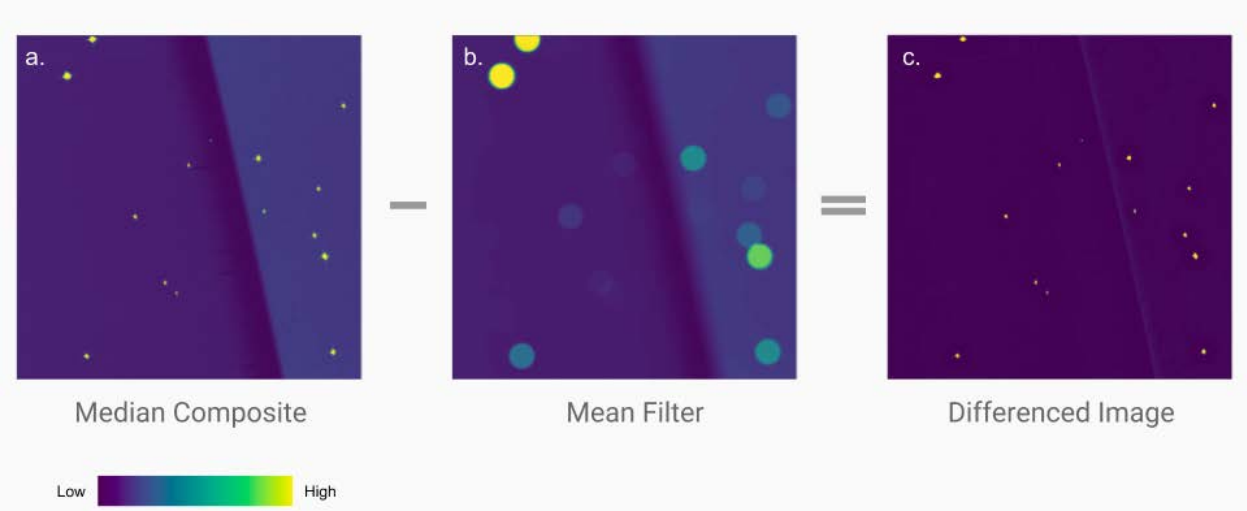

Figure 6. Image differencing technique applied to remove artifacts created in Median Composite step.

The image manipulations up to this point all feature two primary goals: to minimize the background noise of the water, and maximize the signal of the offshore infrastructure to be extracted. The step-by-step adjustment provide nuance to achieve this, however, the most easily interpretable path is to visualize the data in 3D, seeing the drastic changes from what a single image looks like to what the Differenced Image looks like for sample region (Figure 7).

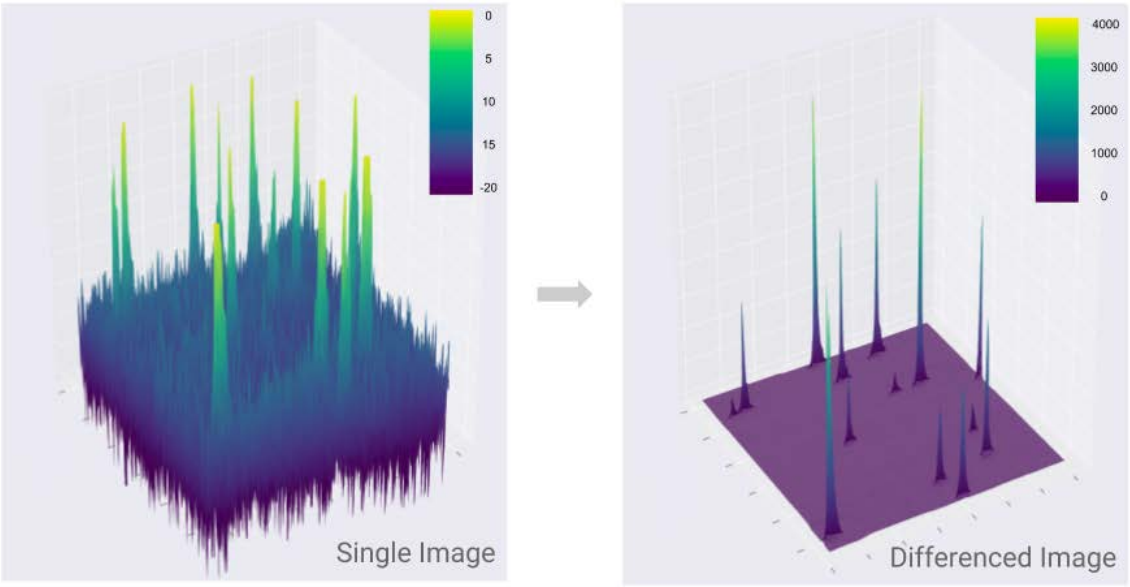

Figure 7. The data cleaning seen from processing each individual image to the Differenced Image step.

Thresholding

What the Difference Image allows for next, is to apply a thresholding technique to the Differenced Image (Pal and Sankar, 1993). By minimizing and normalizing the water in the image to itself, and maximizing the offshore infrastructure signal, the model can now step into this data space and slice through it to classify pixels to belong to either water or offshore infrastructure. Two thresholding method types were applied. First, a global thresholding used a single value as a cut-off value, where any pixels above a selected value were grouped together as potential pixels belonging to offshore infrastructure. The second method, dynamic thresholding, used a relative value for this cut-off. More specifically, a multiplier (e.g. two times) of the previously created Mean Filtered Image was used to determine dynamic cutoff values across the Differenced Image. The resulting image from both the global thresholding and dynamic thresholding methods are the same. A binary image arbitrarily labeled with ones and zeroes was produced, where ones were pixels predicted to be part of an offshore platform (Figure 8).

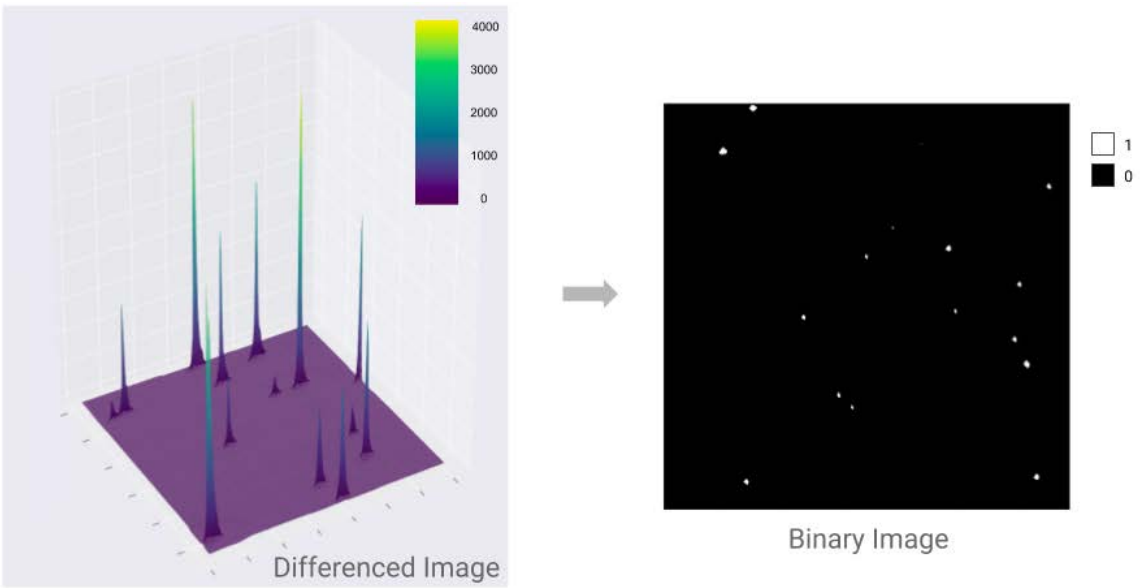

Figure 8. The cleaned data seen in the Differenced Image allows for thresholding application, resulting in a binary image.

Erode & Dilate

Even after these significant data manipulations of the model, residual noise may still occur at this stage. The full study site of the Gulf of Mexico is too large to observe this issue, however, zoomed into an individual platform in the Thresholded Image shows pixels predicted to be part of an offshore platform detached from the cluster that it should be connected to (Figure 9a). To address this issue, an image processing technique called erosion and dilation, which is also synonymous with opening and closing, was used to clean erroneous pixels predicted to be part of an offshore platform (Haralick et al., 1987). These would otherwise create false positives if not addressed.

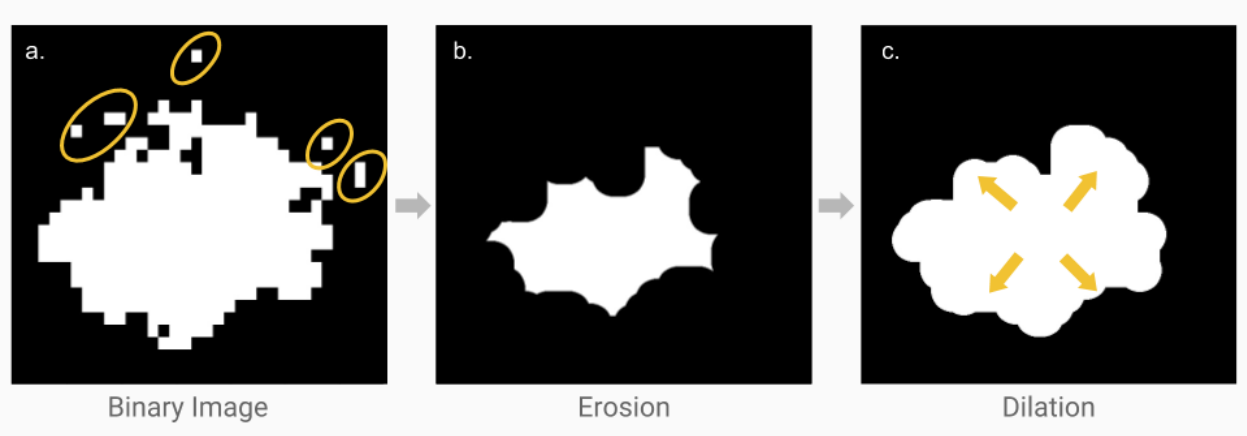

Figure 9. The erosion and dilation steps of the model help reduce false positives otherwise triggered by stray pixels.

First, an erode function was applied to all connected clusters of pixels predicted to be part of an offshore platform. Functionally, this means that for each cluster of offshore platform pixel predictions, a negative buffer was applied from all its edges (Figure 9b). The key parameterization for this function is the distance of the erosion. A range of 5-10m was used. A 5m erosion removed all isolated single pixels, given the imagery were 10m resolution. Accordingly, a 10m erosion would functionally remove a cluster of four pixels. The risk of an erosion function, however, is the potential to unintentionally separate a connected cluster of pixels from a previously attached cluster, so a dilation was then applied to re-connect these in such a scenario (Figure 9c).

Reduce to Centroid

The simplest stage of the offshore infrastructure extraction model is the centroid retrieval stage. The Thresholded Image that was cleaned using the erosion and dilation technique is still a raster-based image, however, a single geographic point is needed for the model output so a GEE Reduce to Centroid function was used to make the conversion. More specifically, this function was constrained to deploy only on clusters predicted to be part of an offshore platform, and for each one of these clusters, the centroid (or center point) was calculated and retrieved in longitude and latitude format (Figure 10c). This was deployed for the full extent of the study site, outputting a GeoJSON of predicted offshore platform locations. This GeoJSON was downloaded, then converted to a Shapefile using command line GDAL applications, then uploaded into GEE, and finally visualized and rendered for accuracy assessment purposes.

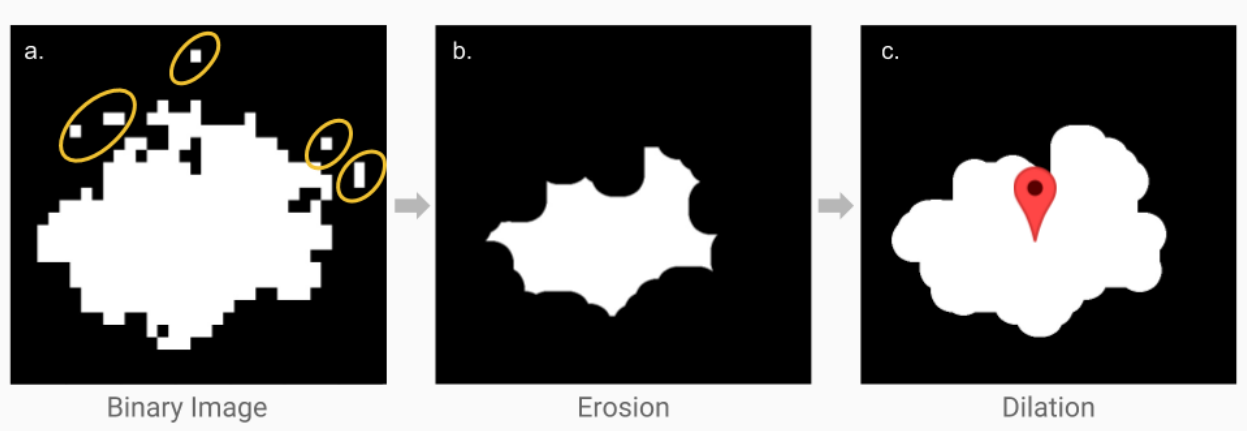

Figure 10. The last part of the model converts the binary raster image into a point feature class by retrieving the centroid (red drop point) of each connected cluster of pixels after the dilation step is complete.

Model Methods: **Vessel Identification Version**

The full offshore infrastructure scripts are fully displayed in Appendix A, however, the client also requested the evaluation of these methods for vessel identifications from SAR imagery. Although this is not the primary methods establishment portion of the project, full script documentation is available to demonstrate feasibility (Appendix B**)**. Much of the model used to extract offshore platforms in the Gulf of Mexico remains intact for vessel identification as well, with only three minor things changed. First, there is no Median Composite image step

considering this analysis is conducted on a single image. Second, the thresholding stage was limited to only the global thresholding method due to client needs. And third, the thresholding value was constrained to be much more conservative due the increased level of background noise in a single image. Otherwise, the vessel identification model remains mostly identical to the offshore infrastructure identification model.

RESULTS

Ground Truth Post-Processing

The ground truth cleaned BOEM offshore platform dataset with 2,110 sites needed one further cleaning step applied after the model was run. Recall, the BOEM dataset was filtered for only active sites with common Complex ID values to isolate only one location per connected array of structures. The model found platforms or other large permanent objects not in the BOEM dataset, thus causing a false positive assessment when in fact, it was not. As such, the SkyTruth team worked to manually check all false positives by using a combination of Sentinel-2 10m spectral imagery and Sentinel-1 10m SAR imagery. Full SkyTruth documentation for ground truth post-processing is available in Appendix C. This resulted in an appended ground truth dataset that increased the number of ground truth sites from 2,110 to 2,135.

Accuracy Assessment I

Twenty-four versions of the model laid out in the methods section were deployed. Of these, twelve used the global thresholding method while the other twelve used the dynamic thresholding method. The changes among the twenty-four models were from fine-tuned adjustments to model parameterization. The variables that were slightly adjust for include the distance for erosion in meters, distance for dilation in meters, cutoff value for global thresholding model, range of multiplier for dynamic thresholding model, and nominal scale for which to run the model. For the lattermost, SAR imagery from Sentinel-1 has 10m pixel resolution, however, experimentation with down-sampling to 20m pixels for processing speed purposes led a small discovery that this was worth leaving in the pool of test models.

Accuracy assessments are more typically conducted classification performance with raster imagery, however, this was not feasible given the point data output and ground truth. As such, spatial matching was used to conduct the accuracy assessment. To do so, a buffer was constructed for each ground truth point, and if a predicted point from the model output was within this buffer, it was considered a spatial match. A 150m buffer was used to make the study directly comparable to the performance of another recent study that also looked to identify offshore platforms in the Gulf of Mexico (Liu et al., 2016). The aforementioned study used a 150m buffer. The full results of all twenty-four models are seen in a full accuracy assessment testing every prediction against the full ground truth dataset (Figure 11). The overall accuracy of each model is function of both its commission error rates and omission error rates, that being false positives and false negatives, respectively. False positives in this context were instances where the model predicted and output the location of platforms that were incorrect. In other words, there was no platform where this prediction occurred. On the flip side, a false negative in this context were instances where the ground truth dataset contained a buffered 150m point of platform that the model failed to find and predict its presence.

	Accuracy Assessment									
	Model	Total Reference	Total Predicted	True Positives	False Positives	False Negatives	Commission Error	Omission Error	Probability of Detection	Overall Accuracy
Dynamic Thresholding Models	1	2135	2134	2026	108	109	5.06%	5.11%	94.89%	90.33%
	2	2135	2084	2009	75	126	3.60%	5.90%	94.10%	90.90%
	3	2135	2045	1987	58	148	2.84%	6.93%	93.07%	90.61%
	4	2135	2012	1953	59	182	2.93%	8.52%	91.48%	89.02%
	5	2135	2134	2026	108	109	5.06%	5.11%	94.89%	90.33%
	6	2135	2084	2009	75	126	3.60%	5.90%	94.10%	90.90%
	7	2135	2075	2011	64	124	3.08%	5.81%	94.19%	91.45%
	8	2135	1967	1916	51	219	2.59%	10.26%	89.74%	87.65%
	9	2135	1946	1892	54	243	2.77%	11.38%	88.62%	86.43%
	10	2135	1924	1871	53	264	2.75%	12.37%	87.63%	85.51%
	11	2135	1906	1854	52	281	2.73%	13.16%	86.84%	84.77%
Global Thresholding Models	12	2135	2070	2007	63	128	3.04%	6.00%	94.00%	91.31%
	13	2135	2032	1970	62	165	3.05%	7.73%	92.27%	89.67%
	14	2135	2158	2056	102	79	4.73%	3.70%	96.30%	91.91%
	15	2135	2113	2031	82	104	3.88%	4.87%	95.13%	91.61%
	16	2135	2079	2012	67	123	3.22%	5.76%	94.24%	91.37%
	17	2135	2050	1986	64	149	3.12%	6.98%	93.02%	90.31%
	18	2135	2024	1963	61	172	3.01%	8.06%	91.94%	89.39%
	19	2135	2008	1950	58	185	2.89%	8.67%	91.33%	88.92%
	20	2135	1966	1909	57	226	2.90%	10.59%	89.41%	87.09%
	21	2135	1931	1873	58	262	3.00%	12.27%	87.73%	85.41%
	22	2135	1884	1837	47	298	2.49%	13.96%	86.04%	84.19%
	23	2135	1869	1822	47	313	2.51%	14.66%	85.34%	83.50%

Figure 11. Full accuracy assessment for all 24 models evaluated separated by thresholding type.

The range of overall accuracy ranged from 83.50% - 91.91%. The commission error rate ranged from 2.49% - 5.06%. And the omission error rate ranged from 3.70% - 14.66%. The top performing global thresholding model had an overall accuracy of 91.91%, 4.73% commission error rate, and 3.70% omission error rate. The top performing dynamic thresholding model had an overall accuracy of 91.45%, 3.08% commission error rate, and 5.81% omission error rate. The top performer overall with 91.91% overall accuracy is visually seen below in Figure 12.

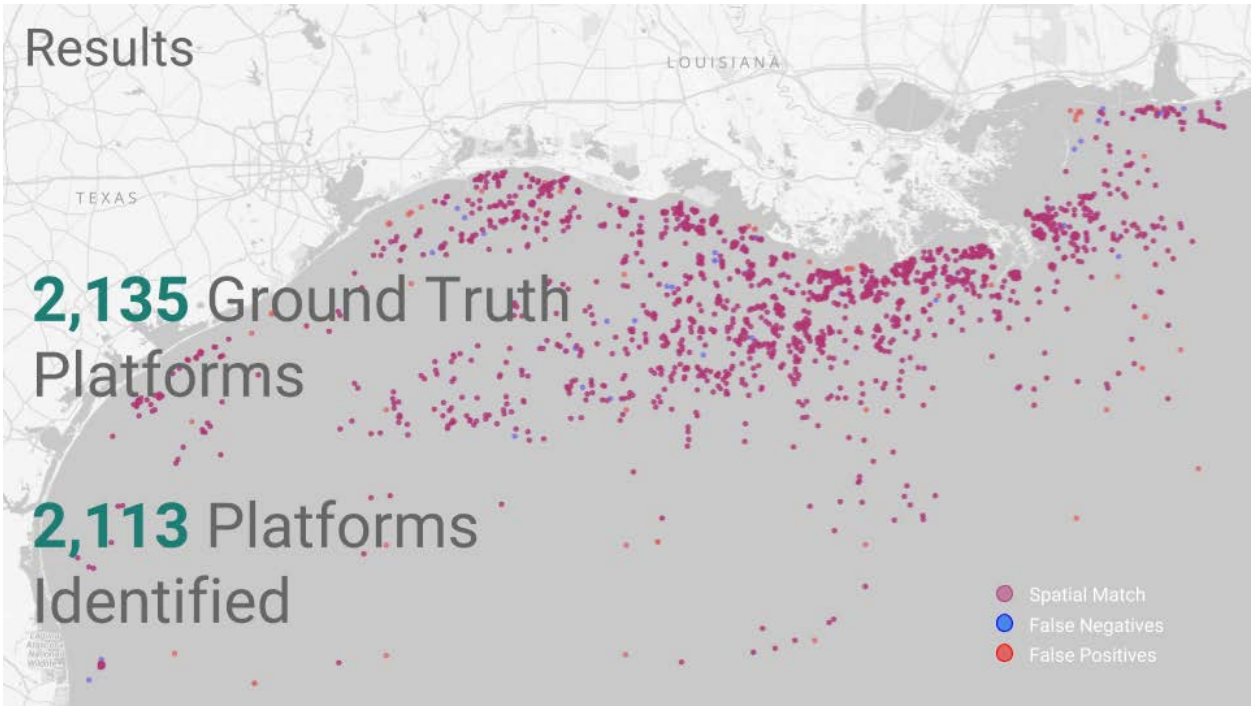

Figure 12. Mapping results of top performing model output.

Accuracy Assessment II: Sensitivity Analysis

The first accuracy provided positive takeaways. The top performing model from each category were between 91.45% - 91.91% accurate, which exceeded the initial SkyTruth goal of greater than 90% overall accuracy. However, both GFW and SkyTruth intend to deploy the model beyond the Gulf of Mexico, so further characterization of the model, especially its caveats were also needed. The primary uncertainty factor when deploying the model globally, was image count availability. In the Gulf of Mexico alone, the image count availability for 2017 alone ranged from 12-88 images (Figure 13). This uncertainty was expected to persist elsewhere too,

so an image count sensitivity analysis was used to equip the clients with an understanding of the model's limits. The secondary assessment required further geographic constraints as well, however. Given the image count variation across the Gulf of Mexico, it was not feasible to include all the ground truth points in this analysis. That said, there were two image pathways that for each month of the year had either the same exact number of images or were difference by just one image. Additionally, this are highlighted by the white arrows, contained nearly three-quarters of the ground truth points so was considered the ideal area to conduct a sensitivity analysis.

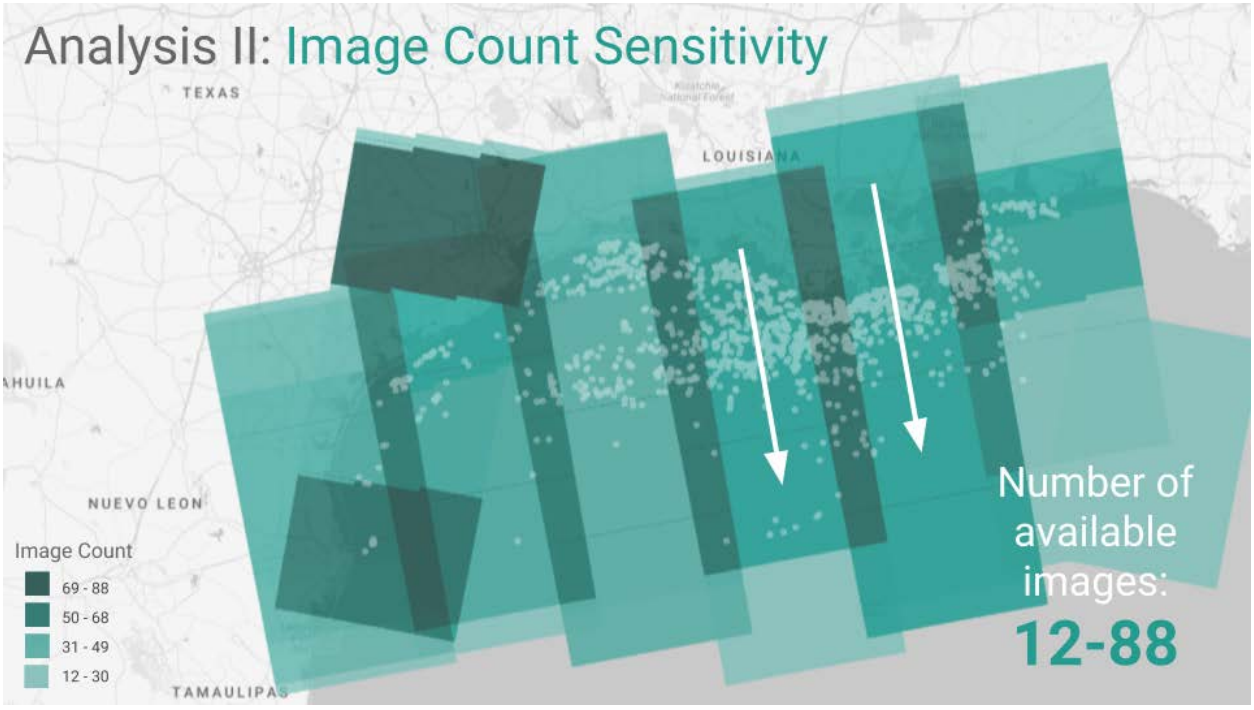

Figure 13. Image count variation across Gulf of Mexico for 2017 demonstrate need for sensitivity analysis.

The image count sensitivity analysis entailed a monthly step down of a temporal window for the model to identify offshore platforms from. For example, the starting point was the full year, or twelve months, of imagery from which a Median Composite was constructed that started the first of five steps in the model to create an output of prediction points. The monthly step down application reproduced this full analysis with one key change: the twelve month window was reduced to eleven months. This monthly step down method was repeated over until there were only three months of imagery of left in the input for the model. Moreover, this step down

method was applied to the top performing model from both the global and dynamic thresholding models.

The results of this image count sensitivity analysis show that the global thresholding model was more robust and less sensitive to a shrinking time frame of image availability (Figure 14). For the dynamic thresholding model, the least number of months for which the model still predicted an overall accuracy greater than 90% was eight months of imagery. This equated to 19 images. For the global thresholding model, the least number of months for which the model still predicted an overall accuracy greater than 90% was five months of imagery. This equated to 12 images. In both models, nearly all its accuracy performance drop-off is attributable to commission error, whereas omission error rates do not increase significantly across the full temporal window change.

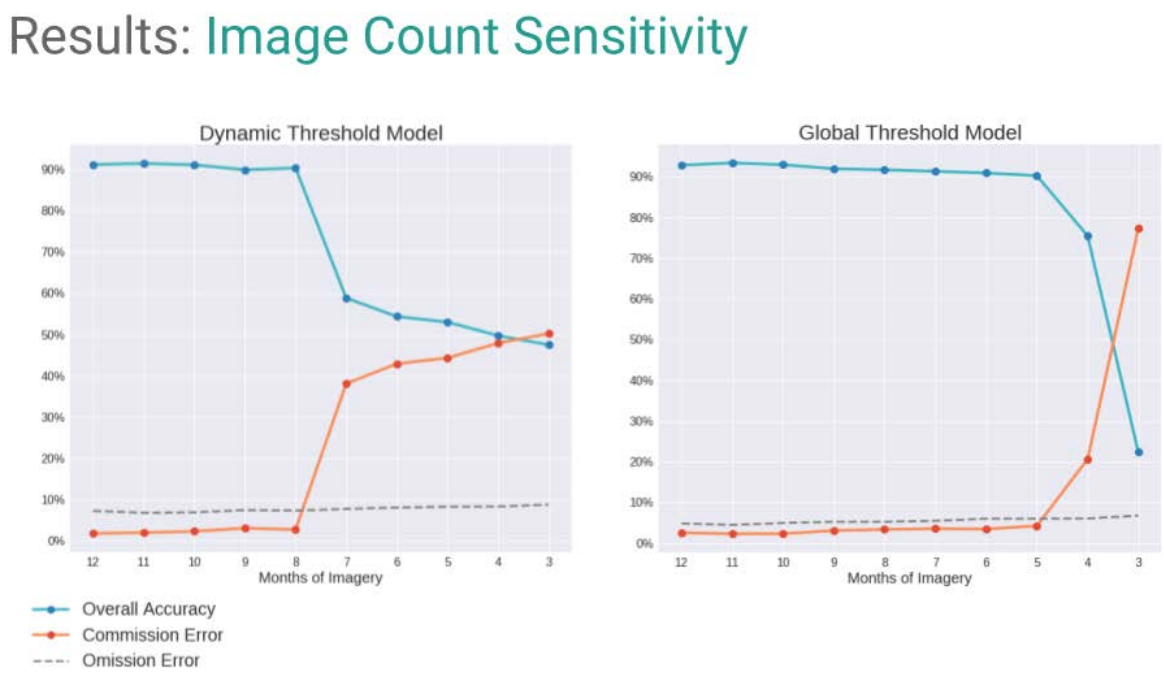

Figure 14. Results of image count sensitivity analysis show that the global thresholding model requires 12 images to retain an overall accuracy of 90% or greater, whereas the dynamic thresholding model requires at least 19 images to do the same.

DISCUSSION

Major Findings

The major finding for this project was the establishment of a robust and scalable offshore infrastructure identification model. Moreover, the characterization of its sensitivity for the client's intent to use it as a general-purpose model beyond the Gulf of Mexico was a key supporting factor as well. Furthermore, to the best of our knowledge, this is the first offshore infrastructure identification model deployed for the Gulf of Mexico that performed with over 90% overall accuracy when validated to a full ground truth dataset (Liu et al., 2016; Marino et al., 2017). Lastly, the large study area forced the model to account for a large geographic region, however, was still constrained to a region of the planet, so further discussion is warranted with respect to different types of limitations moving forward.

Previous Studies

There is growing literature for using SAR imagery for vessel identifications, however, applications for offshore infrastructure are particularly limited (Greidanus et al., 2017; Liu et al., 2016; Marino et al., 2015; Marino et al., 2017; Stasolla et al., 2016; Yang et al., 2013). The only offshore infrastructure identification study applied to the Gulf of Mexico was conducted by Liu et al. 2017, however, spectral imagery from Landsat-8 was used instead. Moreover, the model struggled with relatively high false positive rates and validated the model in a constrained subset region of the Gulf of Mexico that contained 201 platforms. These validated predictions performed very well (196 spatial matches, 4 false positives, 1 false negative), however, combined for less than 10% the total number of sites throughout the Gulf of Mexico. This was not ideal for the clients, GFW and SkyTruth. In comparison, the top performing model from this master's project reported an overall accuracy of 91.91% that was validated against 2,135 ground truth points. The clients believe the model performance metrics conducted on over ten times the number ground truth points provided more confidence to use it a general purpose model beyond the initial study site.

Future Work

Given the success of the project, the clients requested preliminary feasibility tests applied at the earliest convenience. In particular, three ad hoc analyses were conducted. First, deploying the offshore infrastructure model on non-oil platforms was tested. Second, adjusting and evaluating the models agility to detect vessels instead of platforms was tested. And third, constructing a cloud-native framework for comparing these SAR-detected vessels versus AIS-detected vessels was built.

Future Work: Wind Turbine Example

Both clients intend to use the offshore infrastructure model on non-oil platforms and outside the Gulf of Mexico. Future work will include wind turbines, aquaculture, buoys, and any anthropogenic objects that are detectable from satellite imagery. Here, an ad hoc analysis was conducted for the clients to demonstrate potential results and provide early insights for future work. A wind farm was identified just off the coast of Shanghai, China where a total of 158 wind turbines were visible from Sentinel-2 spectral imagery (Figure 15a). The model detected 161 total wind turbines, 158 of which were spatial matches to the manual ground truth, leaving 3 false positives (Figure 15b). This preliminary test indicates potential model viability for deploying the algorithm beyond the Gulf of Mexico as well as for non-oil platform objects, however, full validations for both are likely needed to establish robust methods.

Figure 15. The offshore infrastructure model deployed on a wind farm near Shanghai, China resulted with 3 false positives and 0 false negatives for a Sentinel-2 image containing 158 wind turbines.

Future Work: Vessel Detections

A second ad hoc analysis was conducted for the clients to indicate potential vessel identification applications for the model. Here, a Sentinel-1 SAR image in Indonesian waters was used to test the model's viability for vessel identification (Figure 16a). The model detected 107 total vessels, 104 of which were spatial matches to the manual ground truth, leaving 2 false positives and 3 false negatives (Figure 16b). This is a good indicator for the model's adaptability, however, significantly more robust validation processes are needed in the future to establish reliable methods.

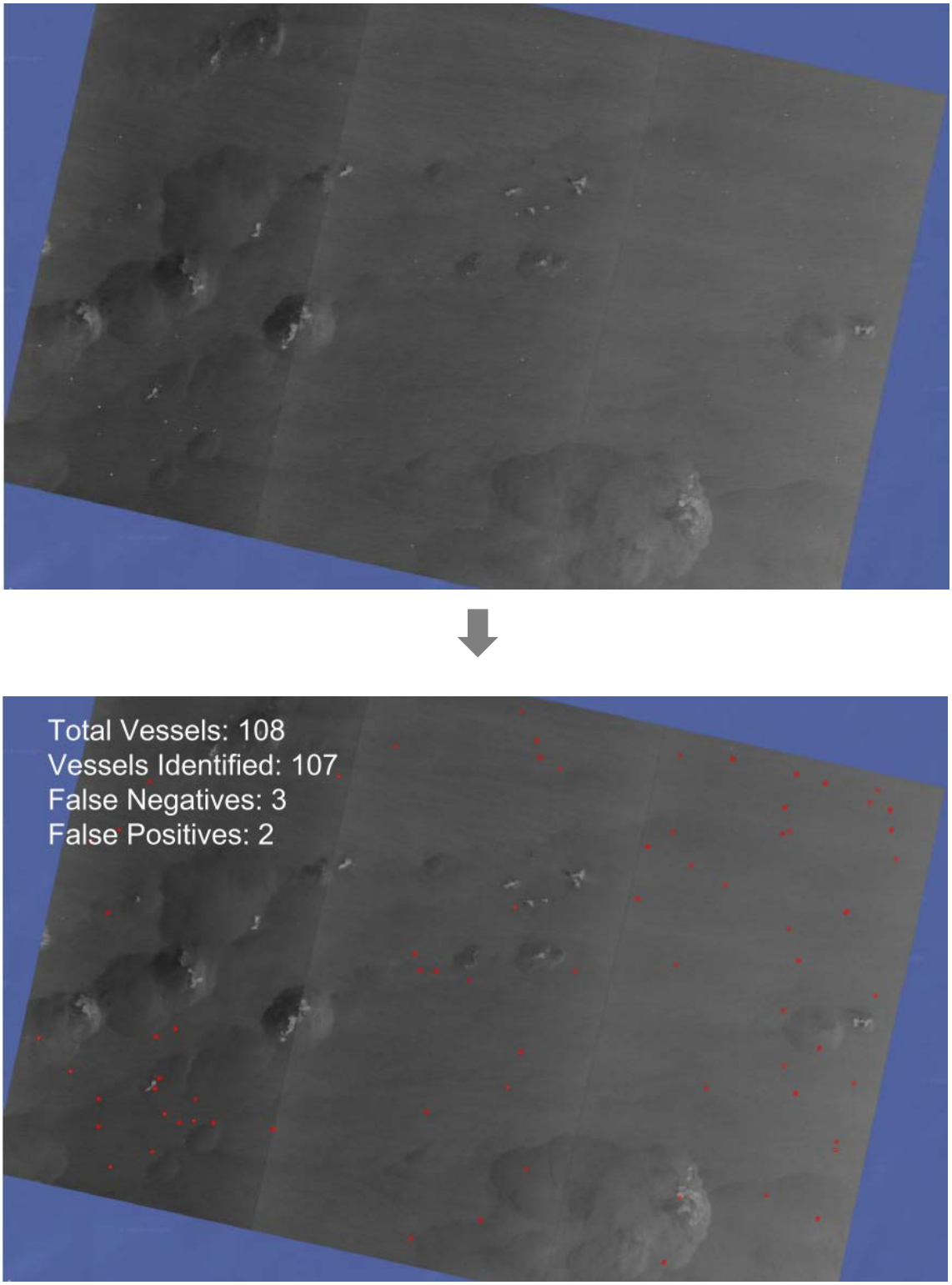

Figure 16. The model adjusted for vessel identifications.

Future Work: Vessel Detection & AIS Matching Database

The third major source for future work adds one additional layer to the SAR vessel detections methods. GFW is an industry leader in deriving information from raw AIS data, and given the long-term aim for SAR vessel identifications is to identify fishing vessels not seen in AIS, the logical next step is to add a layer of AIS-derived vessel locations on top of the SAR-derived vessel locations. This provides immediate and early insights into where scaled versions of this analysis might go in the future for GFW.

Here, a Sentinel-1 SAR image off the coast of China in the Yellow Sea was used to visualize the overlay of both SAR-derived vessel locations and AIS-derived vessel locations (Figure 17). 129 and 132 vessels were identified in this image, respectively, however, the clear differences between the distributions of these locations is the item of interest to the clients. Recall, GFW highlighted two caveats to its global fishing effort dataset: first, low AIS adoption rates and second, poor AIS coverage areas. This example image is in a shipping lane, so most likely indicates that the lack of AIS-derived vessels in the southern portion of the image stemmed from poor AIS coverage. That said, this is only one image of approximately 400-500 Sentinel-1 SAR images captured each day globally where at least a portion of the image overlaps with the oceans, if not completely. As such, employing distributive geoprocessing methods were needed to scale the analyses. To do so, the Google Earth Engine Python API was used to automate the process seen in Figure 17**.** To date, over 45,000 SAR images or approximately 100TB of data have been processed to build a new database that overlays both SAR-derived and AIS-derived vessel locations.

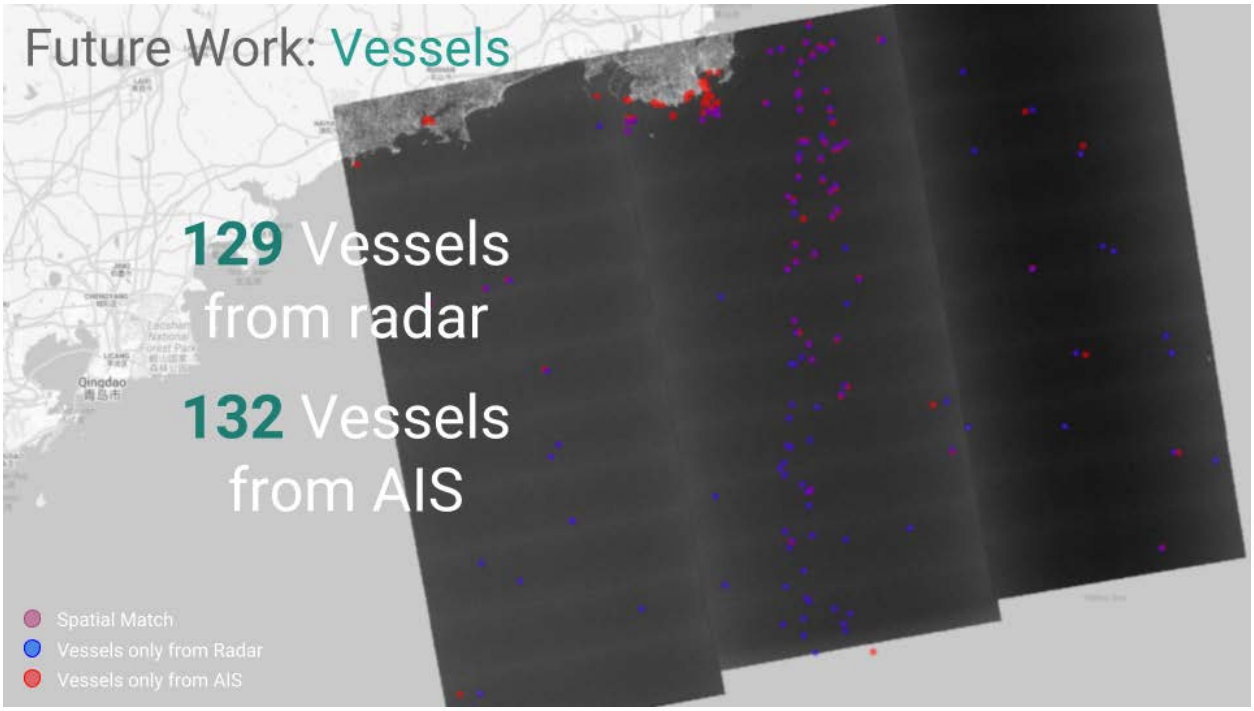

Figure 17. Adding AIS-derived vessel locations on top of SAR-derived vessel locations.

Limitations

Much of the future work and the establishment of the offshore infrastructure identification model are dependent on a reliable ground truth dataset. The degree to which SkyTruth was required to pre-process and post-process the BOEM ground truth dataset was moderately troublesome, so increased research for future validations of non-BOEM ground truth datasets might be worth the extra initial effort to reduce these potential uncertainties.

The largest portion of the analysis was used to establish a robust model for offshore infrastructure identification in the Gulf of Mexico, and even though initial tests of additional ad hoc analyses for wind turbines and vessel identifications proved promising, these indications are just that, an early indication and an extremely limited sample size given the clients intentions to deploy these globally. As such, these portions of the analysis should only be interpreted as early indicators, not robust methods.

CONCLUSIONS

The impetus of this study was initiated by a GFW need for a global offshore infrastructure dataset. In the early stages of model development, extracting offshore platform locations for a remarkably small subset region containing only two dozen platforms proved more than challenging, so the final model performance that produced a 91.91% overall accuracy for all the Gulf of Mexico validated to 2,135 ground truth points was considered a significant success by both clients. Furthermore, additional gains were achieved by testing the model's flexibility for adaptability and implementing cloud-native geoprocessing workflows for scalability.

The concluding recommendations for the clients all surround model validation. A strong foundation was established, however, most were geographically constrained. Conducting similar robust validations for both offshore infrastructure outside the Gulf of Mexico and vessel identifications will provide the clients with heightened confidence to deploy the model globally. Additionally, reaching these future milestones will enable broader scientific inquiry, especially with respect to spatiotemporal trends characterizing the legacy of anthropogenic structures throughout the oceans, and characterizing an appended landscape of fishing activity currently undetected (Liu et al., 2016).

REFERENCES

BOEM (Bureau of Ocean Energy Management), *Geographic Mapping Data in Digital Format*, Jan. 2018, https://www.data.boem.gov/Main/Mapping.aspx, Date Accessed: January 1, 2018.

Cervera, M. A., and A. Ginesi. "On the Performance Analysis of a Satellite-Based AIS System." *2008 10th International Workshop on Signal Processing for Space Communications*, 2008, pp. 1–8. *IEEE Xplore*, doi:10.1109/SPSC.2008.4686715.

Clazzer, F., et al. "On the Impact of Coverage Range on AIS Message Reception at Flying Platforms." *2014 7th Advanced Satellite Multimedia Systems Conference and the 13th Signal Processing for Space Communications Workshop (ASMS/SPSC)*, 2014, pp. 128–35. *IEEE Xplore*, doi:10.1109/ASMS-SPSC.2014.6934534.

Elvidge, Christopher D., et al. "Automatic Boat Identification System for VIIRS Low Light Imaging Data." *Remote Sensing*, vol. 7, no. 3, Mar. 2015, pp. 3020–36. *www.mdpi.com*, doi:10.3390/rs70303020.

FAO (Food and Agriculture Organization of the United Nations), "The State of World Fisheries and Aquaculture (SOFIA)", Rome: Food and Agriculture Organization, 2016.

Global Fishing Watch. "Bright Lights Reveal the "Dark" Fleet". Jan. 2018, http://globalfishingwatch.org/press-release/bright-lights-reveal-dark-fleet/, Date Accessed: March 1, 2018.

Gorelick, N., Hancher, M., Dixon, M., Ilyushchenko, S., Thau, D., & Moore, R. (2017). Google Earth Engine: Planetary-scale geospatial analysis for everyone. Remote Sensing of Environment.

Greidanus, Harm, et al. "The SUMO Ship Detector Algorithm for Satellite Radar Images." *Remote Sensing*, vol. 9, no. 3, Mar. 2017, p. 246. *www.mdpi.com*, doi:10.3390/rs9030246.

Haralick, R. M., et al. “Image Analysis Using Mathematical Morphology.” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-9, no. 4, July 1987, pp. 532–50. *IEEE Xplore*, doi:10.1109/TPAMI.1987.4767941.

Kocak, D. M., and P. Browning. “Real-Time AIS Tracking from Space Expands Opportunities for Global Ocean Observing and Maritime Domain Awareness.” *OCEANS 2015 - MTS/IEEE Washington*, 2015, pp. 1–7. *IEEE Xplore*, doi:10.23919/OCEANS.2015.7404635.

Kroodsma, David A., et al. “Tracking the Global Footprint of Fisheries.” *Science*, vol. 359, no. 6378, Feb. 2018, pp. 904–08. *science.sciencemag.org*, doi:10.1126/science.aao5646.

Liu, Yang, et al. “Detection of Squid and Pacific Saury Fishing Vessels around Japan Using VIIRS Day/Night Band Image.” *Proceedings of the Asia-Pacific Advanced Network*, vol. 39, no. 0, June 2015, pp. 28–39. *journals.sfu.ca*, doi:10.7125/APAN.39.3.

Liu, Yongxue, et al. “Automatic Extraction of Offshore Platforms Using Time-Series Landsat-8 Operational Land Imager Data.” *Remote Sensing of Environment*, vol. 175, Mar. 2016, pp. 73–91. *ScienceDirect*, doi:10.1016/j.rse.2015.12.047.

Liu, Yongxue, et al. “Satellite Data Lift the Veil on Offshore Platforms in the South China Sea.” *Scientific Reports*, vol. 6, Sept. 2016, p. 33623. *www.nature.com*, doi:10.1038/srep33623.

Marino, Armando, et al. “Ship Detection with Spectral Analysis of Synthetic Aperture Radar: A Comparison of New and Well-Known Algorithms.” *Remote Sensing*, vol. 7, no. 5, Apr. 2015, pp. 5416–39. *www.mdpi.com*, doi:10.3390/rs70505416.

Marino, A., et al. “Offshore Metallic Platforms Observation Using Dual-Polarimetric TS-X/TD-X Satellite Imagery: A Case Study in the Gulf of Mexico.” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 10, no. 10, Oct. 2017, pp. 4376–86. *IEEE Xplore*, doi:10.1109/JSTARS.2017.2718584.

Mladenova, I. E., et al. “Incidence Angle Normalization of Radar Backscatter Data.” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 3, Mar. 2013, pp. 1791–804. *IEEE Xplore*, doi:10.1109/TGRS.2012.2205264.

O’Grady, Damien, et al. “Relationship of Local Incidence Angle with Satellite Radar Backscatter for Different Surface Conditions.” *International Journal of Applied Earth Observation and Geoinformation*, vol. 24, Oct. 2013, pp. 42–53. *ScienceDirect*, doi:10.1016/j.jag.2013.02.005.

Pal, Nikhil R., and Sankar K. Pal. “A Review on Image Segmentation Techniques.” *Pattern Recognition*, vol. 26, no. 9, Sept. 1993, pp. 1277–94. *ScienceDirect*, doi:10.1016/0031-3203(93)90135-J.

Stasolla, M., et al. “A Comparative Study of Operational Vessel Detectors for Maritime Surveillance Using Satellite-Borne Synthetic Aperture Radar.” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 6, June 2016, pp. 2687–701. *IEEE Xplore*, doi:10.1109/JSTARS.2016.2551730.

Tetreault, B. J. “Use of the Automatic Identification System (AIS) for Maritime Domain Awareness (MDA).” *Proceedings of OCEANS 2005 MTS/IEEE*, 2005, pp. 1590-1594 Vol. 2. *IEEE Xplore*, doi:10.1109/OCEANS.2005.1639983.

Topouzelis, Konstantinos, et al. “Incidence Angle Normalization of Wide Swath SAR Data for Oceanographic Applications.” *Open Geosciences*, vol. 8, no. 1, 2016, pp. 450–464. *DeGruyter*, doi:10.1515/geo-2016-0029.

Yang, Chan-Su, et al. “Design of Integrated Ship Monitoring System Using SAR, RADAR, and AIS.” *Ocean Sensing and Monitoring V*, vol. 8724, International Society for Optics and Photonics, 2013, p. 872411. *www-spiedigitallibrary-org.proxy.lib.duke.edu*, doi:10.1117/12.2018017.

APPENDIX A: Google Earth Engine Scripts

GEE Script 1:

```
/*
This script outputs the Sentinel-1 SAR Median Composite, temporary Mean
Filter
image, and the Differenced Image.

All three of the aforementioned outputs are used in the offshore
infrastructure
identification model.
*/

var geometry = /* color: #98ff00 */ee.Geometry.Polygon(
        [[[-100.44041416179334, 22.869654655087317],
          [-83.65878725545628, 25.076806340276033],
          [-83.87881909028494, 32.35049335523377],
          [-100.8773367110669, 30.675813748689514]]]);

var aoi = /* color: #d63000 */ee.Geometry.Polygon(
        [[[-93.66943359375, 29.92637417863576],
          [-96.800537109375, 28.178559849396976],
          [-97.36083984375, 27.254629577800063],
          [-97.261962890625, 25.997549919572112],
          [-91.56005859375, 25.94816628853973],
          [-87.374267578125, 27.44004046509707],
          [-87.64892578125, 30.263811840754933],
          [-88.8134765625, 30.18549602210944],
          [-88.8072016843314, 30.02977826989111],
          [-88.81186884824922, 29.909614101481804],
          [-88.87664794921875, 29.807284450222507],
          [-88.86927760947253, 29.689883587631527],
          [-88.94835656755407, 29.602285664639588],
          [-89.25831771324192, 29.437607127773777],
          [-89.4287109375, 29.143765595138238],
          [-89.65530395507812, 29.284003336047167],
          [-89.879150390625, 29.286398892934763],
          [-90.36529541015625, 29.050167257528237],
          [-90.58364868164062, 29.074174942091428],
          [-90.86517333984375, 29.023752349427596],
          [-90.9228515625, 29.120973840852503],
          [-91.04782104492188, 29.15815778336055],
          [-91.2359619140625, 29.21211052272621],
          [-91.34857177734375, 29.279212053761192],
          [-91.43508911132812, 29.321128153792174],
          [-91.77978515625, 29.431225474131686],
          [-92.00088500976562, 29.52089051902536],
          [-92.14096069335938, 29.562707047643393]]]);
Map.addLayer(aoi, {}, 'gom aoi')
//////////////////////------------ FUNCTIONS -------------
//////////////////////

// Angle correction function
function sarAngleCorrection(image) {
```

```
  return
image.select('V.|H.').subtract(image.select('angle').multiply(Math.PI/180.0).
pow(2).cos().log10().multiply(10.0));
}

// Backtransform to Digital Number (dn)
// note that multiplying it 10k is purely for human-readability but also for
casting to interger
function toNatural (image){
  return (ee.Image(10).pow(image.divide(10))).multiply(10000);
}

// function maskLowEntropy(image) {
//   var bad =
image.select(0).multiply(10000).toInt().entropy(ee.Kernel.circle(5)).lt(3.2)
//   return image.updateMask(image.mask().multiply(bad.focal_max(5).not()))
// }

// function maskEdge(image) {
//   var mask = image.select(0).unitScale(-25,
5).multiply(255).toByte().connectedComponents(ee.Kernel.rectangle(1,1), 100);
//   return image.updateMask(mask.select(0));
// }

// function clipEdge(image){
//   var geometry = image.geometry()
//   var edgeClippedImage = image.clip(image.geometry().buffer(-3000))
//   return edgeClippedImage
// }

//////////////////////------------ COMPOSITE -------------
//////////////////////
var inputCollection = ee.ImageCollection('COPERNICUS/S1_GRD')
  .filterDate('2017-01-01', '2017-12-31')
  .filterBounds(aoi)
  .filterMetadata('instrumentMode', 'equals', 'IW')
  .filter(ee.Filter.listContains('transmitterReceiverPolarisation', 'VV'))
  .filter(ee.Filter.listContains('transmitterReceiverPolarisation', 'VH'))
  .filter(ee.Filter.eq('resolution','H'))
  .map(sarAngleCorrection)
  .map(toNatural)
  .median() // note the median reducer is applied after angle corr. & DN
functions applied
  .int() // to save space
  .select('VH')
  .clip(geometry)

print(inputCollection)

//////////////////////------------ IMAGE COUNTS -------------
//////////////////////

var vhVizParam = {"opacity":1,"bands":["VH"],"min": 0 ,"max": 200,"gamma":1}
Map.addLayer(inputCollection, vhVizParam, 'GOM Median Composite 2017-VH')

// var countImg = inputCollection
//     .select("VH")
```

```
//     .reduce(ee.Reducer.count());

//
Map.addLayer(countImg,{"opacity":0.6,"bands":["VH_count"],"min":1,"max":4,"pa
lette":["ffc762","ff5454","ff0584"]},'Image Count Scene', true, 0.9);
// Map.setOptions('SATELLITE')
// print(inputCollection)

// var i = ee.Image("users/bawong/gfw-
skytruth/gom/gomMedianSarDnComposite_2017");
// Map.addLayer(i, vhVizParam)

////////////////////////////////////////////////////////////////////////////
/////
//////////////////////------------ EXPORTS -------------
//////////////////////
////////////////////////////////////////////////////////////////////////////
/////


//////////////////////------------ COMPOSITE EXPORT -------------
//////////////////////

var inputCollectionGeometry = inputCollection.geometry()
Export.image.toAsset({
  image: inputCollection,
  description: 'gom_vh_composite_2017_int',
  region: inputCollectionGeometry,
  scale: 10,
  maxPixels: 1e13})

//////////////////////------------ FOCAL MEAN EXPORT -------------
//////////////////////

var vh_mean = inputCollection
  .select('VH')
  .focal_mean({radius: 250, kernelType: 'circle', units: 'meters'})
  .int()

Export.image.toAsset({
  image: vh_mean,
  description: 'gom_vh_mean_2017_int',
  region: inputCollectionGeometry,
  scale: 10,
  maxPixels: 1e13})

//////////////////////------------ DIFFERENCE IMAGE EXPORT -------------
//////////////////////

var vh_diff = inputCollection.select('VH').subtract(vh_mean).int()

Export.image.toAsset({
  image: vh_diff,
  description: 'gom_vh_diff_2017_int',
  region: inputCollectionGeometry,
  scale: 10,
  maxPixels: 1e13})
```

GEE Script 2 (Global Thresholding Model):

```
/*
This script loops over a Feature Collection overlaying the study site (Gulf
of Mexico) to
identify offshore infrastructure locations.

Inputs are from previous script above. Output is a GeoJSON of the
predictions.

This is the global thresholding version.
*/

var fc = /* color: #d63000 */ee.FeatureCollection(
        [ee.Feature(
            ee.Geometry.Polygon(
                [[[-89.5, 30.5],
                  [-89.5, 29.5],
                  [-87.5, 29.5],
                  [-87.5, 30.5]]]),
            {
              "system:index": "0"
            }),
        ee.Feature(
            ee.Geometry.Polygon(
                [[[-89.5, 29.5],
                  [-89.5, 28.5],
                  [-87.5, 28.5],
                  [-87.5, 29.5]]]),
            {
              "system:index": "1"
            }),
        ee.Feature(
            ee.Geometry.Polygon(
                [[[-89.5, 28.5],
                  [-89.5, 27.5],
                  [-87.5, 27.5],
                  [-87.5, 28.5]]]),
            {
              "system:index": "2"
            }),
        ee.Feature(
            ee.Geometry.Polygon(
                [[[-89.5, 27.5],
                  [-89.5, 26.5],
                  [-87.5, 26.5],
                  [-87.5, 27.5]]]),
            {
              "system:index": "3"
            }),
        ee.Feature(
            ee.Geometry.Polygon(
                [[[-91.5, 29.5],
                  [-91.5, 28.5],
```

```
          [-89.5, 28.5],
          [-89.5, 29.5]]]),
    {
      "system:index": "4"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-91.5, 28.5],
          [-91.5, 27.5],
          [-89.5, 27.5],
          [-89.5, 28.5]]]),
    {
      "system:index": "5"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-91.5, 27.5],
          [-91.5, 26.5],
          [-89.5, 26.5],
          [-89.5, 27.5]]]),
    {
      "system:index": "6"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-91.5, 26.5],
          [-91.5, 25.5],
          [-89.5, 25.5],
          [-89.5, 26.5]]]),
    {
      "system:index": "7"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-93.5, 30.5],
          [-93.5, 29.5],
          [-91.5, 29.5],
          [-91.5, 30.5]]]),
    {
      "system:index": "8"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-93.5, 29.5],
          [-93.5, 28.5],
          [-91.5, 28.5],
          [-91.5, 29.5]]]),
    {
      "system:index": "9"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-93.5, 28.5],
          [-93.5, 27.5],
          [-91.5, 27.5],
          [-91.5, 28.5]]]),
    {
```

```
      "system:index": "10"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-93.5, 27.5],
          [-93.5, 26.5],
          [-91.5, 26.5],
          [-91.5, 27.5]]]),
    {
      "system:index": "11"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-93.5, 26.5],
          [-93.5, 25.5],
          [-91.5, 25.5],
          [-91.5, 26.5]]]),
    {
      "system:index": "12"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-95.5, 30.5],
          [-95.5, 29.5],
          [-93.5, 29.5],
          [-93.5, 30.5]]]),
    {
      "system:index": "13"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-95.5, 29.5],
          [-95.5, 28.5],
          [-93.5, 28.5],
          [-93.5, 29.5]]]),
    {
      "system:index": "14"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-95.5, 28.5],
          [-95.5, 27.5],
          [-93.5, 27.5],
          [-93.5, 28.5]]]),
    {
      "system:index": "15"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-95.5, 27.5],
          [-95.5, 26.5],
          [-93.5, 26.5],
          [-93.5, 27.5]]]),
    {
      "system:index": "16"
    }),
ee.Feature(
```

```
            ee.Geometry.Polygon(
                [[[-95.5, 26.5],
                  [-95.5, 25.5],
                  [-93.5, 25.5],
                  [-93.5, 26.5]]]),
            {
              "system:index": "17"
            }),
        ee.Feature(
            ee.Geometry.Polygon(
                [[[-97.5, 29.5],
                  [-97.5, 28.5],
                  [-95.5, 28.5],
                  [-95.5, 29.5]]]),
            {
              "system:index": "18"
            }),
        ee.Feature(
            ee.Geometry.Polygon(
                [[[-97.5, 28.5],
                  [-97.5, 27.5],
                  [-95.5, 27.5],
                  [-95.5, 28.5]]]),
            {
              "system:index": "19"
            }),
        ee.Feature(
            ee.Geometry.Polygon(
                [[[-97.5, 27.5],
                  [-97.5, 26.5],
                  [-95.5, 26.5],
                  [-95.5, 27.5]]]),
            {
              "system:index": "20"
            }),
        ee.Feature(
            ee.Geometry.Polygon(
                [[[-97.5, 26.5],
                  [-97.5, 25.5],
                  [-95.5, 25.5],
                  [-95.5, 26.5]]]),
            {
              "system:index": "21"
            })]);

var aoi = ee.FeatureCollection("users/brianwong/gom_aoi_boem_czma_2km")

function findStructures (fc) {
  var s1_comp =
ee.Image("users/brianwong/gom_vh_composite_2017").clip(fc).clip(aoi)
  var vh_mean = s1_comp
    .select('VH')
    .focal_mean({radius: 250, kernelType: 'circle', units: 'meters'})
  var vh_diff = ee.Image('users/brianwong/gom_vh_diff_2017')
  var threshold = 50
```

```
  var candidatePixels =
vh_diff.select('VH').gte(threshold).addBands(vh_diff.gte(threshold).rename('V
H_duplicate'))
  var candidatePixelsCleaned = candidatePixels
    .focal_min({radius: 05, units: 'meters'})
    .focal_max({radius: 20, units: 'meters'})

    // Reduce each cluster of candidate pixels to its centroid:
  var centroids = candidatePixelsCleaned.reduceToVectors(
    {reducer:ee.Reducer.median(),
    geometry:fc.geometry(),
    scale:10,
    geometryType:'centroid',
    eightConnected:true,
    bestEffort:false,
    maxPixels:1e15})
  return centroids
}

var exports_tables = fc.map(findStructures).flatten()

Export.table.toDrive({
  collection: exports_tables,
  description: 'gom_infra_fmin05_fmax20_vh50',
  fileFormat: 'GeoJSON'})
```

GEE Script 3 (Dynamic Thresholding Model):

```
/*
This script loops over a Feature Collection overlaying the study site (Gulf
of Mexico) to
identify offshore infrastructure locations.

Inputs are from previous script above. Output is a GeoJSON of the
predictions.

This is the dynamic thresholding version.
*/

var fc = /* color: #d63000 */ee.FeatureCollection(
        [ee.Feature(
            ee.Geometry.Polygon(
                [[[-89.5, 30.5],
                  [-89.5, 29.5],
                  [-87.5, 29.5],
                  [-87.5, 30.5]]]),
            {
              "system:index": "0"
            }),
        ee.Feature(
            ee.Geometry.Polygon(
                [[[-89.5, 29.5],
                  [-89.5, 28.5],
                  [-87.5, 28.5],
                  [-87.5, 29.5]]]),
```

```
    {
      "system:index": "1"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-89.5, 28.5],
          [-89.5, 27.5],
          [-87.5, 27.5],
          [-87.5, 28.5]]]),
    {
      "system:index": "2"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-89.5, 27.5],
          [-89.5, 26.5],
          [-87.5, 26.5],
          [-87.5, 27.5]]]),
    {
      "system:index": "3"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-91.5, 29.5],
          [-91.5, 28.5],
          [-89.5, 28.5],
          [-89.5, 29.5]]]),
    {
      "system:index": "4"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-91.5, 28.5],
          [-91.5, 27.5],
          [-89.5, 27.5],
          [-89.5, 28.5]]]),
    {
      "system:index": "5"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-91.5, 27.5],
          [-91.5, 26.5],
          [-89.5, 26.5],
          [-89.5, 27.5]]]),
    {
      "system:index": "6"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-91.5, 26.5],
          [-91.5, 25.5],
          [-89.5, 25.5],
          [-89.5, 26.5]]]),
    {
      "system:index": "7"
    }),
```

```
ee.Feature(
    ee.Geometry.Polygon(
        [[[-93.5, 30.5],
          [-93.5, 29.5],
          [-91.5, 29.5],
          [-91.5, 30.5]]]),
    {
      "system:index": "8"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-93.5, 29.5],
          [-93.5, 28.5],
          [-91.5, 28.5],
          [-91.5, 29.5]]]),
    {
      "system:index": "9"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-93.5, 28.5],
          [-93.5, 27.5],
          [-91.5, 27.5],
          [-91.5, 28.5]]]),
    {
      "system:index": "10"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-93.5, 27.5],
          [-93.5, 26.5],
          [-91.5, 26.5],
          [-91.5, 27.5]]]),
    {
      "system:index": "11"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-93.5, 26.5],
          [-93.5, 25.5],
          [-91.5, 25.5],
          [-91.5, 26.5]]]),
    {
      "system:index": "12"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-95.5, 30.5],
          [-95.5, 29.5],
          [-93.5, 29.5],
          [-93.5, 30.5]]]),
    {
      "system:index": "13"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-95.5, 29.5],
```

```
          [-95.5, 28.5],
          [-93.5, 28.5],
          [-93.5, 29.5]]]),
    {
      "system:index": "14"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-95.5, 28.5],
          [-95.5, 27.5],
          [-93.5, 27.5],
          [-93.5, 28.5]]]),
    {
      "system:index": "15"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-95.5, 27.5],
          [-95.5, 26.5],
          [-93.5, 26.5],
          [-93.5, 27.5]]]),
    {
      "system:index": "16"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-95.5, 26.5],
          [-95.5, 25.5],
          [-93.5, 25.5],
          [-93.5, 26.5]]]),
    {
      "system:index": "17"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-97.5, 29.5],
          [-97.5, 28.5],
          [-95.5, 28.5],
          [-95.5, 29.5]]]),
    {
      "system:index": "18"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-97.5, 28.5],
          [-97.5, 27.5],
          [-95.5, 27.5],
          [-95.5, 28.5]]]),
    {
      "system:index": "19"
    }),
ee.Feature(
    ee.Geometry.Polygon(
        [[[-97.5, 27.5],
          [-97.5, 26.5],
          [-95.5, 26.5],
          [-95.5, 27.5]]]),
```

```
            {
              "system:index": "20"
            }),
        ee.Feature(
            ee.Geometry.Polygon(
                [[[-97.5, 26.5],
                  [-97.5, 25.5],
                  [-95.5, 25.5],
                  [-95.5, 26.5]]]),
            {
              "system:index": "21"
            })]);

var aoi = ee.FeatureCollection("users/brianwong/gom_aoi_boem_czma")

function findStructures (fc) {
  // var s1_comp =
ee.Image("users/brianwong/gom_vh_composite_2017").clip(fc).clip(aoi)
  // var vh_mean = s1_comp
  //   .select('VH')
  //   .focal_mean({radius: 250, kernelType: 'circle', units: 'meters'})
  var vh_mean =
ee.Image("users/brianwong/gom_vh_mean_2017_int").clip(fc).clip(aoi) // need
to re-to toInt()
  var vh_diff =
ee.Image('users/brianwong/gom_vh_diff_2017').clip(fc).clip(aoi) //aka s_1comp
minus vh_mean
  //var threshold = 50
  var threshold = vh_mean.multiply(ee.Image(2.5).clip(fc)).clip(aoi)
  var candidatePixels =
vh_diff.select('VH').gte(threshold).addBands(vh_diff.gte(threshold).rename('V
H_duplicate'))
  var candidatePixelsCleaned = candidatePixels
    .focal_min({radius: 10, units: 'meters'})
    .focal_max({radius: 20, units: 'meters'})


    // Reduce each cluster of candidate pixels to its centroid:
  var centroids = candidatePixelsCleaned.reduceToVectors(
    {reducer:ee.Reducer.median(),
    geometry:fc.geometry(),
    scale:10,
    geometryType:'centroid',
    eightConnected:true,
    bestEffort:false,
    maxPixels:1e15})
  return centroids
}

var exports_tables = fc.map(findStructures).flatten()

Export.table.toDrive({
  collection: exports_tables,
  description: 'gom_infra_fmin10_fmax20_25xmed',
  fileFormat: 'GeoJSON'})
```

APPENDIX B: Vessel Identification Scripts

```
/*
This script is a preliminary method to identify vessels from Sentinel-1's C-
band SAR
imagery. Generally speaking, the key observation was that data transformed
back to
the unitless digital number (dn) provided an increased ability to
discriminate vessel
pixels from open water.

The current threshold is based off of a quick survey of pixel distributions,
specifically
trying to identify if and where the distribution of 1) vessel pixels and 2)
open water
pixels are different from each other.

Author: Brian A. Wong
2018-02-15
Duke University, Nicholas School of the Environment
*/

// GEE asset imports:
var waterVector_1km =
ee.FeatureCollection('users/christian/LandWaterVectors_derivedFrom_LandWater_
raster_20171006/ProcessedFinal_WaterVector_1000m');

// This is actually the result of the script, however, rendering it in the
GEE JS playground was
// time intensive, so the results are exported then brought back in for
visualization.
var vessel_centroids_vh_thresh_600 = ee.FeatureCollection('users/bawong/gfw-
skytruth/centroid_export_vh_max_thresh_600')

// Test scene from Indoesian waters on 01-17-2018:
var image =
ee.Image('COPERNICUS/S1_GRD/S1A_IW_GRDH_1SDV_20180117T220823_20180117T220848_
020200_02276D_110E');
// try these too:
//var image =
ee.Image('COPERNICUS/S1_GRD/S1A_IW_GRDH_1SDV_20180107T110642_20180107T110707_
020047_02228D_B1C2');
//var image =
ee.Image('COPERNICUS/S1_GRD/S1A_IW_GRDH_1SDV_20180103T222432_20180103T222457_
019996_0220FE_BD14');
//var image =
ee.Image('COPERNICUS/S1_GRD/S1A_IW_GRDH_1SDV_20180121T105015_20180121T105040_
020251_022919_92CC');
//var image =
ee.Image('COPERNICUS/S1_GRD/S1A_IW_GRDH_1SDV_20180117T220848_20180117T220913_
020200_02276D_7946');


// Clipping S-1 edge noise off:
var geometry = image.geometry()
```

```
var edgeClippedImage = image.clip(image.geometry().buffer(-5000))

// S-1 Angle correction:
var angleCorrected =
edgeClippedImage.select('V.|H.').subtract(edgeClippedImage.select('angle').mu
ltiply(Math.PI/180.0).pow(2).cos().log10().multiply(10.0))

// Transform image from decibels to digital number (dn):
var dn =
ee.Image(10.0).pow(angleCorrected.divide(10.0)).multiply(10000).clip(waterVec
tor_1km)
print(dn)

var dn_vh_mean = dn.focal_mean({radius: 250, kernelType: 'circle', units:
'meters'})
//Map.addLayer(dn_vh_mean, {"opacity":1,"bands":["VH"],"max":600,"gamma":1},
'dn mean', true, 0.1)

var dn_vh_diff = dn.subtract(dn_vh_mean)

// I was picking some vessel-attached fishing gear as seperate vessels so to
focal-max was used to
// to essentially connect the aforementioned gear back to its vessel.
var dnFocalMax = dn_vh_diff.focal_max(40, 'circle', 'meters')

// Threshold to isolate candidate pixels based on a quick survey of pixel
distributions.
var threshold = 600
var candidatePixels =
dnFocalMax.select('VH').gte(threshold).addBands(dnFocalMax.gte(threshold))

//Reduce each cluster of candidate pixels to its centroid:
var centroids = candidatePixels.reduceToVectors(
  {reducer:ee.Reducer.median(),
  geometry:geometry,
  scale:10,
  geometryType:'centroid',
  eightConnected:false,
  bestEffort:false,
  maxPixels:1e15});

// Export results and reload as asset - too heavy to render on fly:
// Export.table.toDrive({
//   collection: centroids,
//   description: 'centroid_export_vh_max_thresh_600',
//   fileFormat: 'GeoJSON'})
// Map.addLayer(centroids, {color:'red'}, 'centroids')
//print('Identified Vessel Count', centroids.size())

// Lastly, visualization of results:
Map.setOptions('SATELLITE')
Map.addLayer(image, {bands: 'VH', min:-30, max:0}, 'VH Raw Scene')
Map.addLayer(image, {bands: 'VV', min:-30, max:0}, 'VV Raw Scene')
Map.centerObject(image, 9)
Map.addLayer(dn, {"opacity":1,"bands":["VH"],"max":600,"gamma":1}, 'dn',
true, 0.1)
```

```
Map.addLayer(dn_vh_diff, {"opacity":1,"bands":["VH"],"max":600,"gamma":1},
'dn diff', true, 0.1)
Map.addLayer(dnFocalMax, {"opacity":1,"bands":["VH"],"max":600,"gamma":1},
'dnFocalMax', false)
Map.addLayer(vessel_centroids_vh_thresh_600, {color: 'red'}, 'vessels', true,
0.7)
print('Number of Vessels:', vessel_centroids_vh_thresh_600.size())


var newCentroids = ee.FeatureCollection("users/bawong/gfw-
skytruth/vessel_id/vessels_indo_vh_diff_fmax_thresh_600_20180223");
Map.addLayer(newCentroids, {color: 'blue'}, 'new vessels', true, 0.7)

print('newCentroids', newCentroids.size())
```

APPENDIX C: SkyTruth Manual Ground Truth Validation Process

Identification of Offshore Infrastructure - A Guide for Manually IDing Points

Data Creation:

As a key part of our work to identify offshore infrastructure located in the Gulf of Mexico, we need to create a dataset of verified infrastructure to be used for testing. This will allow us to conduct accuracy assessment, and will also help to inform the creation of more robust thresholds. To develop this set of verified infrastructure locations, we are relying on YOU! You are being asked to create a set of points which represent infrastructure which is identifiable in a Sentinel 1 composite image. If you have questions at any point, direct them to Christian.
The process for creating this dataset is unfortunately rather tedious. The process will be further explained by Christian, in person.

For each geometry you will classify all areas where BOEM Data is provided. For each classification of a geometry you will be required to use up to 5 point sets.

- **The 1st point set is to be used for areas in which a point present in the BOEM dataset is visible in both Sentinel 1 and Sentinel 2 imagery.**
- **The 2nd point set is to be used for areas in which a point present in the BOEM dataset is visible in Sentinel 1 but not in Sentinel 2 imagery.**
- **The 3rd point set is to be used for areas in which a point present in the BOEM dataset is visible in Sentinel 2 but not in Sentinel 1 imagery. (Hopefully this will not be used often)**
- **The 4th point set is to be used for areas in which a point present in the BOEM dataset is not visible in either Sentinel 1 or Sentinel 2 imagery.**
- **The 5th point set is to be used for areas you are unable to make a clear decision, Christian will review these.**

Geometry Area	Assignee	Output Link (when complete)

Geometry 1	Brady	https://code.earthengine.google.com/70bc21e0a28993bef70152e81cf8e79a
Geometry 2	Daniel	https://code.earthengine.google.com/cb5d1be22500db36ff5a4ac2dd45ff24
Geometry 3	Daniel	https://code.earthengine.google.com/03473a5a1689feaf0b4ccf66464684d7
Geometry 4	Daniel	https://code.earthengine.google.com/3bb0e251cfb4312743ef6ddaed055efa
Geometry 5	Brady	https://code.earthengine.google.com/a5aa2628e3d7461dc85007200d74e058
Geometry 6	Daniel / CT	https://code.earthengine.google.com/15324692e5f5fd73aa7fa125c051d74c
Geometry 7	Brady	https://code.earthengine.google.com/8e2923065000f45257b3199741f32a17
Geometry 8	CT	https://code.earthengine.google.com/466d33c970d693b08751114507d80d96
Geometry 9	CT	https://code.earthengine.google.com/a252396b5a83e1d01ee28238e7a96bf0
Geometry 10	CT	https://code.earthengine.google.com/7c6ea9b66d167c321bb542e8a5badec8

Follow the following steps:

1. **Enter your name next to the geometry you are going to classify in the chart above**
2. **Open the following script link: https://code.earthengine.google.com/d0f9b2dbdcc3cf574f13085900593882**
3. **Change the geo variable on lines 43 & 44 to match the geometry you are currently working on (the link defaults to geometry1, so if you are working on another geometry you will need to change it, *e.g.* geometry2, geometry3).**
4. **Add points where the presence of infrastructure can be verified, once finished with your geography, click "Get Link", and add it to the chart above.**

Data Processing & Finalization:

Data from the scripts linked above was exported as GeoJSON and converted to shapefiles using an ogr2ogr command. Shapefiles were initially multipart, but were converted to singlepart using QGIS' Multipart to Singleparts tool.

A "**class**" column was added to all shapefiles, values set between 1-5, to correspond to the following classes:

1. Point from BOEM data visible in both the Sentinel 1 and Sentinel 2 composite
2. Point from BOEM data visible only in the Sentinel 1 composite
3. Point from BOEM data visible only in the Sentinel 2 composite
4. Point from BOEM data visible in neither the Sentinel 1 nor the Sentinel 2 composite
5. Points for which a classification could not be confidently made (*e.g.* areas complicated by data errors, or "fuzziness" in the composite image) and also includes points which clearly corresponded to infrastructure that could not be confidently linked to any BOEM point.

Added a "**geometry**" column where each value corresponds to the geometry area that the points were grouped into during classification (value range: 1-10).

Added "**Longitude**" and "**Latitude**" columns for each classified point in QGIS.

Added a "**UID**" (unique ID) column so that each point has a unique ID number.

After all columns were added, the shapefile was converted to to GeoJSON and CSV.

For each classification of a geometry you will be required to use up to 5 point sets.

- **The 1st point set is to be used for areas in which a point present in the BOEM dataset is visible in both Sentinel 1 and Sentinel 2 imagery.**
- **The 2nd point set is to be used for areas in which a point present in the BOEM dataset is visible in Sentinel 1 but not in Sentinel 2 imagery.**
- **The 3rd point set is to be used for areas in which a point present in the BOEM dataset is visible in Sentinel 2 but not in Sentinel 1 imagery. (Hopefully this will not be used often)**
- **The 4th point set is to be used for areas in which a point present in the BOEM dataset is not visible in either Sentinel 1 or Sentinel 2 imagery.**
- **The 5th point set is to be used for areas you are unable to make a clear decision, Christian will review these.**

Geometry Area	Assignee	Output Link (when complete)
Geometry 1	Daniel - CT	https://code.earthengine.google.com/fb06250962fa8cc9e5287b42b67ad54a

Geometry 2	Daniel - CT	https://code.earthengine.google.com/f7044e06e71ee2cc9201a80feee5c62b
Geometry 3	Daniel - CT	https://code.earthengine.google.com/67a14ea38bc52c6fde05066c90c1459a
Geometry 4	Daniel - CT	https://code.earthengine.google.com/15281a143f94a88b3de7fd49379902b9
Geometry 5	Daniel - CT	https://code.earthengine.google.com/5543755fa771b3df60bf46797e637ced
Geometry 6	Brady - CT	https://code.earthengine.google.com/ed4ecbaa690448d98efc552b2bc8f07c
Geometry 7	Daniel - CT	https://code.earthengine.google.com/d388a1de1326a7cceef1b756b8e04e75
Geometry 8	Daniel - CT	https://code.earthengine.google.com/a6abe461cda1dff469edd8d3dd76440b
Geometry 9	Brady - CT	https://code.earthengine.google.com/b46ea32a146751b6bcbab19ef76642e3
Geometry 10	Brady - CT	https://code.earthengine.google.com/cba528e1336615695c400595fc6c797d

Follow the following steps:

1. **Enter your name next to the geometry you are going to classify in the chart above**
2. **Open the following script link: https://code.earthengine.google.com/40fa436310d1a5cb5feb3973c58430de**
3. **Change the geo variable on lines 51 & 52 to match the geometry you are currently working on (the link defaults to geometry1, so if you are working on another geometry you will need to change it, *e.g.* geometry2, geometry3).**
4. **Add points where the presence of infrastructure can be verified, once finished with your geography, click "Get Link", and add it to the chart above.**