Detecting windmill parks in the ocean on Sentinel satellite imagery

Aelbrecht Rudolf   Leclercq Frederic

Abstract

In this report we try to identify windmill parks in the ocean based on satellite imagery. A convolutional neural network was trained to detect individual windmills and boats. Land coverage maps and noise reduction techniques were applied in the preprocessing phase. Density based clustering algorithms (DBSCAN and OPTICS) were compared in order to optimize the intermediate result. The final result is a generated Esri shapefile that encircles all identified parks with a polygon.

1. Introduction

The Sentinel-1 mission is the European Radar Observatory for the Copernicus joint initiative of the European Commission (EC) and the European Space Agency (ESA). Copernicus is a European initiative for the implementation of information services dealing with environment and security. It is based on observation data received from Earth Observation satellites and ground-based information.

The Sentinel-1 mission includes C-band imaging operating in four exclusive imaging modes with different resolution (down to 5 m) and coverage (up to 400 km). It provides dual polarisation capability, very short revisit times and rapid product delivery. For each observation, precise measurements of spacecraft position and attitude are available.

Synthetic Aperture Radar (SAR) has the advantage of operating at wavelengths not impeded by cloud cover or a lack of illumination and can acquire data over a site during day or night time under all weather conditions. Sentinel-1, with its C-SAR instrument, can offer reliable, repeated wide area monitoring.

The Sentinel-1 products are freely downloadable in GeoTIFF format. GeoTIFF is a public domain metadata standard which allows georeferencing information to be embedded within a TIFF file. The potential additional information includes map projection, coordinate systems, ellipsoids, datums, and everything else necessary to establish the exact spatial reference for the file.

We aim to detect windmill parks in the ocean based on this satellite imagery.

2. Related work

The most relevant project we found to what we set out to do is a project developed in Duke University that is planning to map the energy infrastructure of the world using satellite imagery (duk, 2020a). They began with mapping all windmills in the USA and explained their approach (duk, 2020b) using a combination of machine learning and simulated imagery. They applied their model on colored satellite imagery.

Another related project is written on Medium (Moraite, 2019) which explained how it is relatively easy to identify ships using a CNN. This article has no sources for its data set due to link-rot, we assume it is related to another project using a different approach (Sagar, 2019). In both cases we can see that they used high resolution color satellite imagery.

3. Method

Compared to the existing works, we noticed that while looking at repositories of satellite imagery that cloudless imagery is quite rare and not feasible to parse on a global scale, since a lot of areas are missing. This is one of the reasons that we chose to use the single channel image data from Sentinel 1A, which is not affected by weather. Another disadvantage we had with our data set is that the resolution is quite low, and glare make it at times impossible

To lack of resolution and clarity made it hard to discern the shape of a windmill and thus to train a neural net to identify the distinct shape of a windmill. We therefor decided to focus on detecting parks instead of individual windmills.

3.1. Satellite Imagery Conversion

In order to be sure we could re-use our code we decided to re-project the satellite file to the World Geodetic System 1984 - WGS84 (EPSG:4326)

```
gdalwarp -t_srs EPSG:4326 in_f out_f
```

This also proved necessary due to the warped nature of the raw satellite imagery. By re-projecting the imagery to a fixed coordinate space we could easily switch between coordinates and pixel positions by using existing libraries.

3.2. Dataset Creation

As mentioned before the dataset from which we started is the Sentinel 1A satellite imagery. Since we did not have any labeled data we had to create one ourselves.

3.2.1. Manual Labeling

To create a manual labeled set we gathered a collection of coordinates including boats, land and windmills. This can be easily done in mapping software (e.g. QGIS). Once we had our set of coordinates we could start training it model. The model would extract tiles on the coordinates and use those images as training data. We quickly noticed that the amount of samples was not enough (around 150). So we looked at a way of increasing the number of samples.

3.2.2. Assisted Labeling

The issue with our first data set was that its size causing a lot of wrong identifications. This bad model actually gave us a great way of creating more samples. We expanded the number of samples we had by saving all right and wrongly identified windmills. To include data like oceans and beaches we would randomly sample at different locations. To ensure a high quality of our samples we also review each one to make sure they clearly identified the object they were representing and that they were centered.

To reduce loading times we would save each sample as an image instead of coordinate, making it possible to load our data set without the need of the original satellite image.

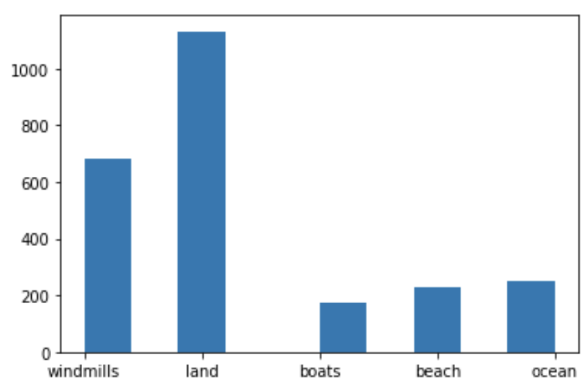

Figure 1. Statistics of final data set showing the number of examples of each category of classification class.

One might notice a large difference between the number of samples (fig. 1) per category. This was largely due to us wanting to initially train a model that could differentiate between land and non-land images. Due to difficulties of training such model samples of land would not be used in the final result. We tried to have an even amount of samples representing windmill and non-windmill tiles.

3.3. Noise reduction using OpenCV

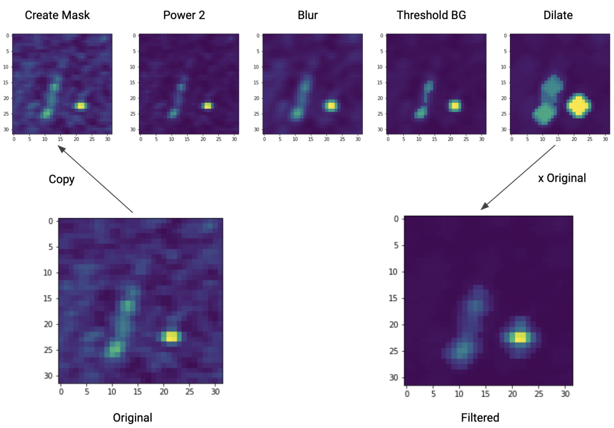

Figure 2. Step by step process of removing noise from a tile containing a single windmill example.

We noticed that depending on where the image the intensity of the image might be significantly different. To address this we would pre-process all images before feeding them to our neural network. By masking images in our data set we noticed a more consistent result in training. Without it was noticeable that when adding new samples our model would randomly train better or worse. By masking we would get more consistent results, however this was not empirically tested.

The masking itself was done using the OpenCV library. It consists of 4 steps (fig. 15):

- Take the power, increasing highlights and decreasing the background.
- Apply a $3x3$ blur kernel, removing small artifacts
- Remove the background by nullifying values below the avg value
- Dilate remaining features to include edges

We would then multiply the mask with our original image creating a nice mask around the windmill or boat tile. When using land or ocean imagery we noticed that this would amplify the amount of noise (fig. 11).

3.4. Labeling ocean objects using a CNN

3.4.1. Optimal Network

Before we started designing the topology of our neural network we first looked online to see if there are any example typologies that would be usable for our application. One article we found showed a working neural net to identify boats (Moraite, 2019). Whilst the article used color satellite imagery and we are used single channel, the network itself was a good basis to build on. The network however did not produce that good of results. We suspect that this is due to the very different shapes that windmills and boat can have depending on where and when the image was taken.

By tweaking the hyper-parameters however and increasing the number of filters to store more information in our network we were able to train a satisfactory model. We tweaked our parameters in a way that our model would be able to detect most windmills and boats but exclude any beaches or other artifacts present in the ocean. Each convolutions layer uses relu as its activation function. Our model had less artifacts when using a 2 class output, this is why we reduce our network to two final nodes, using sigmoid in the final step. The topology is as followed:

- Conv2D(32, 5x5)
- Pool2D(2x2)
- Conv2D(64, 5x5)
- Pool2D(2x2)
- Flatten()
- Dropout(0.5)
- Dense(2)

All steps contribute to a the stable training of our model. The introduction most notably fixed our issue of over-fitting that was initially noticeable with our small data set. For our result we settled on using the weights at 75 epochs, the moment that our loss would stop decreasing (fig. 3) for our test set.

Looking at our confusion matrix (fig. 4) we see that the model occasionally makes mistakes but these will be filtered away in the next step of our pipeline. One might notice that the amount of water-class examples is less than the object-class. This difference is due to the fact that examples of ocean tiles are mostly noise, adding more noisy ocean tiles

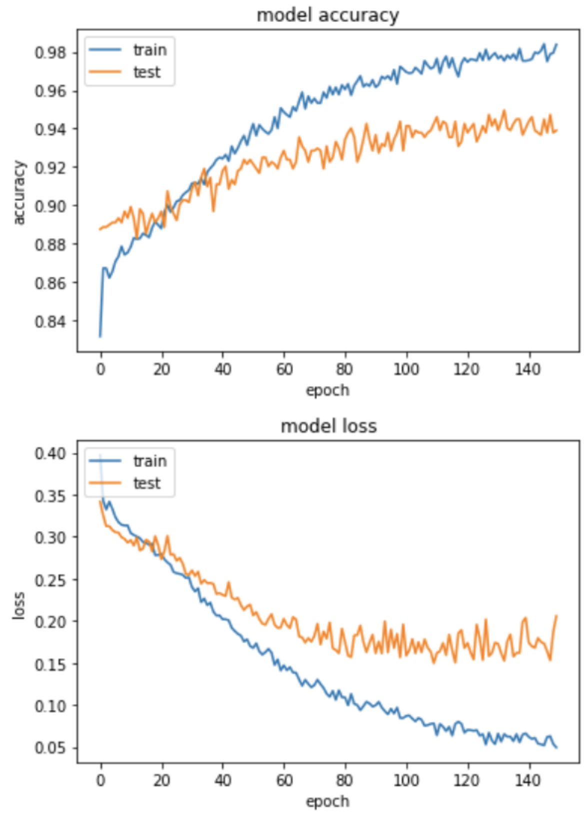

Figure 3. Training statistics of our final model.

did not improve our results, so we did not extract any more samples.

3.4.2. Alternative: Multi-class

As we mentioned we chose to train our model on two classes instead of the original 5 we designed our dataset for. For completeness sake we included the results of our initial network using 5 classes. The training results look similar, however we achieve a significantly lower accuracy and much higher loss in comparison to using two classes (fig. 6). Also note that land examples are excluded since land noise looks very similar to ocean noise and would decrease our accuracy even more.

Looking at the confusion matrix we can see that it is hard to differentiate beaches from boats and windmills. This is an issue that is hard to resolve since beaches have random patterns that sometimes produce artifacts approximating what one would identify as a boat or windmill.

What one can also notice is that boats and windmills can be differentiated quite well. However we noticed in later tests

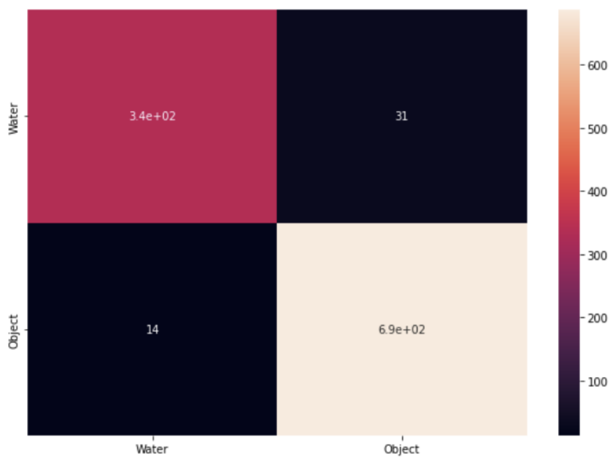

Figure 4. Confusion matrix of our final CNN.

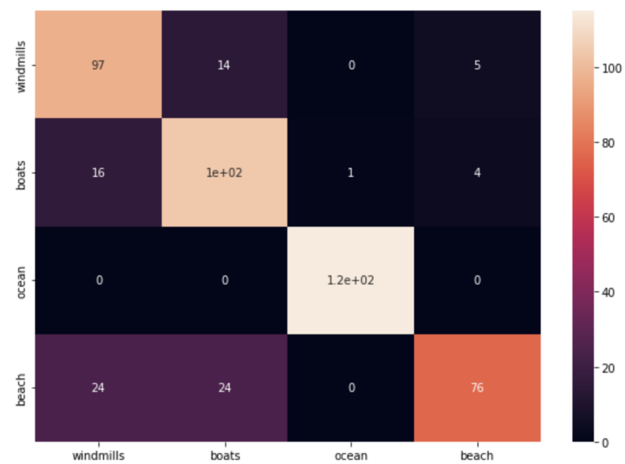

Figure 5. Confusion matrix using the same topology as our best performing network but modified for multi-classification.

that the shape of windmills varies greatly between satellite snapshots and that in reality windmills will have a boat-like shape in most cases.

3.4.3. Alternative: Different Topology

We experimented with a lot of different typologies. But in all cases we got worse results while training. One of the ideas was that our network should perform better when we gradually dense all our nodes to the final two nodes. The network itself is identical to the final model, however we replace the last step where we dense in to two nodes with a multiple dense layers (128, 64, 16 and 2). This gave us a better result on our training set but did not improve accuracy (fig. 7).

Another attempt was to build a network using a small amount of filters compared to our original network, 6 and 16 compared to 32 and 64. The model quickly perfected the identification of training samples but performed awful on our test set (fig. 8).

3.5. Detecting Ocean Tile

Since land has buildings which show up as rectangular objects in our satellite imagery, this was labeled in most cases as being a ship/windmill. We had to remove land images from our data set to be able to train our model correctly.

Our first approach was to train a different neural network that would differentiate between land and ocean. This however seemed like a bigger task than we initially though and gave up on the idea after not achieving any good results. We tried to follow a method described by the University of Chinese Academy of Sciences ((Hu et al., 2018)) but quickly realized this was beyond the scope of this project.

We started looking at different tools we could use to accurately define areas that are oceans. We found a project provided by Copernicus that has generated a land coverage map of every area on earth ((cop)). By downloading a segment a using the imagery as a bitmap for masking oceans we can quickly exclude any tile that can't be a windmill. We added some padding to the edges land bodies to make sure beach features get excluded but this is not a reliable way of excluding beaches as beaches can vary on time related factors like tides.

3.6. Clustering of ocean instances

3.6.1. Comparing clustering algorithms

As we expect to see multiple windmills per park we can exclude the outliers. Boats or even anomalies detected in the ocean can be removed by using the proper clustering technique. Ordering points to identify the clustering structure (OPTICS) and density-based spatial clustering of applications with noise (DBSCAN) are algorithms for finding density-based clusters in spatial data. We assume the number of clusters relatively low and the density of the clusters rather similar, therefore we tend to use the DBSCAN technique. However we tried the OPTICS clustering method as well with multiple values for the hyper-parameters.

It is clear that the DBSCAN clustering technique produced a result much more in the line we'd expect.

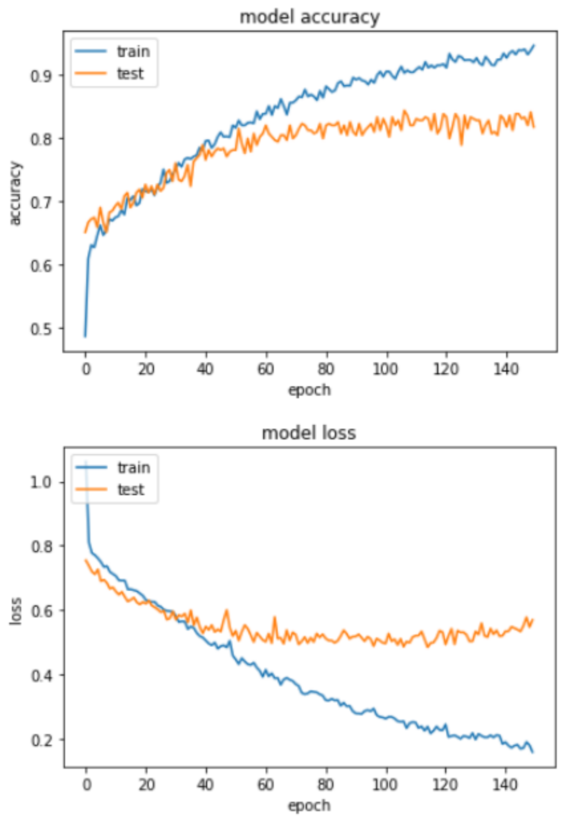

Figure 6. Training results when using multiple classes.

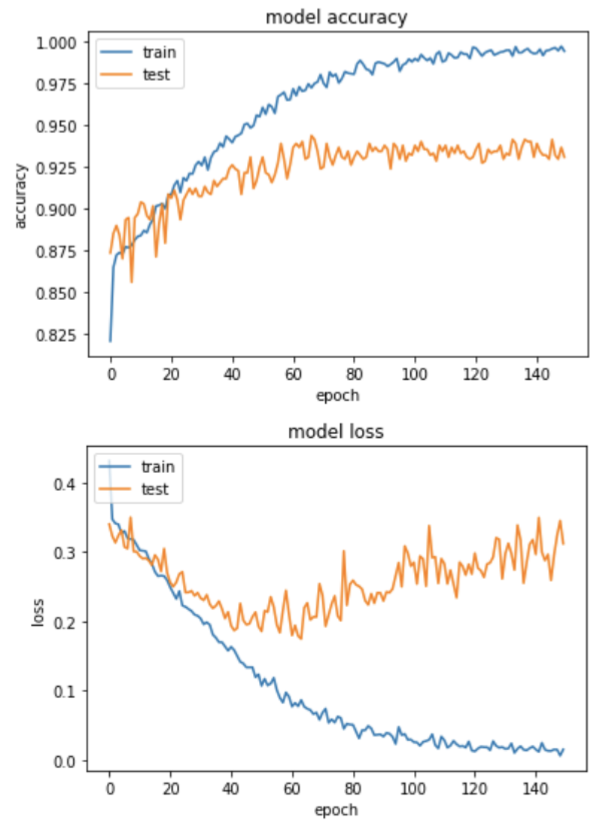

Figure 7. Alternative approach to CNN using multiple dense layers.

3.6.2. Creating a boundary shape

Create final shape as output of our pipeline

3.7. Pipeline Overview

Show a topology of our work. Link previous subsections together.

4. Results

4.1. Filtering clustered objects

Since clustering is an expensive operation and greatly depends on the distribution of points we first filter our set of result points to remove densely packed zones. We do this by saving all coordinates containing a windmill/boat in a map and checking for each point we add if it has any neighbors, if so it will skip adding this point. By using a map this algorithm has a logarithmic complexity making it feasible to parse large zones like the North Sea. An comparison of how windows don't overlap can be seen in Figure 12.

4.2. Final results

We parsed satellite images east of England, west of Denmark and south of China. Looking at the final results (fig. 13) we saw all windmill parks being marked as one. Clusters of boats were also labeled as parks, but these could be filtered when parsing the same area twice at different timestamps and retaining only parks that exist in both snapshots.

4.3. Exceptions

Our model did not perform equally on every case we tested (fig. 14). Coastal areas around Denmark that are periodically flooded are considered "ocean" even though beaches are clearly visible. These areas were still included in our parser which caused beaches to be labeled as windmill parks. One option of resolving this would be to expand borders or try to detect beaches separately from windmills/boats.

The second case in which our detection fails is when a large amount of boats are present. This is notable in parsing the

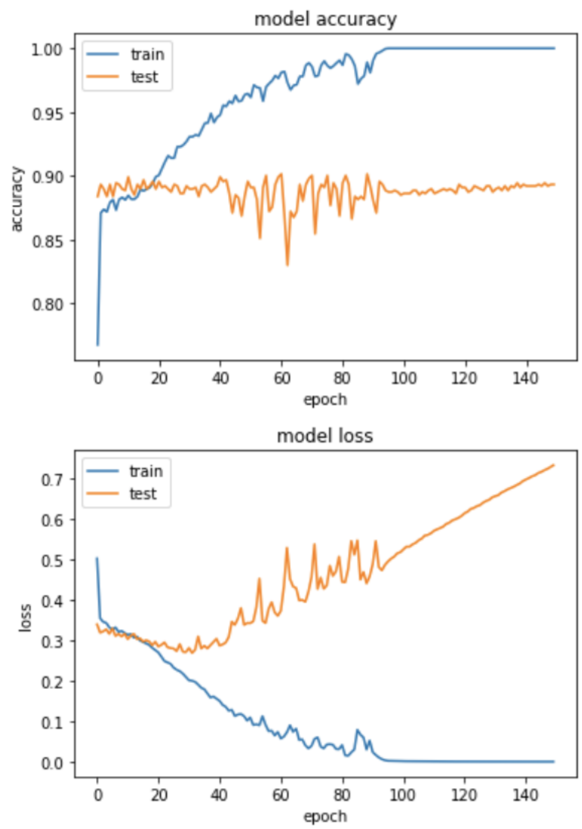

Figure 8. Alternative approach to CNN using less filters.

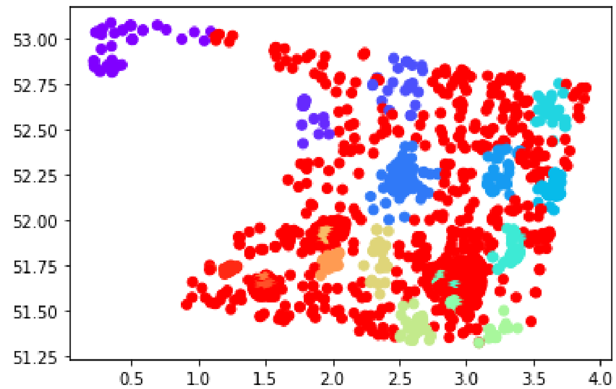

Figure 9. OPTICS Clustering result.

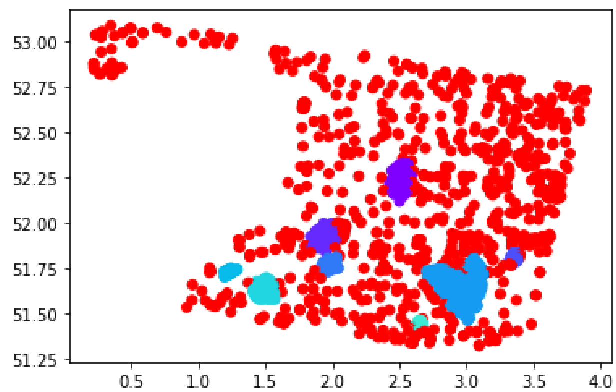

Figure 10. DBSCAN Clustering result.

Chinese coast (fig. 14). When a large amount of boats are clustered in an area the clustering will label this as a park. This could be avoided by filtering based on time but since harbors are always densely packed some a thorough way of filtering might be needed.

5. Conclusion

All in all, we are satisfied with this result. The combination of a neural network and a clustering technique resulted in a relatively good detection method. We can detect wind parks in the North Sea area, with only very few false positive polygons - which are dense clusters of boats. The model and clustering techniques performed extremely well in the Skaggerak area, in stark contrast with the result in Chinese waters, where our model performs very poorly. Possibly due to the different orientation of the windmills, but further research is needed to know more about this issue.

5.1. Future Work

In order to distinguish boats from windmills we might think of using the same pictures with another timestamp. Moving objects are most likely boats and we expect to have a much better final result, however the computation load will increase. Using Sentinel's API in order to download new images automatically in order to set up a fully automated system might be something to consider for achieving a state-of-the-art way of identifying windmill parks.

References

Copernicus land monitoring service. https://land.copernicus.eu. Accessed: 2021-12-23.

Deep learning for rare energy infrastructures in satellite imagery. https://bigdata.duke.edu/projects/deep-learning-rare-energy-infrastructures-satellite-imagery, 2020a.

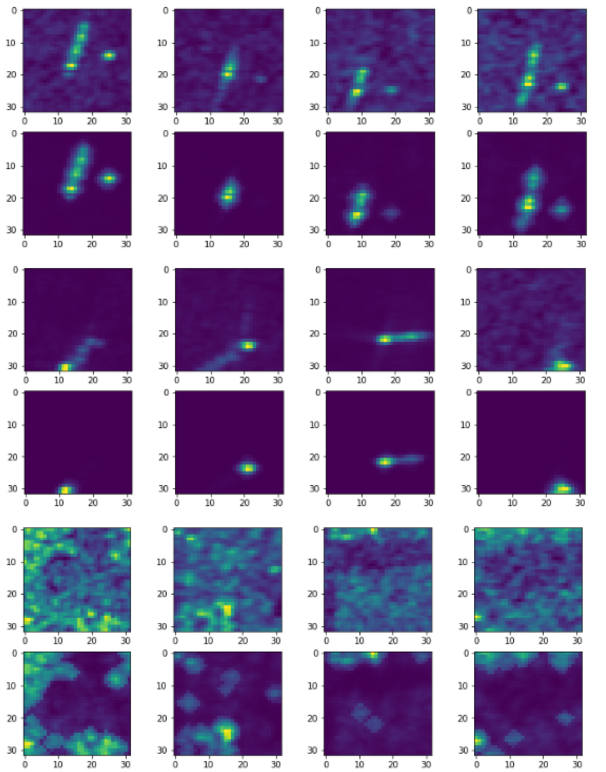

Figure 11. A comparison between noise reduction showing windmills, boats and land based imagery.

Github deep learning for rare energy infrastructures in satellite imagery. https://dataplus-2020.github.io, 2020b.

Hu, Y., Zhang, Q., Zhang, Y., and Yan, H. A deep convolution neural network method for land cover mapping: A case study of qinhuangdao, china. *Remote Sensing*, 10(12), 2018. ISSN 2072-4292. doi: 10.3390/rs10122053. URL https://www.mdpi.com/2072-4292/10/12/2053.

Moraite, D. Detecting ships in satellite imagery, Apr 2019. URL https://medium.com/dataseries/detecting-ships-in-satellite-imagery-7f0ca04e7964.

Sagar, A. Deep learning for ship detection and segmentation. https://towardsdatascience.com/deep-learning-for-ship-detection-and-segmentation-71d223aca649, Nov 2019.

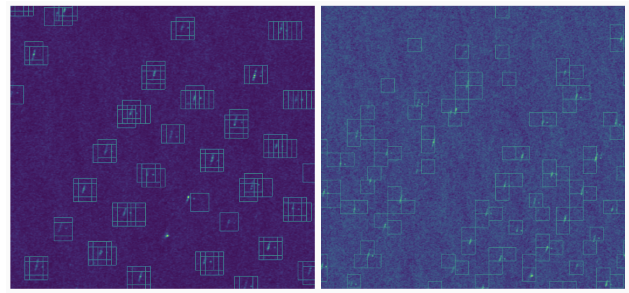

Figure 12. Squares indicate a detected windmill, in the left picture every windmill is plotted, on the right we first ran our filtering algorithm to include any close neighbors before adding them to our results.

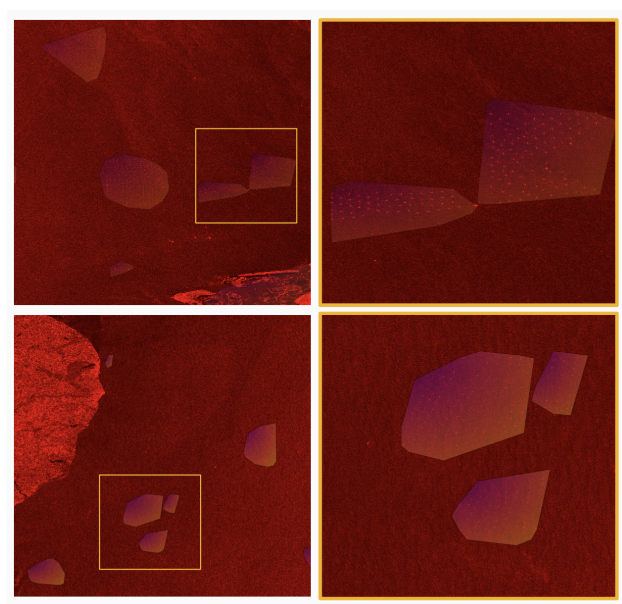

Figure 13. Final result of our algorithm parsing west of Denmark (top) and east of England (bottom). Right images shows an enlarged view clearly showing how the clustering creates a boundary even when parks are close to each other.

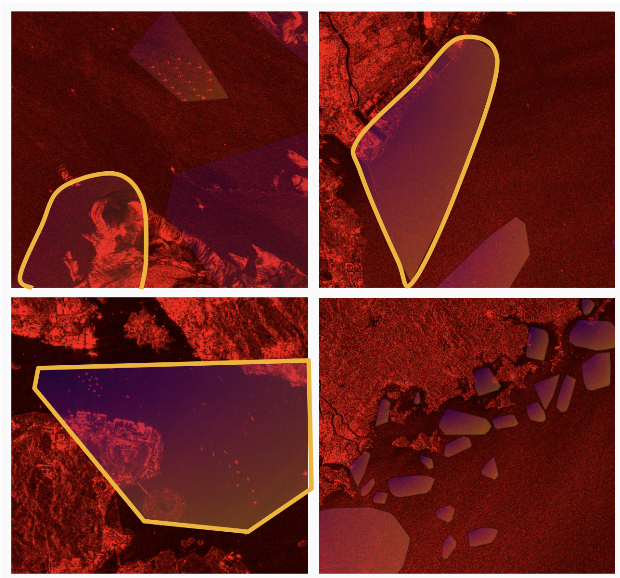

Figure 14. Model performing not as expected in coastal areas in Denmark (top) and harbors around China (bottom).

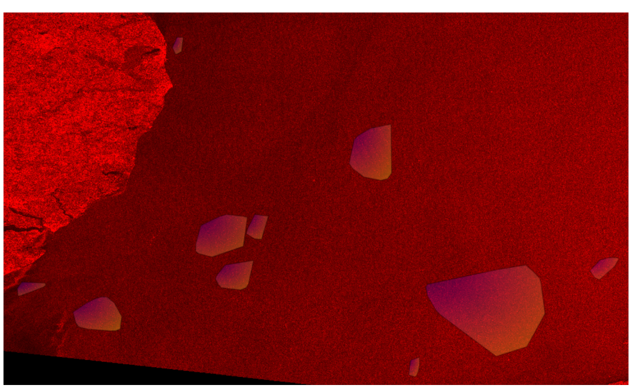

Figure 15. Result North Sea area.