Final Project Report

Detecting Ships in an Image using Image Segmentation

Name (姓名)：Christopher Harth-Kitzerow

Student No. (学号)：J11903099005

Abstract

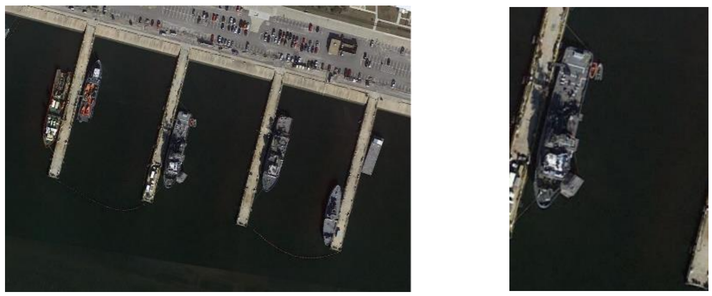
This project uses color thresholding, morphological filtering and proposes ship detection algorithms to recognize ships in an image. To achieve segmentation, color thresholding is used on ocean color to binarize all non-ocean parts of the image. Ship detection algorithms are used to differentiate port, docks and sideboats from ships. Afterwards dilation and erosion lead to a clear segmentation without small unwanted objects in the segment.

Index Terms — Image Segmentation, Color Thresholding, Morphological Filtering, Erosion, Dilation, Ship Recognition

1. INTRODUCTION

The proposed image segmentation algorithm was designed for an image which shows an above view from different ships, small boats docking on a port. The main elements in the image consisted of the ocean, one large port, multiple docks, multiple ships and small boats and small objects spreaded out in the image. Recognizing the ships bears multiple difficulties: Ships are very close to docks, making it difficult to separate their shape from the dock. Also the docks have a similar rectangular shape as the ships. The ships have vastly different colors and even throw shadows on nearby objects such as the docks. Small side boats are often directly touching a ship, making it difficult to separate both boundaries. Small objects often have rectangular structures and are shape wise indistinguishable from ships or boats.

To overcome all of those challenges standard image segmentation algorithms [1] and deep neural networks [2] fail as they do not consider these obstacles. Thus, during this project multiple custom made algorithms had to be developed to ensure clean separation and recognition of each ship. This includes a custom algorithm to automatically rotate the image, recognizing port and docks, custom erosion [3] and dilation [4] algorithms that are made for shapes of the ships and a side boat segmentation algorithm to separate small boats from big ships in small physical distance. All algorithms were developed based on ground truths which are not only valid for this image but also generalizable for similar images. This ensures consistent, high-accuracy results which outperform current deep learning standards such as Yolov3 [5]. In the main image the segmentation procedure presented by this project successfully recognizes every ship and boat in the image. Figure 1 shows the original image.



(a) (b)

Figure 1: Original Image

2. THE DEVELOPED ALGORITHM

The flowchart of the presented segmentation algorithm can be seen in figure 2. All segmentation steps use ground truths which hold for similar images and are designed to be generalizable with parameter and threshold tuning. Green font color is used to indicate algorithms that have been developed by the author of the project, blue fonts indicates state of the art improvements in segmentation.

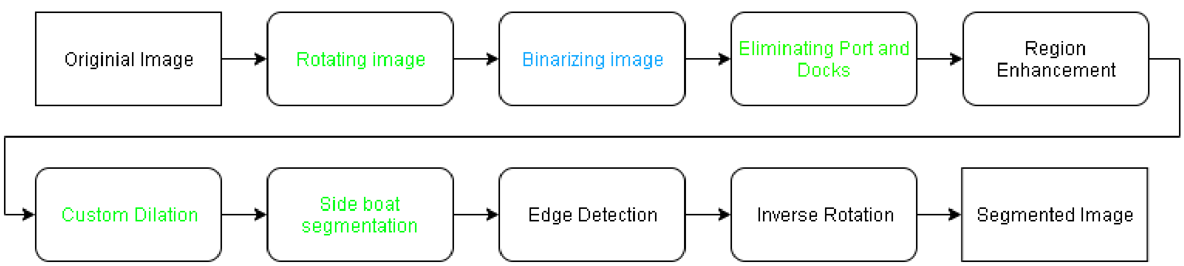


Figure 2: Segmentation algorithm flowchart

2.1. Image Rotation

The first step of the algorithm is to rotate the image. This is achieved by binarizing the image and using the horizontal port line for orientation. As the port is the largest object in each image of this kind, one can compare its horizontal width in a row before and after rotation (ships cannot be larger than docks or the port as they cannot dock in a port smaller than their own size). As a straight line will have more horizontal white pixels than a rotated line, pixel value maximization can be used as a target function for the custom developed automatic rotation algorithm. The author of this project calls the developed rotation algorithm binary search heuristic as the image is first rotated in both ways by a starting angle. This part represents a heuristic. Afterwards it is checked which of the three versions (not rotated, clockwise rotation, counter clockwise rotation) satisfies the target function best. The best version is used as the new basis for a rotation half of the intensity of last rotation. This part represents the binary search part of the developed algorithm as halving rotation in each step leads to O(log(n)) time complexity to find optimal rotation [6]. After reaching 0.5-degree rotation intensity, the algorithm terminates as this value is sufficient to perform further row and column based segmentation algorithms.

2.2. Image Binarization

The next step is binarization. In a general approach images get binarized with a threshold [7]. This is far from optimal as shadows and dark objects on the ships will be assigned a pixel value of 0, thus disappearing and disconnecting the ships. This project uses an approach which improves state of the art approach by using a color threshold for binarization. One ground truth for every image of this kind is that ocean color is consistent among the image and ships have to be located on water. Thus, using a color threshold based on ocean color is a suitable way to binarize the image [8]. This way the whole ocean gets assigned a pixel value of 0, while shadows and dark objects get assigned a pixel value of 1 as their color is slightly different from the ocean. Fine-tuning with a low threshold ensures optimal results. After binarizing one can clearly see the superior results of the color thresholding binarizing method over the state of the art binarizing approach.

2.3. Segmenting a port in an image

For separating the port from the image, another custom algorithm had to be developed by this project. As the image got rotated, one can utilize row and line based evaluation to segment and delete the port of the image. As previously discussed, a ground truth is that the port has to be the biggest object in the image. Thus a threshold after counting pixels in each line is sufficient to delete the largest object in the image.

2.4. Segmenting a dock in an image

For separating the docks from the image, this project proposes an algorithm based on the ground truth that docks have to be the largest vertical objects in the image after deletion of the port. That is due to the fact that ships cannot dock at docks which are smaller than their vertical boundaries. Thus, any docking ship has to be smaller than the largest vertical object in the image. Thus, a threshold after counting pixels in each column is sufficient to segment and delete docks in an image.

As small objects are besides the dock, the proposed algorithm also deletes neighbouring columns to get rid of those objects, thus performing a local rectangular erosion.

2.5. Region enhancement in an image

After deleting ports and docks, only ships, boats and small unwanted objects are left in the image. These can be deleted with a combination of erosion with different structuring elements and border clearing [9]. As these measures also delete surface area of the ships and transforms straight lines into criss cross lines, dilation with a small diamond structuring element is used to straighten and smooth the lines again. This should lead to deletion of small unwanted objects without affecting surface area and boundaries of ships and boats.

2.6. Custom dilation for ships in an image

Even with smoothing and dilation, certain ships elements may be unconnected. A general approach would be dilation with a large rectangular structuring element. However, this will not lead to satisfying results as parts of a ships can be too far apart from another to make dilation reasonable. Thus, a custom dilation algorithm had to be developed for this problem. The algorithms define two thresholds t1 and t2. It will count column pixels until white pixel values some up to t1. If they are found within the second threshold t2 then all pixels in the range of t2 will be connected with white pixels. This leads to the intended 1D dilation in vertical direction.

2.7. Side boat segmentation in an image

Even though by now only ships and boats should be left on the image, sideboats may be connected to a ship without any space in between. To separate those side boats from the ship nearby, another custom algorithm had to be implemented. It straightens the ships and counts white pixel values column by column in a line of 10 pixels from both sides. If within those 10 pixels the difference between the maximum pixel value in a column and another column is too high, it can only be due to reaching boundaries of a smaller boat. This insight can be used then to delete pixels between the ship and the boat at that point in order to separate their surface areas.

2.8. Edge detection and inverse rotation in an image

After performing all those steps, all unwanted objects and noise should be removed, ships are one connected surface area and side boats are separated from nearby ships. In a last step edge detection [10] is used to draw a border around each detected surface area. Also the image is rotated back and the borders are layed over the original image with red color.

3. Experiment And Result

In this section experimental results of using the proposed segmentation algorithm on the image and similar images are shown and explained.

3.1 Image Rotation

Rotating the image with the proposed binary search heuristic algorithm leads to a straight image in less than one second. The result can be seen in Figure 3.

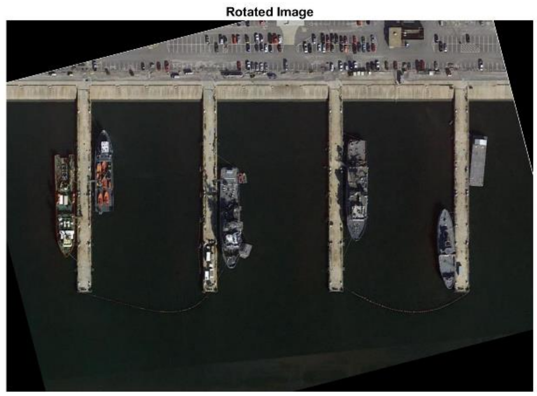


Figure 3: Rotated image

3.2 Image Binarization

When binarizing the image with a state of the art threshold approach, shadows and dark objects on the ships get deleted and detail is lost. When using a color threshold on the ocean color instead, ships stay mostly intact. One can compare results in Figure 4 and 5.

Figure 4: Binarized image

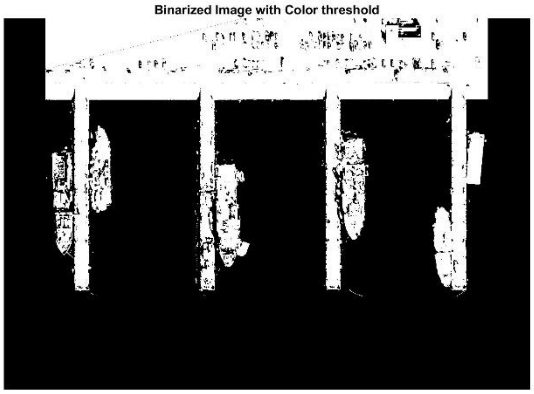


Figure 5: Binarized image with Color threshold

3.3 Eliminating the port

Eliminating the port of the image with the proposed port segmentation algorithm leads to a clean elimination of the port without any leftover noise or accidental deletion of non-port objects. Figure 6 shows the results after deleting the port.

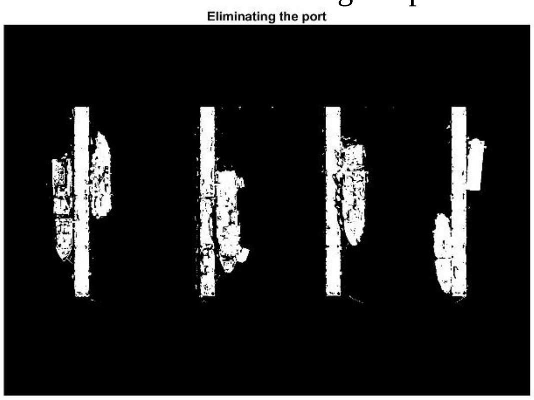


Figure 6: Image after eliminating port

3.4 Eliminating the docks

Figure 7 shows the test image after eliminating the docks. One can see that even with the built in erosion, leftover noise remains on the image which has to be removed during the next steps.

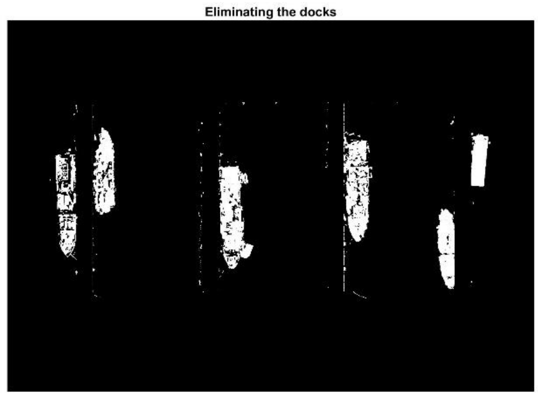


Figure 7: Image after eliminating docks

3.5 Region enhancement

To get rid of the noise, first erosion is used. As there is also small noise near to the boundaries of the ships, border clearing is used to remove this noise. As global erosion unavoidably also deletes parts within a ship, hole filling is used to revert this unwanted change. Border clearing leads to a criss cross border that does not represent the straight line border of a ship accurately. This project recommends a special kind of dilation, with a small diamond as a structuring element to transform the criss cross line into a straight line. This has a smoothing effect and accurately restores borders. The results can be seen in Figure 8.

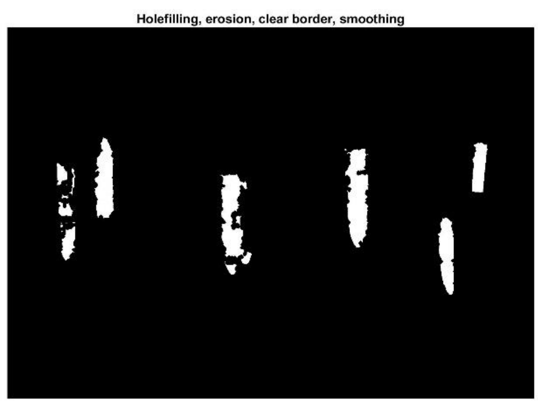


Figure 8: Image after region enhancement

3.6 Custom dilation for ships

Some ships in Figure 8, especially the one all the way to the left became disconnected surfaces due the previous steps of removing all unwanted objects in the image. This is reasonable and unavoidable if all noise has to be deleted. However, some ships are disconnected to an extent, that standard dilation does not achieve connecting those surfaces, even when a large rectangular structuring element is used that represents the shape of a ship. The developed custom dilation algorithm which is explained in section 2.6 on the other hand leads to a clear and clean connection of the former disconnected surface areas, without changing shapes of borders. In Figures 9 and 10 one can clearly see the outperformance of the custom dilation algorithm proposed by this project and regular dilation.

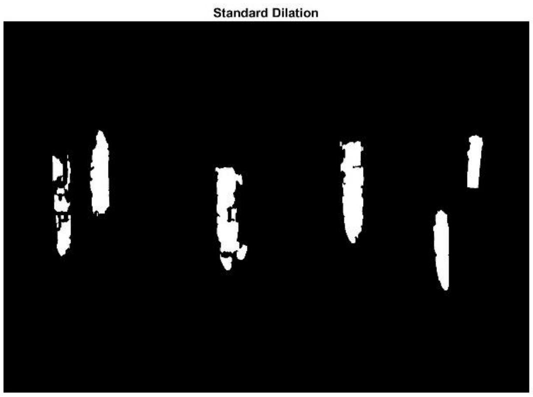


Figure 9: Rectangular dilation

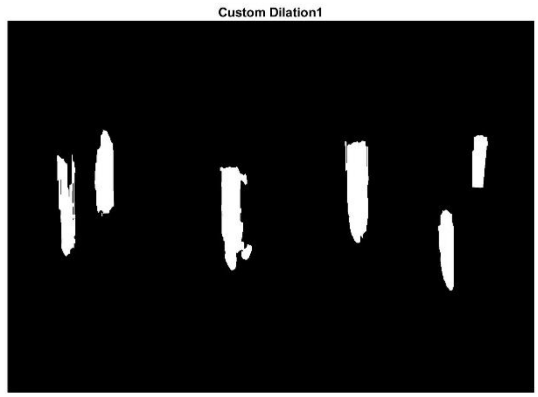


Figure 10: Custom ship dilation

3.7 Side boat segmentation

In Figure 10 one can see that due to their direct proximity, the ship in the middle and the two sideboats on the lower and upper right side next to it from one surface area. In order to disconnect these surfaces and segment each ship and boat separately, the proposed side boat segmentation algorithm of section 2.7 is used to disconnect the surfaces. The results show a clear distinction between all three objects and can be seen in Figure 11.

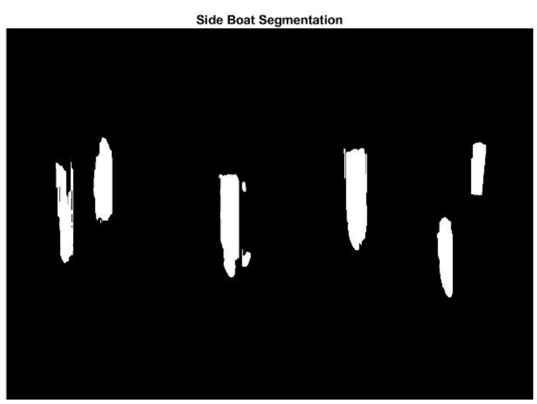


Figure 11: Image after side boat segmentation

3.8 Final results

The surface areas of all ships in Figure 11 are connected, distinct and no noise is left on the image. In this case, edge detection can result in a perfect result that can be seen in Figure 12 after inverse rotation. Using the detected shapes as a red overlay on the original images in Figure 13, one can see that the algorithms perfectly recognises all ships and boats distinctly and accurately highlights the shape of each ship.

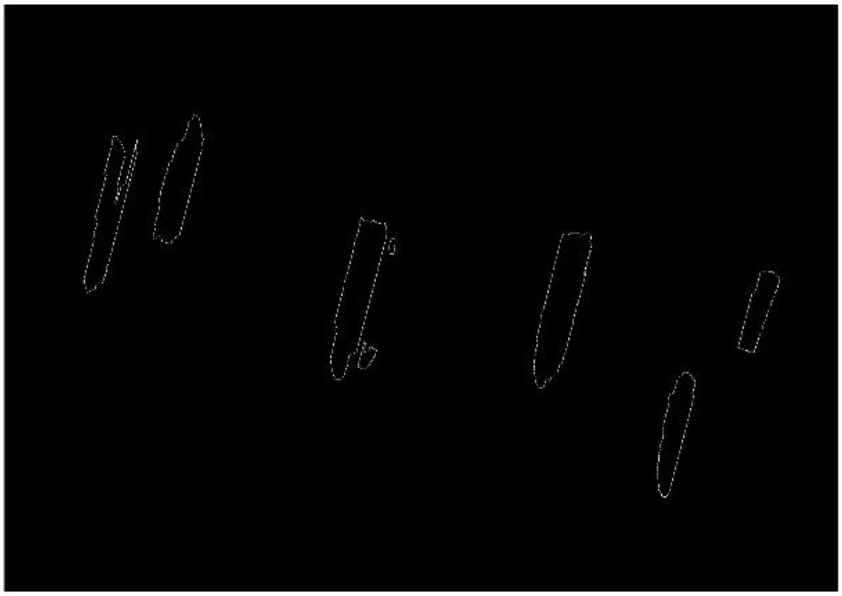


Figure 12: Image after edge detection and inverse rotation

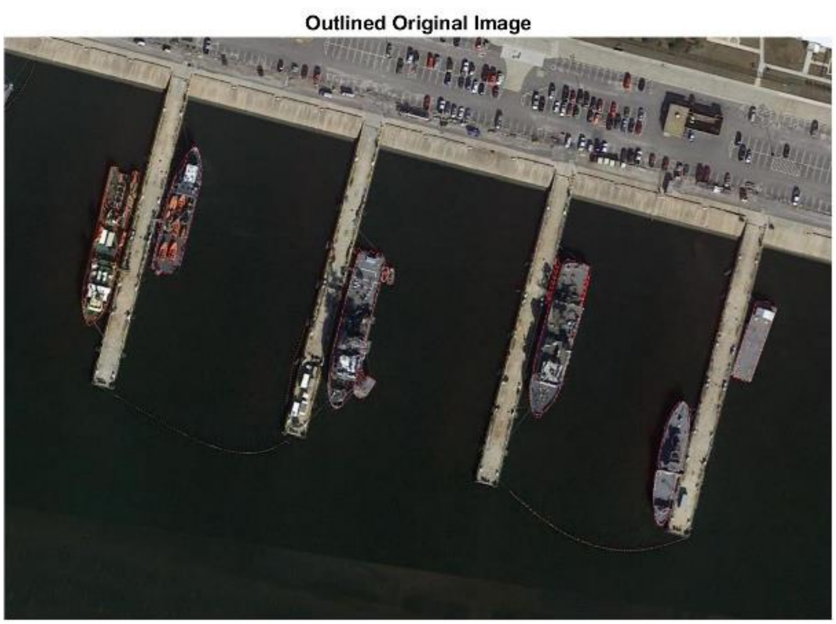


Figure 13: Outlined Original Image (Red outline)

4. DISCUSSION

4.1 Results with Yolov3 Deep learning network

As deep learning is very popular in image recognition and performs outstanding results in certain use cases with large amounts of data, a comparison of ship recognition with a deep neural network and the proposed image segmentation approach is reasonable. In Figure 14 one can see the results after using the Yolov3 Deep neural network on the test image. As one can see, Yolov3 fails to detect one big ship, all of the small boats and falsely detects a dock as a ship. This is a significant underperformance compared to the proposed segmentation algorithm. Also, Yolov3 does not include shape or diameter attributes for further classification or processing. Another disadvantage of a deep learning model and especially the complex Yolov3 is that it is a black box. No one can explain why exactly it fails. A hypothesis for a final comparison statement can be that Yolov3 is more generalizable for highly different images with a trade-off in lower accuracy, information and human understanding in similar images. Similar images are compared in the next section.

Figure 14: Deep learning ship recognition of Yolov3 model

4.2 Comparing results with similar images

Even though the task consists of only developing a segmentation algorithm for one specific image, the author of this project took additional effort to evaluate the proposed algorithm and Yolov3 on different similar images. At first, it should be noted that finding similar images which are 2D and from above is hard as search engine results will almost only contain 3D images which are not suitable for comparison. The following three images visible in Figure 15 have been found and tested with both methods.

(a)

(b)

(c)

Figure 15: Test images

Image a of figure 15 shows one ship, image b shows 4 ships and one boat and image c shows 12 ships and 2 boats. One should note that whenever using the proposed algorithm of this project, parameter tuning for several steps such as side boat segmentation is necessary. The data in table 1 shows result after parameter tuning for each of the images.

Table 1: Number of ships and boats (smaller than ships) detected by human eye, Yolov3 and the proposed algorithm

Image/#ships,boats	Human	Yolov3	Algorithm
A	1,0	2,0	2,0
B	4,1	2,0	4,1
C	12,2	10,0	12,4

Table 1 shows that Yolov3 overall failed to recognize 4 ships and all boats (smaller than ships. Also it recognised on piece of rectangular concrete as a ship in image a. The proposed algorithm by this project did not fail to recognize any ship or boat after parameter tuning but recognized 3 times noise or other objects as ships or boats. This is not surprising as only small noise gets removed in the image as the algorithm is not made for mid-size unwanted objects which are of similar size of ships but noticeably smaller than docks or the port. The results indicate that for similar 2D images shot from above and after parameter tweaking, the proposed algorithm outperforms Yolov3. One reason is surely the lack of data for these kind of 2D images. Given the amount of only 20000 different images which are vastly different in some cases which were used to train the Yolov3 Deep Learning model it is actually surprising how well Yolov3 performed.

4.3 Reasons for Yolov3's lower accuracy

As similar images to the test image are sparse, only three similar images have been tested afterwards. When tweaking parameters of the proposed algorithm it also recognizes each ship and boat in the three test images but falsely recognizes mid-size non-ship objects as ships in rare cases. The proposed algorithm outperforms a Yolov3 Deep neural network trained with 20000 ship images in all test cases, which is not surprising given the uniqueness of the test image. As ship image databases contain mainly 3D images of ships which are not taken from directly above it is actually surprising how the Yolov3 model performed still arguably well. However, Yolov3 has more disadvantages compared to the proposed algorithm apart from lower accuracy. It does not include shape or diameter attributes for further classification or processing which is possible with the proposed algorithm. Also while parameter tuning and code evaluation is possible for the proposed algorithm, Yolov3 is a black box and it is nearly impossible to say why exactly it fails at certain tests. Thus, for all tested images Yolov3 showed lower accuracy, less information and allowed for less human understanding. Most likely though, Yolov3 performs better at 3D or highly different images of ships compared to the proposed algorithm.

5. CONCLUSION

This project demonstrated how image segmentation with multiple custom made, novel segmentation algorithms, custom made morphological filtering methods [11] and state of the art region enhancement methods can lead to clear segment of ships in an image with multiple ships docking in a port. For the presented algorithms ground truths have been used which should hold true for every similar image such as relative size of port and ships, shape of ships and consistent ocean color. During this project, 4 custom made, novel segmentation and image processing algorithms were developed and several state of the art improvements have been applied. This lead to a 100% recognition rate of every ship and small boat on the image with no false positive recognition of unwanted objects. The developed algorithms consisted of an efficient automatic rotating algorithm, a port and dock elimination algorithm, a ship dilation algorithm and a side boat segmentation algorithm. All algorithms have tweakable parameters to adjust for similar images. The method only has a 2 secs runtime per image on Windows 10, Intel Core i5-4200 @1.6GHz, 2C4T. Also, further classification possible due to shape and diameter information of ships and boats.

It can be concluded that the proposed segmentation algorithm by this project uses improvements over state of the art approaches such as binarization with the ocean as a color threshold and custom made segmentation algorithms such as side boat segmentation to achieve a perfect result on the test image and even a nearly perfect result on three similar images even though the task did not demand it. In the main test image all ships and even small side boats were detected with boundaries as well as their whole surface area and no false detection was made. This means that compared to an AI solution, one could even use information as diameter and shape of each ship for further processing or classification. However, AI solutions might catch up in the future as authors of algorithms such as Yolov3 are working on masking objects instead of only showing bounding boxes [5]. Nevertheless, during this project it was shown that due to lack of large available data sets of 2D images from above of ships in a port, image segmentation outperforms current deep neural networks. Further tests may be helpful to support this claim.

REFERENCES

[1] Matlab Documentation, Image Segmentation, https://www.mathworks.com/discovery/image-segmentation.html
[2] Wikipedia, Deep Learning, https://en.wikipedia.org/wiki/Deep_learning
[3] Matlab Documentation, Erosion, https://uk.mathworks.com/help/images/ref/imerode.html
[4] Matlab Documentation, Dilation, https://uk.mathworks.com/help/images/ref/imdilate.html
[5] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. arXiv, 2018.
[6] Wikipedia, Binary search algorithm, https://en.wikipedia.org/wiki/Binary_search_algorithm
[7] Matlab, Documentation, Binarizing with threshold, https://www.mathworks.com/help/images/ref/imbinarize.html
[8] Matlab Documentation, Color Thresholding, https://uk.mathworks.com/help/images/ref/colorthresholder-app.html
[9] Matlab, Documentation, Border Clearing, https://www.mathworks.com/help/images/ref/imclearborder.html
[10] Matlab Documentation, Edge Detection, https://www.mathworks.com/discovery/edge-detection.html
[11] Matlab Documentation, Morphological Filtering, https://www.mathworks.com/help/images/morphological-filtering.html

APPENDIX: ENLARGED ALGORITHM FLOWCHART

Green: New Algorithm developed by author of this project (Novelty)
Blue: Improvement of state of the art approach