DTU Aqua
National Institute of Aquatic Resources

Development of mitigation strategies for control of Pacific oysters in Danish coastal waters

Appendices

Pernille Nielsen[1], Camille Saurel[1], Ciaran McLaverty[1], Patrick Joyce[1], Lone Madsen[1], Kamille E. Krause[1], Belén Jiménez-Mena[1], Einar E. Nielsen[1], Jonathan Gundorph[2], Katarzyna Gorazda[2], Daniel H. Olesen[2], Kerstin Geitner[1], Katla Hrund Björnsdóttir[3], Roberto Flore[3], Charlotte Colvin[4], Ian McCarthy[4], Chris Richardson[4], Antonio Agüera Garcia[1,5] and Pedro S. Freitas[1]

1 DTU Aqua, National Institute of Aquatic Resources, Technical University of Denmark
2 DTU Space, National Space Institute, Technical University of Denmark
3 DTU SkyLab FoodLab, Technical University of Denmark
4 School of Ocean Sciences, College of Natural Sciences, Bangor University, UK
5 Current address: Institute of Marine Research, Department for Benthic Resources, Norway.

DTU Aqua Report no. 414a-2022

11. Appendices to chapters 2, 3, 4, 7, 8 and 9

Appendix 2.1: Monitoring and distribution of Pacific oysters in the Limfjorden and Isefjord

Appendix 3.1: Student report: Automatic detection and estimation of submerged oysters from drone images

Appendix 3.2: Student report: Methods to estimate oyster coverage by using different algorithms

Appendix 4.1: Genetic tables

Appendix 7.1: Stakeholder interview questions

Appendix 8.1: Tables mini-dredge assessment

Appendix 8.2: Tables floating escavator assessment

Appendix 9.1: Gastronomic possibilities of large Pacific oysters

Appendix 2.1: Monitoring and distribution of Pacific oysters in the Limfjorden and Isefjord

Table A2.1. Estimates of Pacific oyster population biomass (tons) in the deep areas (> 3 m depth) in bivalve fishing area (Fiskerstyrelsen) of the Limfjorden from 2017 to 2022. Empty cells mean no sampling of that fishing area in that year.

Fishing Area		Number	Area	Population Biomass (tons)					
Number	Name	Stations	km^2	2017	2018	2019	2020	2021	2022
1	Nissum Bredning, Sydvest	47	49.79	40.4	23.5	19.0	2.5	27.4	27.9
2	Nissum Bredning, Nordvest	40	46.00	44.1	58.4	51.9	53.5	78.7	99.4
3	Nissum Bredning,Sydøst	17	20.11	0.0	5.8	0.0	1.1	0.0	4.3
4	Nissum Bredning, Nordøst	17	19.34	0.8	0.0	0.0	21.0	3.6	21.9
5	Venø Sund, Lavbjerg Syd	17	16.22	0.0	48.7	0.0	16.4	0.0	31.5
6	Venø Sund, Lavbjerg Nord	29	30.02	30.8	5.0	28.9	9.3	32.5	38.1
7	Venø Bugt, Nord	33	39.59	0.0	9.0	0.0	7.7	10.8	49.4
8	Venø Bugt, Syd	33	31.87	26.7	34.4	98.7	15.6	208.8	215.0
9	Kås Bredning	50	43.94	64.5	13.0	14.1	19.2	135.4	16.5
11	Salling Sund, Syd	12	12.44	16.0	15.7	6.3	2.8	173.5	0.0
12	Lysen Bredning	7	6.13	35.3	18.3	94.1	19.1	14.7	5.9
13	Salling Sund, Nord	10	10.16	8.0	3.0	22.8	7.7	16.4	0.0
15	Sønder Bredning	15	30.36	35.1	57.0	13.9	21.6	0.0	7.9
16	Øster Bredning	11	46.21	0.0	0.0	0.0	0.0	23.7	0.0
17	Risgårde Bredning, Vest	3	20.02	0.0	0.0	0.0	0.0		0.0
18	Risgårde Bredning, Øst	4	22.72	0.0	0.0	0.0	0.0		0.0
19	Hvalpsund	6	14.88	0.0	0.0	0.0	0.0		0.0
20	Lovns Bredning, Vest	9	27.39	0.0	0.0	0.0	0.0	0.0	0.0
21	Lovns Bredning, Øst	14	23.27	0.0	0.0	0.0	0.0	0.0	0.0
23	Mors, Vest	3	12.61	8.4	0.0	0.0	0.0		0.0
24	Nees Sund	2	6.47	0.0	0.0		0.0		0.0
25	Visby Bredning	1	19.14	0.0	0.0		0.0		0.0
26	Dragstrup Vig	1	18.30	0.0	0.0		0.0		0.0
27	Vilsund	2	12.04	0.0	0.0		0.0		0.0
28	Thisted Bredning, Vest	8	32.59	0.0	0.0		0.0		0.0
30	Thisted Bredning, Sydøst	7	27.48	0.0	0.0		0.0		0.0
32	Feggesund / Hovsør Havn	8	14.04	0.0	0.0		0.0	23.1	26.7
33	Løgstør Bredning, Vest	35	40.51	1001.4	512.3	1012.7	490.9	495.6	189.0
34	Løgstør Bredning	20	50.37	553.9	668.5	1436.5	631.4	199.0	78.9
35	Livø Bredning, Vest	21	46.26	0.0	20.7	47.9	0.0	8.4	30.0
36	Livø Bredning, Øst	18	36.21	0.0	116.5	7.4	39.9	35.4	44.5
37	Bjørnsholm Bugt	13	35.76	12.8	0.0	0.0	13.4	7.4	0.0
38	Løgstør Bredning, Øst	15	34.23	6.9	0.0	0.0	0.0	18.0	43.3
39	Løgstør Grunde	10	34.67	0.0	0.0	3.7	0.0	0.0	0.0

Table A2.2 Location of shore sites surveyed in 2019 and 2020 in the Limfjorden.

2019: Station number	Basin	Fishing Area	Latitude	Longitude	N
DSC	Salling Sund	13	56.7907	8.87469	9
1	Thisted	32	56.9754	8.92	15
2	Løgstør	33	56.9748	8.934	9
4	Løgstør	33	56.9451	8.90807	9
6	Livø	35	56.9119	8.91426	9
9	Dråby	14	56.8829	8.84245	15
Dråby Vig	Dråby	14	56.8658	8.82817	15
13	Dråby	14	56.8306	8.84411	11
14	Dråby	14	56.8105	8.86758	13
15	Kås	9	56.6823	8.70587	15
Hesterørøddevej	Agerø	23	56.6701	8.64911	15
18	Kås	9	56.6922	8.76948	15
20	Salling Sund	11	56.7109	8.77634	15
Salling Sund B	Salling Sund	11	56.7307	8.81457	15
Salling Sund A	Salling Sund	11	56.7483	8.83254	15
26	Agerø	23	56.7051	8.61664	13
27	Agerø	23	56.7201	8.62577	9
28	Agerø	24	56.7236	8.57441	9
29	Thisted	30	56.9384	8.84028	8
31	Thisted	30	56.9034	8.81576	8
33	Thisted	28	56.8933	8.7475	9
34	Thisted	28	56.8895	8.73329	7
35	Thisted	28	56.8877	8.68596	7
37	Visby-Vilsund	27	56.8828	8.64011	6
39	Visby-Vilsund	27	56.8628	8.64734	5
40	Visby-Vilsund	27	56.8385	8.64056	15
42	Visby-Vilsund	27	56.8325	8.63126	9
44	Visby-Vilsund	26	56.8178	8.64483	9
46	Visby-Vilsund	26	56.8106	8.66562	15
47	Visby-Vilsund	26	56.7922	8.61436	13
50	Visby-Vilsund	25	56.7742	8.55187	11
51	Visby-Vilsund	25	56.7575	8.55734	9
53	Visby-Vilsund	25	56.7387	8.50447	9
Branden	Sønder	15	56.797	9.02607	15
55	Sønder	15	56.802	8.96284	15
56	Salling Sund	13	56.7805	8.91959	9
57	Salling Sund	13	56.7516	8.86807	15
59	Salling Sund	11	56.7228	8.83758	15
Harre Vig	Lysen	12	56.7091	8.89731	13
Lysen Bredning	Lysen	12	56.687	8.84077	22
63	Øster	16	56.8021	9.06298	15
64	Øster	16	56.7891	9.10377	15
65	Livø	35	56.8436	9.06809	15

2020: Station number	Basin	Fishing Area	Latitude	Longitude	N
66	Livø	35	56.8391	8.97772	15
68	Kås	9	56.6786	8.78953	9
70	Kås	9	56.6322	8.742	12
72	Venø Sund	6	56.6076	8.68916	9
73	Venø Bugt	7	56.5684	8.74249	9
74	Venø Bugt	7	56.5194	8.74044	12
75	Venø Bugt	8	56.4713	8.68527	11
Struer	Venø Bugt	8	56.4997	8.61219	16
77	Venø Sund	5	56.5445	8.57332	9
78	Venø Sund	6	56.5844	8.54433	9
79	Venø Sund	6	56.6133	8.59229	10
80	Nissum	4	56.5625	8.55465	9
81	Nissum	1	56.5473	8.49743	1
82	Nissum	1	56.5577	8.47052	8
83	Nissum	10	56.5693	8.30642	12
84	Nissum	10	56.5872	8.30366	11
86	Nissum	Udf	56.6003	8.25048	1
87	Nissum	Udf	56.6254	8.23145	1
89	Nissum	216	56.7464	8.25163	3
90	Nissum	216	56.7696	8.28038	9
91	Nissum	216	56.7573	8.29917	9
92	Nissum	216	56.7193	8.31327	11
93	Nissum	2	56.6922	8.34923	9
94	Nissum	2	56.6812	8.35552	1
95	Nissum	2	56.6704	8.40145	9
96	Nissum	4	56.6368	8.47626	9
97	Nissum	4	56.62	8.49722	9
99	Nissum	4	56.5908	8.51465	11
103	Agerø	24	56.7199	8.47371	1
105	Visby-Vilsund	25	56.7939	8.48507	9
107	Visby-Vilsund	27	56.8733	8.62627	9
109	Thisted	28	56.9495	8.73148	11
110	Thisted	30	56.9598	8.79121	1
111	Thisted	32	56.9823	8.84069	1
115	Venø Bugt	7	56.55	8.63247	12
116	Venø Sund	5	56.5524	8.61214	12
117	Venø Sund	5	56.5773	8.63494	11
118	Kås	9	56.6503	8.6338	9
119	Nissum	1	56.5772	8.39496	8
Agger Tange	Nissum	216	56.7209	8.25722	15
Kommune	Salling Sund	13	56.7917	8.85833	35
Fur	Øster	16	56.8068	9.02175	7
Rønland	Nissum	Udf	56.6718	8.21792	16
Rønland	Nissum	Udf	56.6699	8.21383	3

Appendix 3.1: Student report: Automatic detection and estimation of submerged oysters from drone images

Technical University of Denmark

30220 - Synthesis project

Automatic detection and estimation of submerged oysters from drone images

In recent years, native Danish Limfjord Oysters have increasingly been outcompeted, in terms of food and space, by the invasive species of the Pacific Oyster. In response, Dansk Skaldyrscenter (DSC), a part of DTU Aqua, has taken steps to try and prevent this, by removing the Pacific Oysters where ever they can, to strengthen the native Limfjord Oyster's position in the Limfjord. One strategy they have employed is using drones to fly along the coastlines and photograph the oysters through the water, to get an estimate of how the environments look from place to place, in order to know where to concentrate their efforts. This has led them to hire researchers and drone pilots affiliated with DTU Space, to develop an algorithm that may help them get an estimate of the number of oysters in each image with the use of image analysis methods. Furthermore, the use of multi-spectral cameras, and their performance compared to optical cameras, is also one of the main objectives, in order to investigate if multi-spectral cameras are better suited for the detection of objects through the water surface.

This report details some of the efforts employed to aid DSC in the detection of the oysters in the captured drone images.

Contents

List of Figures

1 Introduction

Oysters are a salt-water bivalve mollusc that live in marine or brackish habitats. They are mostly irregular in shape, and comes in a wide spectrum of sizes and colors. The oyster is an important organism for many underwater ecosystems, and provides habitat for a number of other marine species. The oyster can significantly improve the quality and clarity of water around it, by removing plankton and particles from its water column, and it can also sometimes produce pearls, although most are not very valuable. In the Limfjord of Denmark, it is said that the best oysters in the world can be found. They are quite rare, and in fact, the oyster beds found just outside the Dansk Skaldyrcenter (DSC) are part of the last large colony of wild European oysters left in the world [NIELSEN-BOBBIT, 2020]. However, oysters are not always a welcome sight in marine habitats. For instance, the local, native oyster (also known as the Limfjord Oyster) is currently being out-competed by the more aggressive Pacific Oyster, which replicates at a rate of 100 times that of the Limfjord Oyster. The Pacific Oyster differs in shape, often being more elongated compared to the round shape of the Limfjord Oyster. Due to the growing threat against the Limfjord Oyster, the Pacific Oyster is now considered an invasive species in Denmark, and must be removed in order to protect the local Limfjord Oysters of Denmark.

To increase their efforts to effectively locate and remove the Pacific Oysters, Dansk Skaldyrscenter (DSC), which is a part of DTU Aqua, has hired researchers and drone pilots affiliated with DTU Space to develop an algorithm based on drone images of the oysters, that can quickly segment what are oysters and what are not oysters in the images. Furthermore, the algorithm should provide an estimate of the number of oysters present, and where exactly they are located. This will give Dansk Skaldyrscenter a much better overview of the current situation, and an edge in their efforts to remove the Pacific Oysters for good. [Stubgaard, 2020]

Figure 1: Pacific Oysters (top) versus Limfjord Oysters (bottom)

In addition to this task, another objective is to make a qualitative estimate of the use of a multi-spectral camera, versus a normal, optical camera, which is usually mounted on a drone. The motivation behind this objective is that a multi-spectral camera should, in theory, be able to penetrate deeper into the water, with less attenuation, and thus more effectively be able to provide visibility of the oysters present, so the algorithm can segment the oysters with a higher accuracy. But, as is elaborated upon later in this report, using a multi-spectral camera comes with its own set of challenges.

1.1 Sites of interest

The Dansk Skaldyrscenter has six primary sites of interest, where Oysters can be found. These, along with their main features, are illustrated below:

- Clustered & individual population structures
- Small slope
- No reef structures yet
- Easy to classify
- Lightest bottom sediment, larger oysters contrast

Figure 2: Lyssen Bredning, Limfjord

- Subtitdal
- Large & healthy reef areas
- Site with the deepest dense-type populations (-100cm)
- Dense populations in shallower areas
- Lots of macroalgae
- Sandy/muddy bottom, small rocks/pebble areas

Figure 3: Klosterbugten, Limfjord

- Location of DSC
- Most shallow reef
- Steeper slope
- Rocky bottom, pebbles and sand at deeper end
- Oysters look similar to bottom substrate
- Difficult site to classify
- Many oysters dead, mortality cause unknown

Figure 4: Strandevejen, Limfjord

- Within intertidal zone
- Very large mixed reef with blue mussels
- Not on the shore

Figure 5: Agger Tange, Limfjord

- Small slope
- Only small patches of reef
- Many dead, white shells
- Low energy area, small tides
- Reasonable amount of macroalgae and blue mussels.
- Different kind of algae species

Figure 6: Branden, Limfjord

- Larger tidal amplitude (ca. 2m)
- Significant wave, storm and surge action
- Oyster populations in intertidal zone
- Reefs single-species or mixed with blue mussels
- Significant height change relative to bottom substrate
- Mainly sandy bottom

Figure 7: Wadden Sea, southwestern Jutland

All except the Wadden Sea, which lies in the southwestern part of the Jutland Peninsula, are located in the Limfjord area of northern Jutland. The Limfjord differs from the Wadden Sea, as it is a microtidal system, with a small tidal amplitude between 15-50cm, and the oyster populations here all occur in the subtidal zones, emerging only when the water levels drop due to the wind/sea level pressure changes. But the sites in the Limfjord area may also differ from each other a great deal, as can be seen in Figure 2-7 above. For example, Branden has many types of algae and different oyster structures mixed with blue mussels, in small reef patches. Meanwhile, Klosterbugten has large and healthy reef areas, many at much larger depths. Furthermore, a site like Strandevejen has a rocky bottom that looks similar to the oysters present there, while Lyssen Bredning has a much lighter bottom sediment, making the oysters stand out much more clearly. These

trait differences are important to take into consideration, and will yield different results when trying to segment and classify the oysters at the different sites.

2 Method

2.1 Hardware

The pictures of the oyster beds are taken by a drone, with a mounted payload of two cameras: The MicaSense RedEdge-MX multi-spectral camera, and a standard Go-Pro camera. The payload with these two cameras, as well as the type of drone that carries them (DJI M-600), can be seen in Figure 8

Figure 8: Left: M-600 drone, right: Payload setup: GoPro camera (left camera) and MicaSense RedEdge-MX (right camera)

The payload setup seen in Figure 8 allows the drone to take two pictures simultaneously, one with each camera. In this way, both pictures will be taken at the same time, and from roughly the same angle, which will make a comparison between the two cameras feasible. this is illustrated, side by side, in Figure 9.

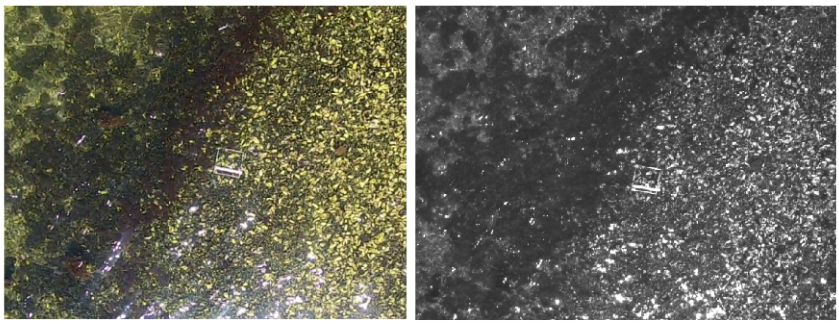


Figure 9: GoPro picture (left) and MicaSense RedEdge-MX picture (right)

In the comparison seen in Figure 9, band one of the MicaSense RedEdge-MX camera is used, but there are in fact five bands. Those five bands can be seen illustrated in figure Figure 10, and show vastly different perspectives of the same area. The reason for this difference of perspective is that each band has its own, distinct wavelength (see caption in Figure 10).

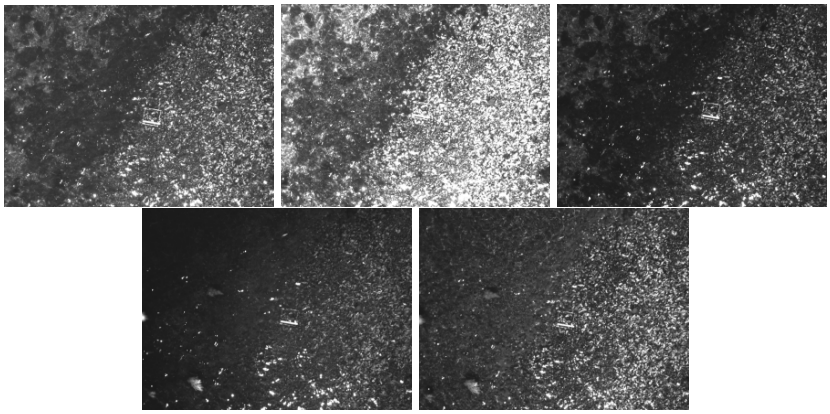


Figure 10: Comparison of the five MicaSense RedEdge-MX spectral bands.
From left to right: Blue (475nm), Green(560nm), Red(668nm), Infrared(840nm),RedEdge(717nm)

Upon examination of Figure 10, note that in some of the bands with the higher wavelengths, the large rocks are visible on the bottom, while in the lower wavelength bands they are completely invisible. This allows the choice of spectral band to depend on what one is looking for.

The MicaSense RedEdge-MX camera can be seen in Figure 11, along with the DLS (Downwelling Light Sensor) 2. The DLS 2 is an instrument that provides accurate and reliable data, significantly reduces the need for post-processing and improves the radiometric accuracy. The DLS 2 also contain an integrated GPS, making the setup procedure more simple. The DLS 2 records data on the amount of light from the sky for each of the 5 bands of the Micasense RedEdge-MX camera, and captures this data throughout the flight (embedded within the metadata of each image for each band). In post processing, data from the DLS can be used to correct for changing illumination conditions during flight. The data it collects is based on its measures of irradiance, which is heavily dependent on the sensors' orientation relative to the sun as it flies. For example, a light sensor pointing directly at the sun will measure a different value than one pointing straight up at the top of the sky. This is why it is important to have a correctly calibrated magnetometer, which can provide heading and orientation information to help specialized processing software (like Agisoft Metashape) to make better decisions [Support, 2020d], [Support, 2020a].

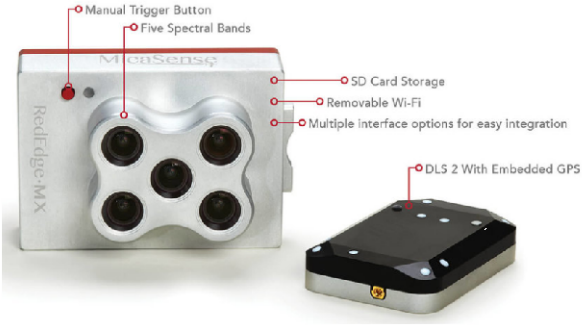


Figure 11: The Micasense RedEdge-MX camera (left) alongside the DLS 2 sensor (right) [Support, 2020b]

2.2 Theory

The wavelengths of the different spectral bands differentiate greatly from each other as they serve different purposes. It is important to look into, as some of these bands will potentially be able to penetrate deeper into the water column than a normal RGB-camera. One of the main objectives of this report is to investigate if this is the case. An illustration of the wavelengths covered by the five spectral bands, as well as their bandwidths, can be seen in Figure 12.

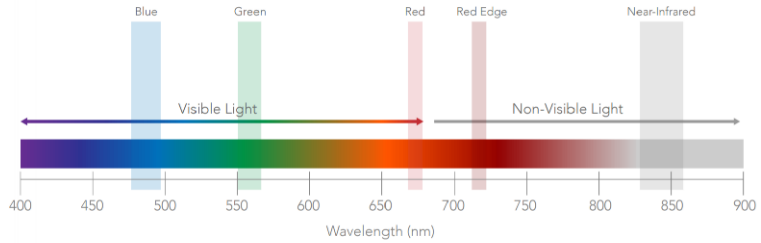


Figure 12: The wavelengths of the five spectral bands in the MicaSense RedEdge-MX camera [Support, 2020c]

Lower wavelengths should, in principle, allow for less attenuation in the water columns. Due to this fact, this makes the blue and green bands of the MicaSense RedEdge-MX camera especially promising for water penetration. This is elaborated upon further in Figure 13:

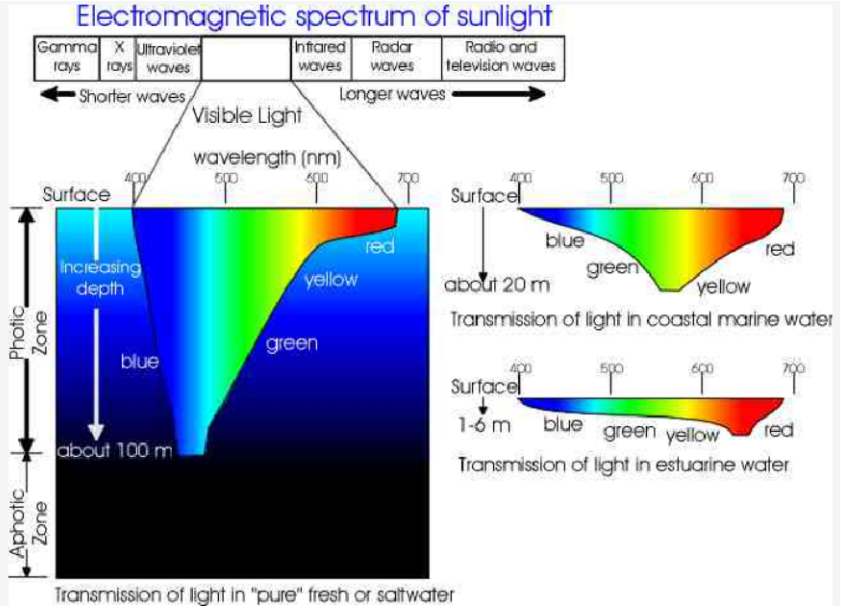


Figure 13: Illustration showing the amount of light at different depths. [Reef2Reef, 2016]

As seen in Figure 13, the blue band should penetrate far deeper than any of the other bands. The reason for this is that blue light is attenuated and absorbed the least in the water. The deeper into the ocean light travels, the more the electromagnetic light is absorbed, which is why the ocean becomes a more intense shade of blue with increasing depth, as all the other wavelengths are being absorbed. So intuitively, one might expect that there should always be the greatest visibility in the blue band of the multi-spectral camera. But it is important to keep in mind that the pictures seen in Figure 10 are taken at a quite shallow depth, with the deepest depth being around 2-3 metres, where not much of the green light has been absorbed yet. Furthermore, note that the green part of the light distribution is broader than the blue in the beginning, and it only becomes thinner deeper beneath the surface. Additionally, Figure 13 illustrates that transmission of light in coastal marine water environments actually has a

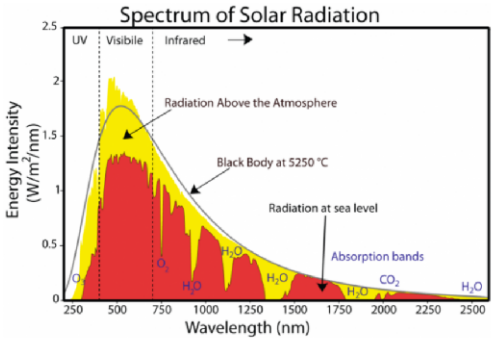


Figure 14: Distribution of light intensity across the different wavelengths. [Timothy Bralower, 2020]

larger amount of green light than blue light. And finally, as seen in the spectrum of sunlight in Figure 14, there is in fact a larger amount of light intensity in the green part of the spectrum than in the blue part, peaking between λ= 500-550nm and then decreasing thereafter. Even though the blue band should have the least amount of attenuation, the intensity of light is a bit lower here at shallow depths, because there isn't as much blue light as green light in a ray of sunlight. So this could be an explaining factor that the green band has more visibility/ reflectance in very shallow locations, while the difference then diminishes rapidly at increasing depths, as explained by Figure 13 and Figure 14.

To conclude, the choice of the spectral band that is best used for detecting oysters might not be as simple as saying the blue or the green band, but would be a choice depending on the depth of the oysters, as well as other factors. Considering that most oysters lie at a depth of around 0-3 meters, one might be tempted to say that the green band would be preferable to use for classification of oysters at these depths, but then again, as seen in Figure 15, the increased reflectivity is a double-edged sword, as there are also significantly more distortion from light reflecting off the surface of the water, and the metal quadrant looking very similar to the oysters as well.

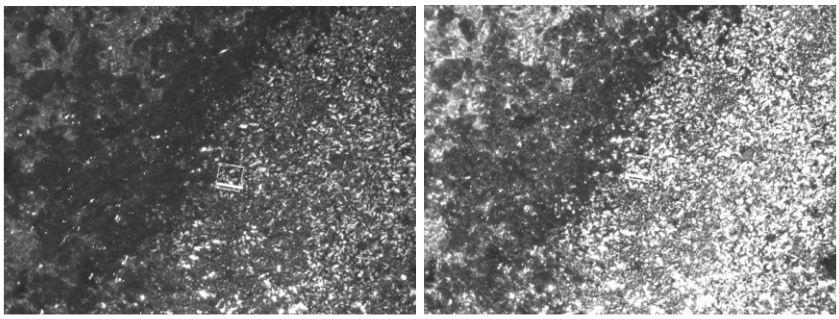


Figure 15: Comparison between MicaSense blue band (left) and green band (right)

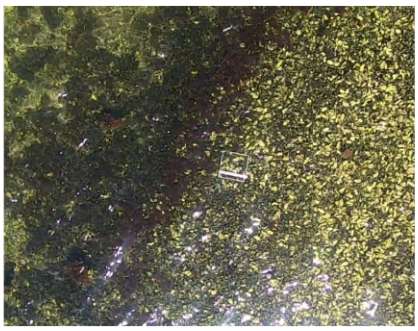


Figure 16: GoPro optical picture of the same area

Another constraint that is important to take into consideration when flying with the MicaSense RedEdge-MX camera, is that water bodies are chaotic environments, and contain disturbances from sun glare, waves, refraction and reflection effects, etc. Additionally, the exposure time and gain is adjusted automatically within the MicaSense RedEdge-MX camera based on what it detects. But this adjustment is built for agricultural field monitoring and not water environments, which can make the pictures appear very dark. It is, however, possible to adjust the gain and exposure time manually for each band. The exposure time and gain are inversely related, when it comes to finding the best visibility conditions in water bodies. In other words, when lowering the exposure time, one must raise the gain. After some trial-and-error experiments in the field, it was found that the MicaSense RedEdge-MX camera has the best visibility through water with an exposure time of 7.8 ms and 2x gain (ISO 200), for depths of 1-3 meters.

3 Machine learning, Classification and Segmentation

The concept of *Machine Learning* (ML), which is an important subarea of Artificial Intelligence, is the implementation of an idea held by Alan Turing: *"How can we build intelligent machines?* Or more elaborately: *Can we construct a machine that can do the same things a human can do? and how may we do it?* Turing proposed an answer to this question. Instead of building a computer program that behaves like a human from scratch, we should build a machine which, at first point, cannot do a great many things, but is capable of *learning from past experiences*. This is the fundamentals of Machine Learning.

Within Machine Learning, there exist many different algorithms and approaches for solving a problem. The approach depends heavily on the type of the problem at hand. If the user has an existing ground truth, in other words a "target" of what the outcome should look like, the user would, in most cases, make use of *Supervised Learning*. However, if no such ground truths are available, the user would rather look at the statistical similarities of the data set, an approach which is known as *Unsupervised Learning*. These two categories will now be described more in-depth in the following subsections.

3.1 Unsupervised Learning

The concept of Unsupervised Learning is to learn without a teacher. In other words, one attempts to perform a clustering/grouping of data without having knowledge of the "truth" before hand (also known as a priori knowledge), and instead looks at the statistical similarities within the data, in order to group different observations into different classes. These groups are separated from each other in a so-called *feature space*, where the area that is spanned by each class is known as a *cluster*. Three examples of simple cluster types can be seen illustrated in Figure 17.

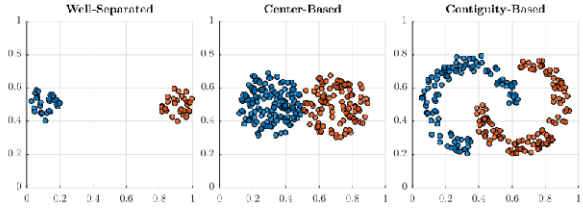


Figure 17: Three simple, distance-based cluster types. The color indicate the K = 2 clusters.

3.2 Supervised Learning

The concept of Supervised Learning is the process of classifying data based on some pre-defined ground truths. In its essence, the user makes use of a data set comprised of N observations, $x_1, \ldots x_N$, and N targets $y_1, \ldots, y_N$ and then attempts to develop a way to predict the targets y from the observations x:

$$y = f(\mathbf{x}, \mathbf{w}) + \epsilon \tag{1}$$

where w is a vector of tunable parameters and ϵ represents a noise term. The learning then consists of the selection of parameters w based on the training data, X, y.

Supervised Learning is limited in terms of scalability of the target function at hand. While Unsupervised Learning is the natural procedure for cognitive mammals (us), it might be different for a computer. So one must always adapt the strategy to the problem.

3.3 Classification results with Supervised Learning

There exists countless machine learning models, too many to mention here. Apart from the Supervised and Unsupervised Learning categories, there are also the Semi-Supervised Learning and Reinforcement Learning categories, although those will not be mentioned further in this report. To give the reader a better overview, a simplified but good representation of some of the most commonly applied machine learning models within their respective categories can be seen illustrated in Figure 18

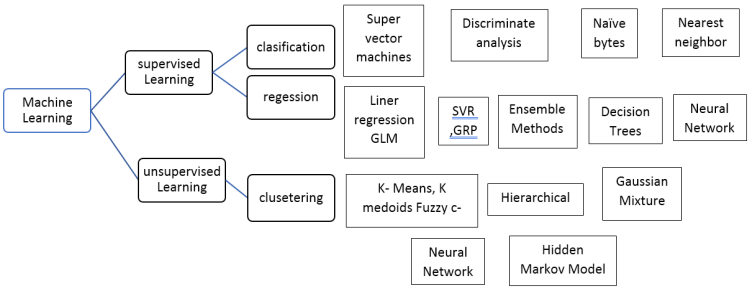


Figure 18: Simple overview of Supervised and Unsupervised Learning

In this project, the feature of interest is the oyster, which is photographed lying either on the exposed sand above the sea level (usually during low tides), or submerged beneath the water. The oyster is a white or light-colored blob feature (in Computer Vision, a blob (Binary Large OBject) is a region of connected pixels in an image, in which some properties are constant or approximately constant [what-when-how, 2020]). It comes in many various sizes, and can at most times be distinguished from the bottom substrate, which consists of sand, mud, pebbles etc. The classification of the oyster is challenging, due to various factors such as the refraction of objects being photographed through the water surface, the number of particles in the water which attenuates the signal, glare from sunlight on the water surface, waves and movement in the water, etc.

To begin with, classification will be done based on RGB (Red,Green,Blue)-based images of oysters, taken by a Go-Pro camera from above the water surface, to demonstrate the utility of various image analysis models. The Supervised Learning approach is attempted first. Considering that there does not exist any label data for the drone pictures, since those were captured by the author and his associates, these ground truths are drawn manually.

The model of choice in this category is the *Quadratic Maximum Likelihood Classification* algorithm. The reason for choosing this algorithm is that it is a relatively simple, but robust, algorithm based on well-established concepts within Machine Learning, such as Bayes' theorem and the Mahalanobis Distance, and it is well suited as a starting point for the classification purposes.

The focus of this algorithm is to produce a thematic map from the image data, where the digital numbers of the image represent the reflected or emitted Electromagnetic(EM)-radiation in different wavelength bands. Various physical objects/regions, such as light or dark oysters, sand in various colors, rocks etc. will be grouped into different classes as ground-truth beforehand by the author, using a polygon-drawing command named *Roipoly* in Matlab. Then the classifier is trained on the image and tested to see how well it classifies the physical objects in the image. A flowchart of how the algorithm operates can be seen in Figure 19

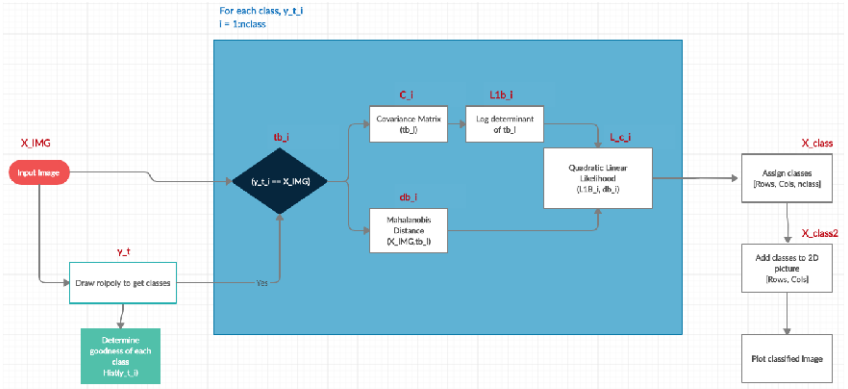


Figure 19: Flowchart of the Quadratic Maximum Likelihood algorithm

For further elaboration on how this algorithm works, the interested reader may refer to subsection 6.1 in the appendix.

To train the classifier, the training image (image No. 884) seen in the left side of Figure 20 was used, where eight distinct classes were assumed to be present: *Light Oysters (1), Dark Oysters (2), Light Rocks (3), Dark Rocks (4), Algae (5), Underwater Sand (6), Brown Sand (7) and finally Misc Objects (8, being the car tire)*.

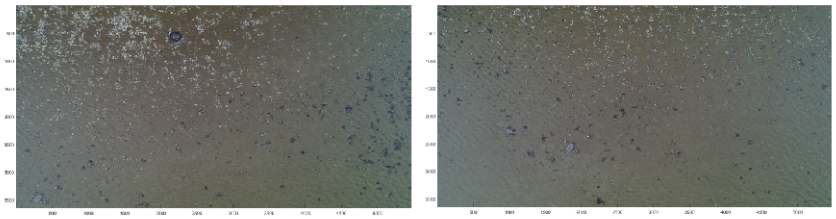


Figure 20: Image No. 884 for training (left), and test image 889 (right)

After training the algorithm on this image, the algorithm is tested on a different image (image No. 889), seen in the right side of Figure 20, which resulted in the following results seen in Figure 21.

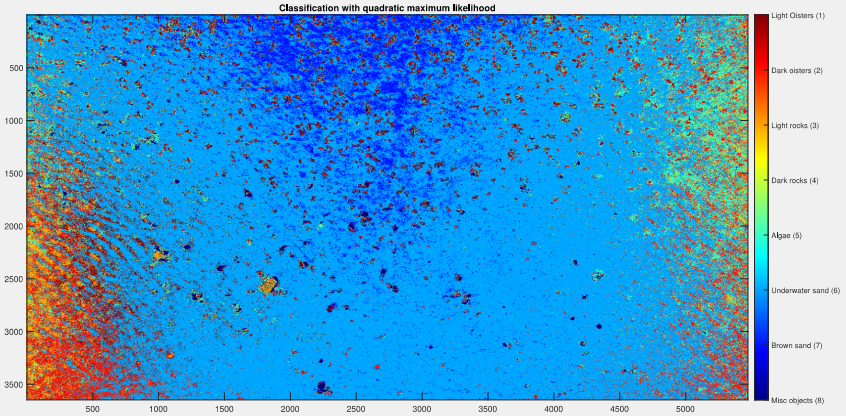
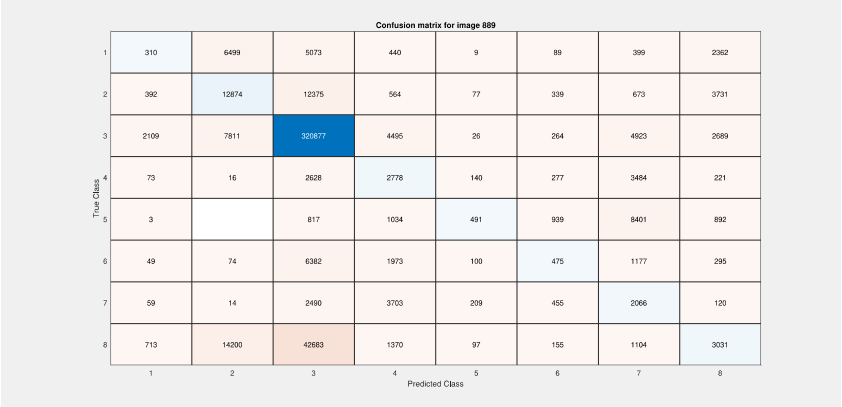


Figure 21: Test image 889 classification results

In addition, the confusion matrix for the classification results can be seen in Figure 22, showing the predicted class being and the true class predictions. The elements on the diagonal is when the predicted class equals the true class, which is the desired scenario.



classes beforehand in this way can potentially lead to some problems. Also, since the

chosen classes used for the targets are manually drawn by the user, this can also potentially lead to a rather large human-error factor as well. This begs the question if the Unsupervised Learning method may be a better approach to solving the oyster classification problem.

3.4 Classification with Unsupervised Learning

Now, the approach of Unsupervised Learning is attempted. Referring to Figure 18, there are many algorithms to choose from. However, one of the most well-known algorithms is that of the K-means clustering.

3.4.1 Classification with the K-means algorithm

In short, the K-means clustering groups observations into K initial clusters. Then, it calculates the euclidean distance of every observation to each cluster center (or centroid). Each observation is assigned to the cluster centroid that is closest to it. When the grouping is completed, the mean of all observations belonging to each cluster is computed, and the new cluster 'centroid' is moved to this mean. Then, each distance is calculated again for each observation, to see if some of the observations move to a different cluster after the cluster centroid has been moved. When the cluster centroids no longer move, the K-means algorithm is done and the data has been successfully clustered. The algorithm workflow is illustrated by the flowchart seen in Figure 23

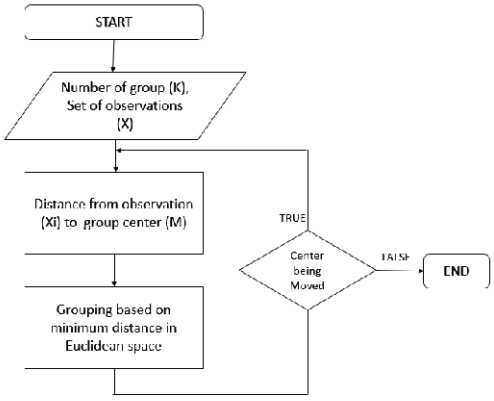


Figure 23: Flowchart illustrating the K-means workflow [Lai, 2017]

For further elaboration on how this algorithm works, the interested reader may refer to subsubsection 6.1.4 in the appendix.

The reason for using the K-means clustering algorithm is that it works well with data such as pictures of oysters, since oysters tend to either lie in great clusters

or lie sparsely and separated from each other on the ocean bottom. Additionally, they are often easily distinguishable from the bottom substrate. In this way, one can say that the oyster classification problem 'mimics' the way that the K-means algorithm is usually explained. Furthermore, it is always a good idea to use the K-means algorithm as a starting point, to get an overview of what the data and environment looks like, since it is also a quite simple and intuitive algorithm that is easy to understand and interpret. This makes it a good first step, that one can use to move into more advanced techniques later.

When running the K-means algorithm it can be helpful to let Matlab compute its estimate on the optimal K amount of clusters before clustering the data. This can be done with a command from Matlab's Machine Learning Toolbox named *evalclusters*, which for the image should return the optimal amount of clusters (optimal K). Running this algorithm returns an optimal K of only 4, instead of the initial 8 classes that was assumed by the author to be present in the training image. After the K-means algorithm is done, it returns a nice clustering, where most of the oysters has been clustered correctly into class No. 3, as can be seen in Figure 24

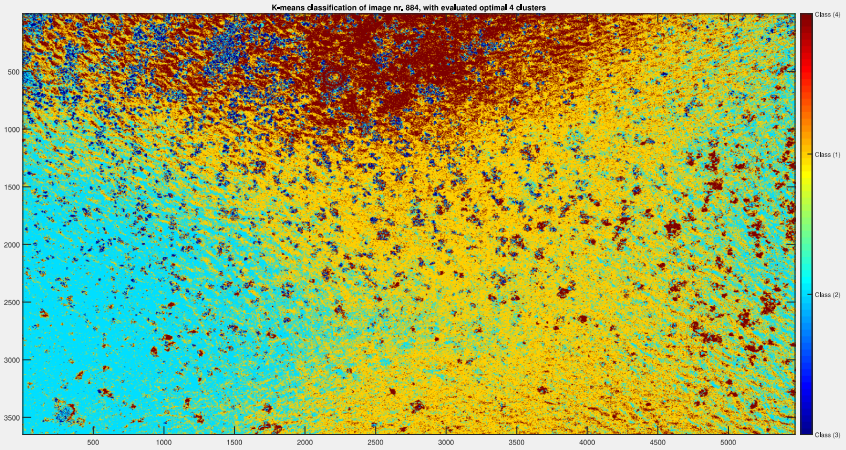


Figure 24: Results of the K-means classifcation

3.4.2 Classification with the Gaussian Mixture Model

A different approach to the Unsupervised Learning problem is to make use of a density-estimating approach instead, such as the Gaussian Mixture Model (GMM). The reason for choosing this algorithm, is that the GMM is more flexible than K-means, as it can account for clusters of different shapes, and informs the user of how probable the data is as well. It is also a lot less sensitive to data scaling, which is good considering the drone photographs environments of different depths, and features of different sizes and shapes.

The GMM model makes use of the multivariate normal distribution as a building block to create a more flexible distribution. The goal is to describe the probability distribution a given set of observations X has originated from. For every cluster, it is going to learn a Gaussian distribution that explains what that cluster looks like. In general, the GMM model is more flexible than K-means, as it can account for clusters of different shapes, and it also tells the user how probable the data is, making it easier to choose the K amount of clusters. In addition to this, it is also much less sensitive to data scaling. The workflow of the Gaussian Mixture Model, as implemented through Matlabs Machine Learning Toolbox, is visualized in Figure 25

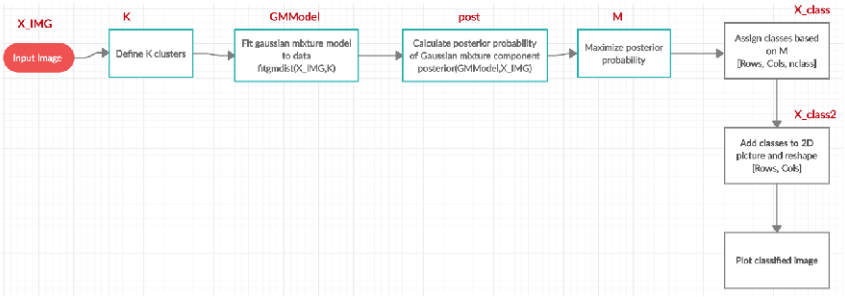


Figure 25: Flowchart of the Gaussian Mixture Model

For further elaboration on the workflow of the GMM model, the interested reader may refer to subsubsection 6.1.5.

Clustering the image with 8 clusters returns a classified image, that can be seen illustrated in Figure 26

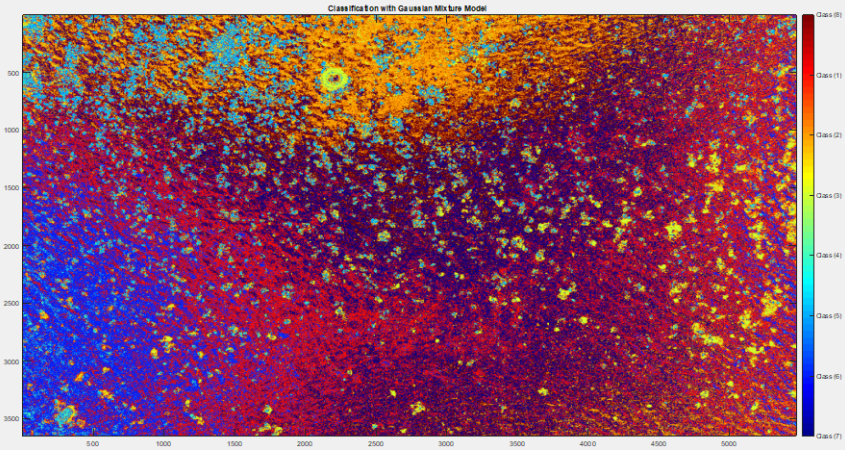


Figure 26: Full RGB-classification with a Gaussian Mixture model

When classifying with the GMM model for 8 classes, it seems to classify oysters into different kinds of oysters (yellow and turquoise). It also seems to differentiate between different types of sand. Very light and exposed sand is orange in colour, while light and submerged sand is blue, and the rest of the sand is either red or purple in color. Interestingly, it also classifies the tire as being one big, donut-shaped oyster. Hence, the algorithm seems to account for and segment the oysters of different colors and shapes. The above results proves that it is possible to quickly segment the image into different classes. However, if one wishes to get an estimate on the number of oysters being present in the image, which is one of the objectives of this project, the methods of Instance and Semantic Segmentation may be a better choice, which looks only at the desired class of interest and leaves the rest of the features as they are in the input image.

4 Oyster segmentation

In image analysis, an important concept is that of segmenting images, in order to single out particular features of interest. In relation to this project, this can be done to segment and count the number of oysters present in each image frame. There are two main types of segmentation methods to consider: the Semantic Segmentation and the Instance Segmentation.

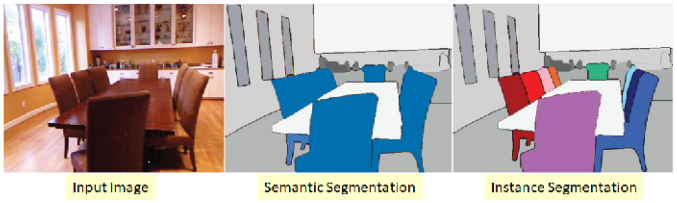


Figure 27: A comparison of Semantic and Instance Segmentation [Datascience.com, 2020]

Taking Figure 27 as an example, Semantic Segmentation is when every object belonging to a specific class (in this case, chairs) is labelled as the same classification. In other words, it treats multiple objects that may belong to the same class as a single entity.
Instance segmentation, however, takes the Semantic Segmentation algorithm a step further and counts every "instance" of the chair-category as a separate class, thus being able to distinguish between different "types" of chairs, for example.

For segmentation, the input image shown in Figure 28 (image No. 13) will be used:

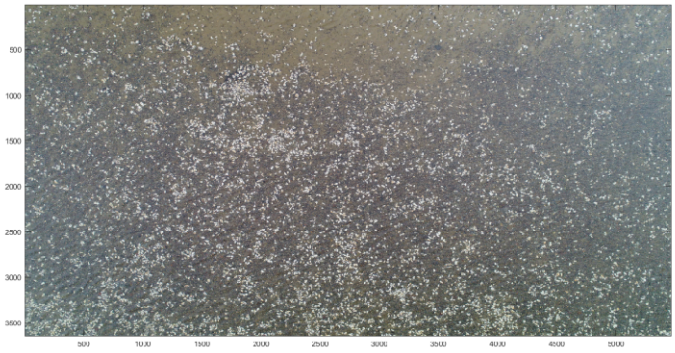


Figure 28: Input image No. 13

Before any segmentation can be done, one must apply the proper preprocessing steps. Firstly, the image must be smoothed with a 2-D Gaussian smoothing kernel. The reason for this is that each oyster has a lot of irregularities, discolourings etc. on its surface, that the algorithm will otherwise distinguish as being separate oysters. An illustration of the importance of this, as well as the difference between using different standard deviations (sigma) for the smoothing kernel, is shown in Figure 33.

4.1 Results with Semantic Segmentation

After the Gaussian smoothing has been done, it is fairly simple to perform Semantic Segmentation by using a threshold value. All pixels in the image have a pixel value between 0 and 255 in each band of the RGB-spectrum. Since all oysters in the image have a relatively high pixel value, i.e. oysters are usually very white or approximating a white colour, the threshold can be set quite high, such as 150. Then, a mask is made on the image which filters out all pixels above that threshold value, and color codes it. The workflow of the Semantic Segmentation is illustrated in the flowchart of Figure 29

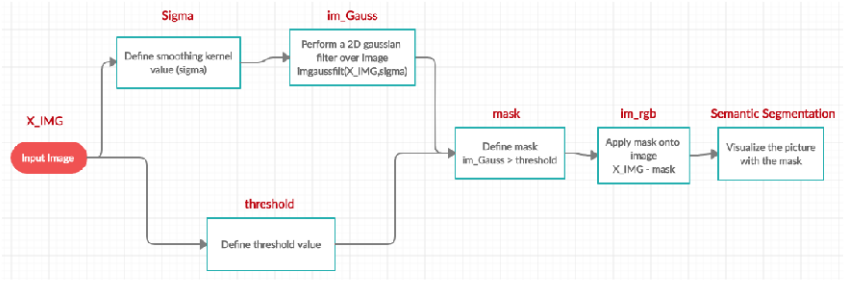


Figure 29: Flowchart showing the workflow of the Semantic Segmentation algorithm

Using the algorithm described above highlights most of the oysters, however some of the very brown or dark oysters are not segmented properly, because their intensity value is lower (130-140 for example). One could just lower the threshold value to include this, but unfortunately some of the sand is so light it also approximates these values of around 130-140. This means that if one lowers the threshold too much, most of the sandy bottom will potentially be classified as one, very large oyster, which is naturally undesirable. The results of Semantic Segmentation can be seen illustrated in Figure 30.

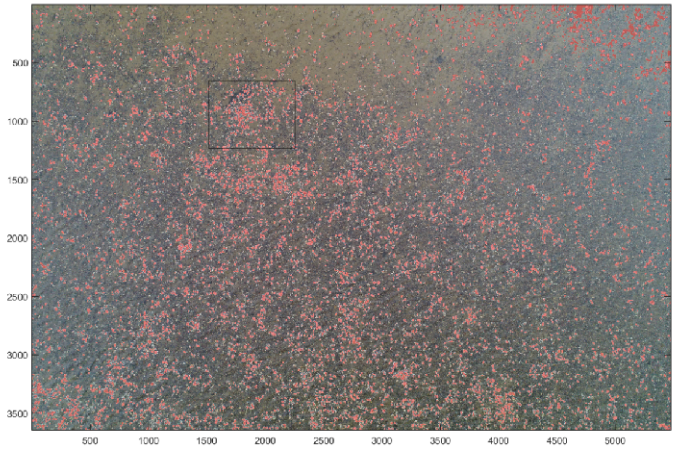


Figure 30: Results of the Semantic Segmentation

In Figure 30, most oysters have been color coded a red color, although it is a bit hard to see from the current distance. Hence, a zoomed in view is provided in Figure 31, covering the region bordered by the black rectangle seen in Figure 30.

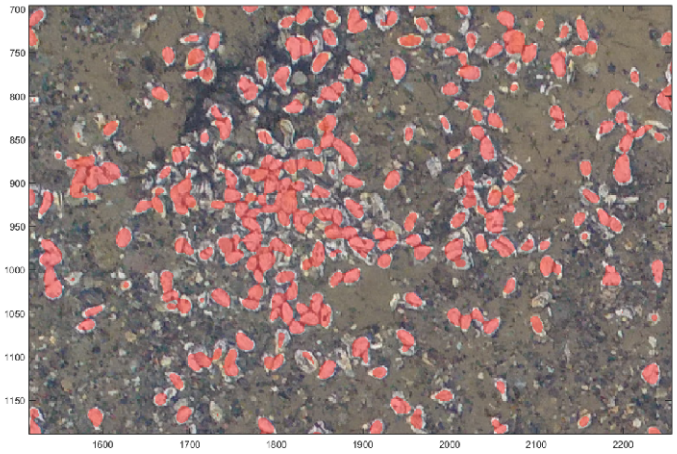


Figure 31: Results of the Semantic Segmentation, zoomed in to the black rectangle shown in Figure 30

It appears that not every oyster has been successfully segmented, although the majority has been. This, again, has to do with the choice of threshold. Lowering the threshold could include more oysters, but risk that the sandy bottom begin to be segmented as oysters as well.

4.2 Results with Instance segmentation

While Semantic Segmentation is a good first step to single out the oysters in an image, it is not sufficient if one wants to count the amount of oysters in the image as separate entities, i.e. count each "instance" of the oysters. This is where Instance Segmentation comes into play. Instance Segmentation also makes use of a Gaussian smoothing kernel at first, like Semantic Segmentation. Then, following the smoothing step, the *Watershed algorithm* is applied to the image. The watershed algorithm is, in itself, a type of segmentation in image analysis. It treats the image as if it was a topographic map, where the brightness levels of every pixel represents the height of the pixel within the map, and then it finds the lines that run along the top of the ridges. The concept is that if one imagines a bucket of water is poured out over the map, the water will then distribute itself and collect into different catchment basins of different heights, corresponding to each local minimum in the image. If water falls on the watershed ridge line separating two basins of equal height, where each basin is flooded to the level just before they merge with each other, it would be equally likely to collect into either of the two catchment basins. The algorithm then constructs a one-pixel thick dam (or ridge line) separating the two regions. The main reason for using the watershed algorithm in this context is that the oysters can often lie in clusters, which can make it difficult to distinguish them from one another. This ridge line is then what

is used to separate oysters from each other, by defining hard lines between oysters that lie very close together.

Following these steps, the image is ready for instance segmentation. Firstly, the regional maximum in connected components of pixels with a constant intensity value is found. Secondly, the centers and centroids of these points that are larger than our defined threshold are defined, based on the properties of the image regions. Lastly, the centers of the Radial Basis Function mode is used to retrieve the final Instance Segmentation of interest. The full workflow can also be seen illustrated in the flowchart in Figure 32

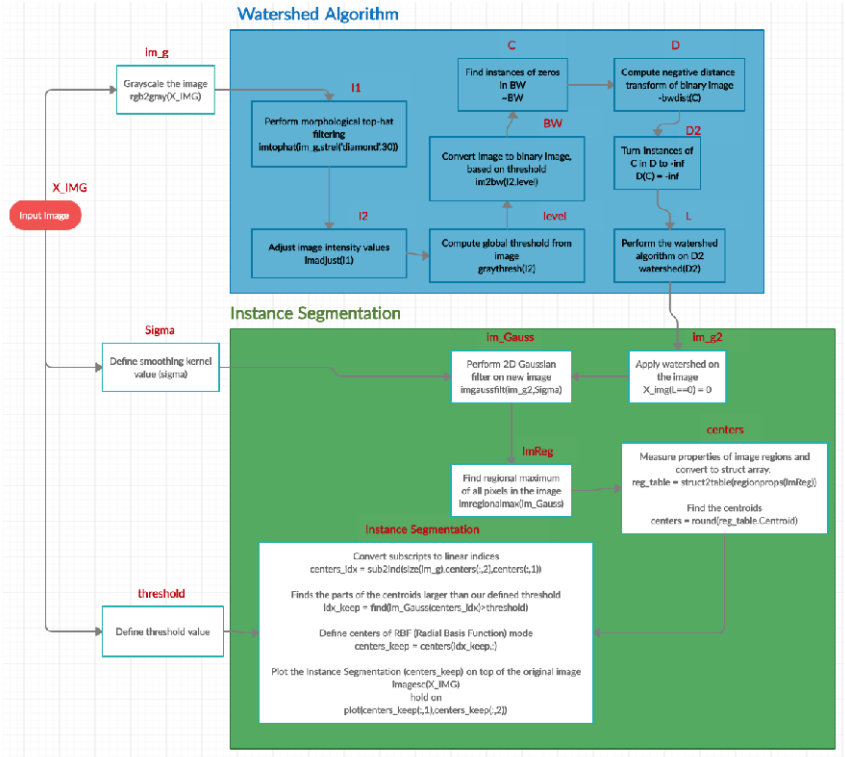


Figure 32: Flowchart illustrating the full workflow of the Watershed Algorithm and the Instance Segmentation

The result of this algorithm can be seen in Figure 33, using four different values of sigma, i.e. the standard deviation of the 2-D Gaussian smoothing kernel. Note that the results are vastly different depending on which value of sigma that is used.

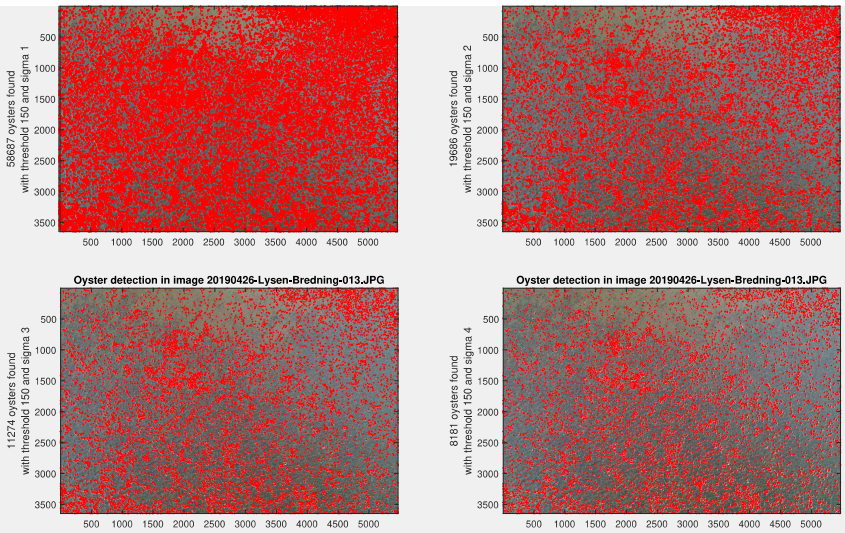


Figure 33: Results of the Instance Segmentation for four different values of sigma

It can be a little difficult to visually see what is happening in Figure 33, because the pictures are very large and contain very many pixels. To better demonstrate what is going on, the figure can be shown again, but zoomed in, to better show how the clusters of oysters are being segmented, as seen in Figure 34. Each 'red dot' corresponds to one oyster being found by the algorithm.

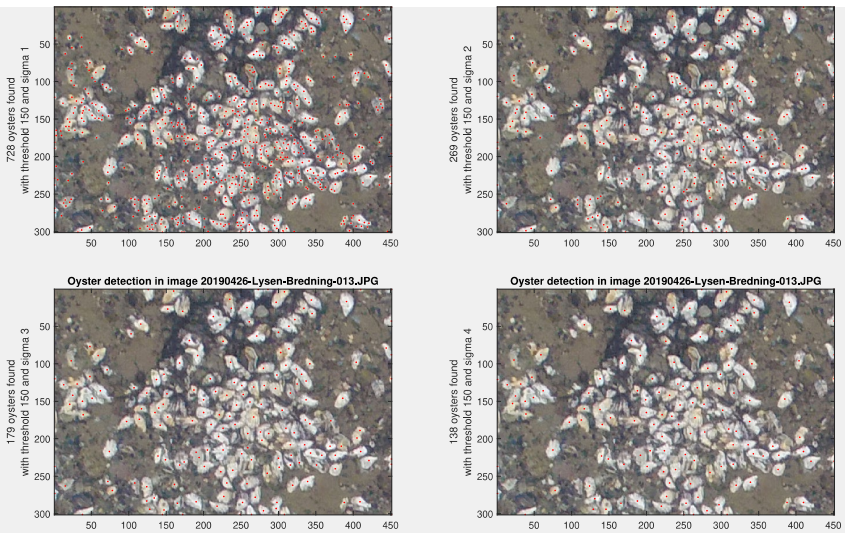


Figure 34: Results of the Instance Segmentation, zoomed in

As seen in Figure 34, using a sigma value of less than 4 leads to classifying the same oyster as multiple different oysters. This problem seems to disappear with a sigma of four. One downside of this, however, is that a sigma of four sometimes doesn't classify some of the darkest or most obscure oysters, but this is a better trade-off than counting the same oysters multiple times, in the authors opinion. Otherwise, a sigma of three can also be chosen, or a value between a sigma of three and four. Using a sigma higher than four leads to a lot of oysters not being detected at all.

To get a final classification success rate, the picture in Figure 35 will function as a ground truth. I.e. the amount of oysters in the following picture has manually been counted beforehand, and then the algorithm has been run, to see how many oysters are being classified correctly as distinct oysters.

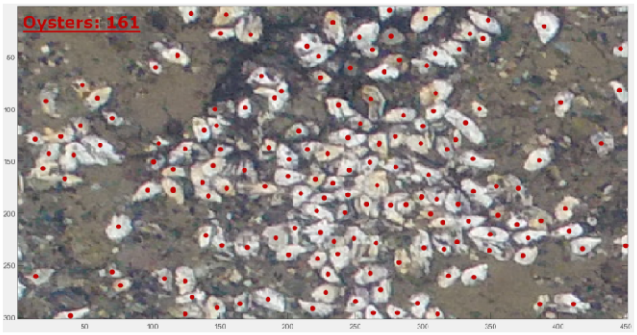


Figure 35: 161 manually counted oysters

In total, the author has counted approximately 161 oysters in Figure 35. The count is made by eyeballing the picture, and as such may be prone to human error. The picture here is the same as what was used as input in Figure 34. Comparing these figures with each other, it seems that the sigma that comes closest to the ground truth is in fact a mean value between three and four, i.e. a sigma value of 3.5 would produce the most accurate result for counting the oysters in the image, leading to a result of 157 oysters, as seen in Figure 36.

Figure 36: Oysters detected with a sigma of 3.5

Compared to the ground truth, this is a successful classification rate of 97.5%.

Now that the "ideal" environment of segmenting oysters has been covered, it raises the question: how will this classification rate look for oysters that are submerged underwater?

4.3 Instance Segmentation results with submerged oysters

For this test, the picture in Figure 37 will be used:

Figure 37: Input image 1

This region has the deepest water depth of the set of pictures being used in this analysis. As is apparent in the picture, oysters seem to be a bit darker here, likely due to the signal attenuation through the water. Furthermore, there seem to be a lot more oysters that are darker in colour. If detection of these brownish oysters are of interest, the threshold needs to be adjusted. Running the algorithm for this picture with a sigma value of 3.5, yields the following results for the full picture, and a zoomed-in picture, respectively, as seen in Figure 38 and Figure 39:

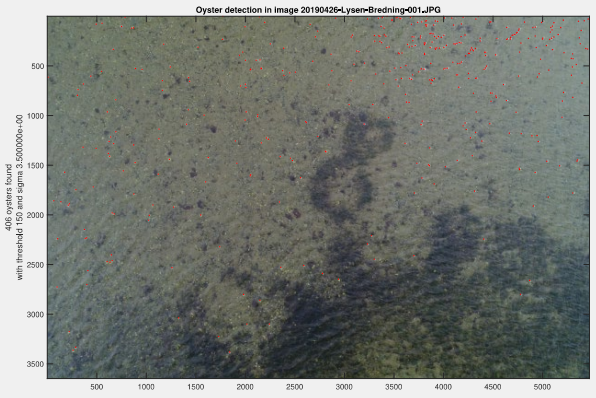


Figure 38: Instance segmentation results, 406 submerged oysters, threshold: 150

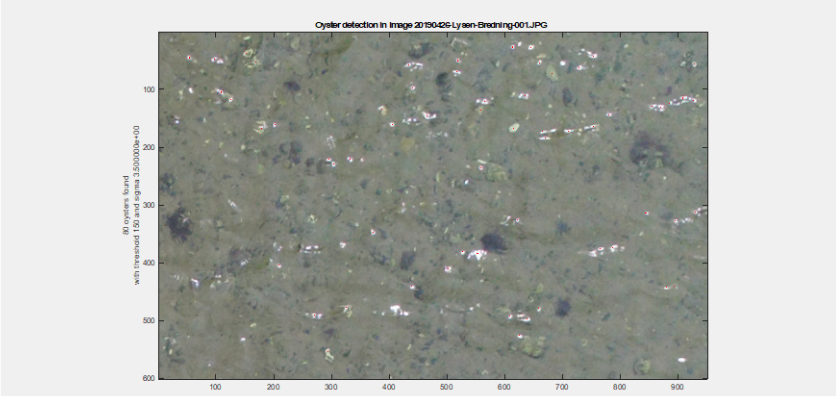


Figure 39: Instance segmentation results, 80 submerged oysters, threshold: 150

As can be seen, the algorithm manages to find 406 oysters in the full image here, and about 80 oysters in the zoomed in image. It is also apparent that the darkest oysters do not always get detected. Furthermore, it is not certain that all detected 'white blobs' are oysters, but rather just white rocks. But overall, the algorithm seems to work according to what it is told, i.e. *find all the distinct blobs with an intensity value above 145*, even under submerged conditions.

4.4 Instance Segmentation results with a MicaSense RedEdge-MX multi-spectral camera photo

For the last part of this chapter, the instance segmentation of oysters photographed by a MicaSense RedEdge-MX photo will be examined, and compared to a corresponding Go-Pro photo of the same area, taken at the same time.

The following 3 pictures will be compared.

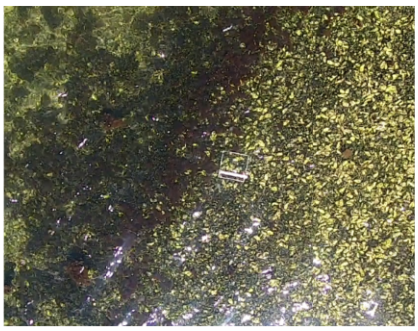


Figure 40: Go-Pro photo

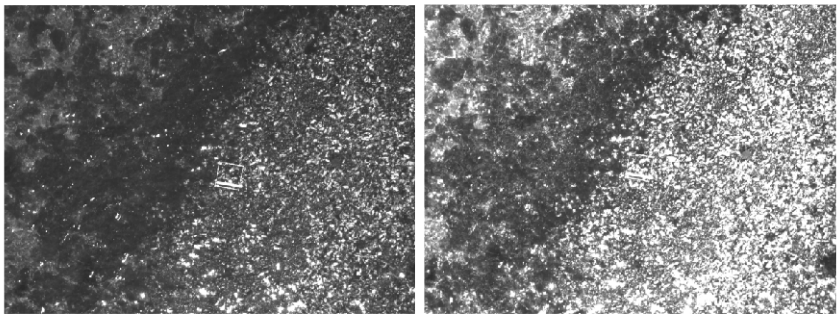


Figure 41: MicaSense blue band (left) MicaSense green band (right)

Firstly, Instance Segmentation is performed on the Go-Pro picture, as seen in Figure 42. Interestingly enough, after playing around with the sigma value, a very low sigma value is required here to get the best segmentation (1088 oysters with a sigma of 1.5 - 2). Increasing it beyond this leads to a lot of oysters not being classified at all, which also suggests that the choice of sigma is dependant on the depth of the oysters in the image (more refraction = less sigma). Additionally, the effects of glare from sunlight is very widespread in these pictures, leading to the glare from the surface being classified as oysters by the algorithm. However, this issue may be circumvented by installing a circular polarized filter on the camera.

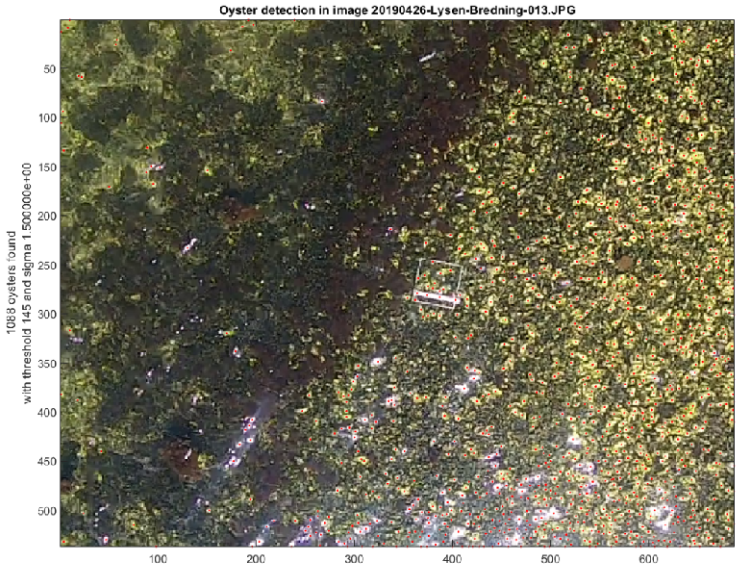


Figure 42: Results of the Go-Pro Instance Segmentation

Now, the same algorithm is run on the MicaSense photos, using the same

value of sigma:

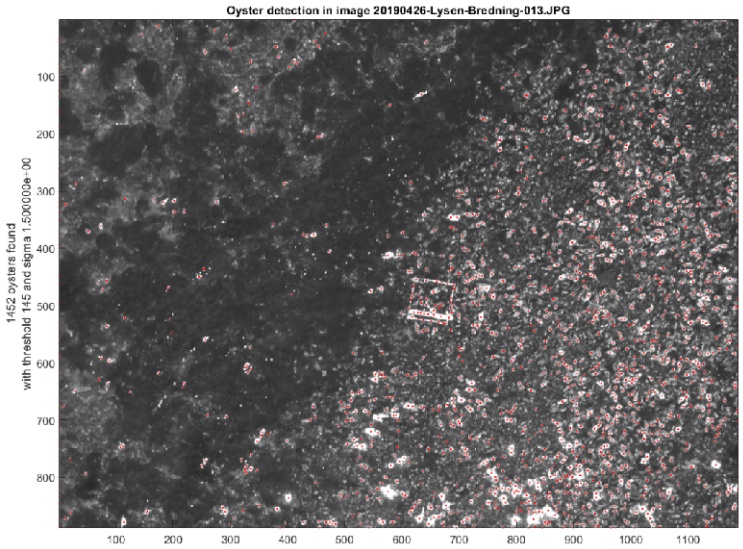


Figure 43: Results of the MicaSense Blue Band Instance Segmentation

It is seen that the blue band of the MicaSense camera seem to make the oysters "light up" in a different way, which is interesting. Also, apart from the distortion from sun glare, the rectangular box around the middle of the image causes a lot of distortion as well, as it has the same reflectivity values as oysters. Hence, the algorithm counts the box as multiple different oysters. It is also important to note that there are still many oysters or features that do not get segmented (the darker or more gray features on the bottom).

Now, the instance segmentation is performed on the green band image.

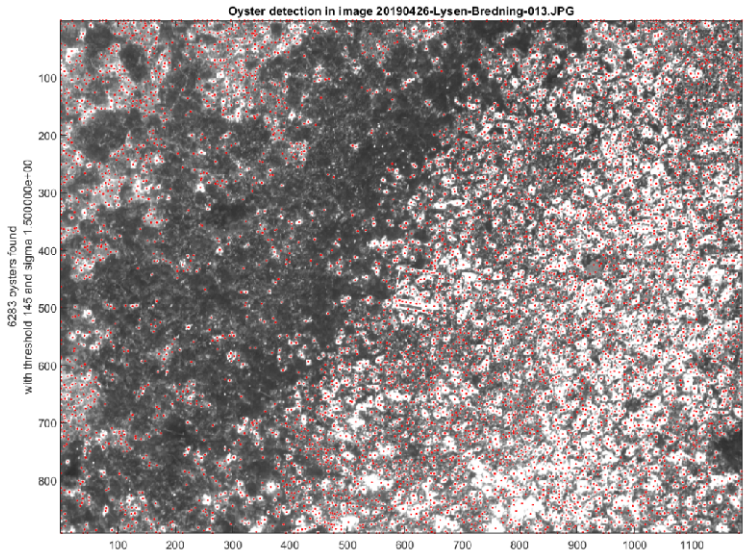


Figure 44: Results of the MicaSense Green Band Instance Segmentation

From Figure 44, it is clear that the reflectivity values of the oysters "explode" in the green band. Why this is the case is not known, but may confirm earlier speculations that oysters, being a living organism, has a much higher reflectivity value than other features on the ocean bottom. Unfortunately, they reflect "too much", and it is difficult to see what is what in the image, and also if it is only oysters reflecting this way, or if other features do so as well. But it looks like a promising method to make the features of interest stand out clearly in contrast to the background, even without any image analysis processing being made.

To further examine the above results, the sigma is increased slightly to a value of 3, to suppress some of the blobs a bit and distinguish them more from each other. Then, histogram equalization has been made on the image to further enhance the contrasts of the image (the background becomes more dark, the features of interest become more white). This leads to the following result in Figure 45.

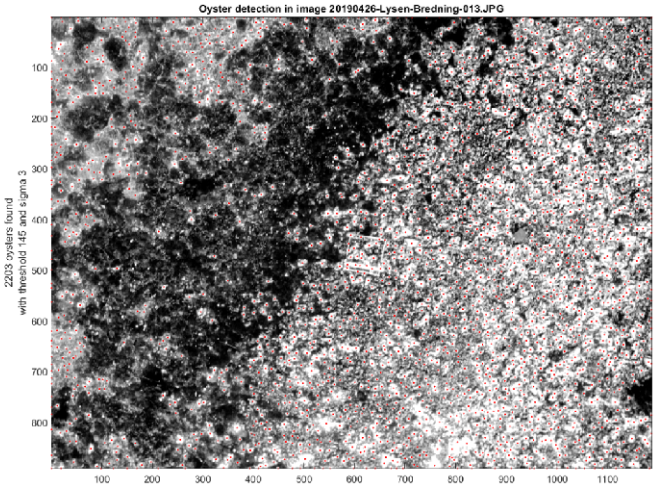


Figure 45: Results of the MicaSense Green Band Instance Segmentation, with histogram equalization and a sigma of 3

Based on what was seen in the Go-Pro image in Figure 42, we know that the right half of the image is the only section containing oysters, and the left part of the image contains a kind of algae. The above result in Figure 45 gives us a nicer overview of what is going on, and what is what in the image. It can also be seen that doing this histogram equalization lowers the amount of oysters being wrongly detected in the left section of Figure 45 (false positives). One could experiment with this further and try to lower the threshold to exclude more of them, but at the risk of not detecting oysters that actually are oysters (false negatives).

While the algorithm still does not perform perfectly, and counts features that are not oysters as being oysters, this still showcases that the MicaSense RedEdge-MX camera holds potential for photographing oysters beneath the water surface.

5 Discussion

One of the main obstacles of this project was to make an automated estimation of the sigma value, i.e. the Gaussian smoothing kernel, mentioned in the previous sections, so it could quickly iterate through all the pictures taken by the drone, and optimally segment them all at once. This method is also known as Automatic Scale Selection in literature (see [Lindeberg, 1996]). One approach to solving the task, and making an automated selection of the sigma value, was through scale-space representation methods, where the same image was to be modelled on multiple scales, successively suppressing fine-scale information until the optimal smoothing was achieved. The degree of smoothing would change from picture to

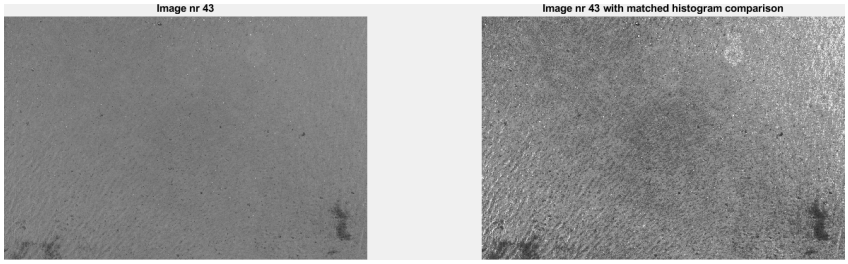


Figure 48: Histogram matching distorts the image

Of course this is an extreme example, as the histogram of a very homogeneous image is matched to the histogram of an image with many different features. But it shows that a more complex method is needed to make an automatic scale selection for all the pictures in the series.

Finally, it was decided to try and perform Instance Segmentation for a lot of randomly chosen images, with manually (drawn) ground truthing, to see if there is some trend in the optimal value of sigma. The results are seen illustrated below, where GroundTruth is the number of oysters in the figure estimated by the author, S3 count denotes the algorithm's estimate of oysters using Instance Segmentation with a sigma value of 3, and S4 is the algorithm's estimate using a sigma value of 4.

Image No.	GroundTruth count	S3 count	% diff	S4 count	% diff
15	230	260	**13.04%**	199	13.48%
27	23	204	786.96%	89	**286.95%**
64	78	104	33.33%	73	**6.41%**
109	113	134	18.6%	101	**10.62%**
141	685	706	**3.06%**	507	25.98%
89	42	9	**78.57%**	3	92.86%
199	71	170	136%	108	**50%**
4	89	46	**48%**	24	73%
32	53	59	**11%**	44	17%
126	63	5	**92%**	3	95.2%

In every case, a sigma of 3 or 4 was optimal, where the classifier with the best percentage difference is marked with bold. The % difference means the percent-wise difference from a 100% classification rate (i.e. a % diff equal to the value 0 would be ideal, while a high number is undesirable). The choice of images was based on having a large variation compared to each other; For example, one set of images had oysters that were covered with algae, while some featured dense clusters of oysters, and a third set featured oysters sparsely distributed over the ocean bottom. It can be seen that it performs very badly for a number of the images above. For example, the reason for the classifier performing very poorly in image No. 27, is

due to the sand bottom being very bright, causing the algorithm to confuse it with oysters, as the color intensity of the sand crosses the threshold for what colors an oyster may have (generally) across all the images. So at first sight, it seems like this algorithm performs quite poorly at segmenting images containing oysters. But that is only true for images that have very bright sand, or are at a greater depth, where the effects of refraction and signal attenuation are greater. For images where the oysters stand out clearly in contrast to the background or other features, it seems to be performing well at counting the oysters. Even more surprisingly, the algorithm proved to be especially good at estimating dense clusters of oysters (separating the oysters from each other), much more than what was expected. So the take-away lesson is that the algorithm may be used with a good classification rate, but only for pictures with an "ideal" environment. Unfortunately, that excludes a large part of the data set at this point. It is therefore up to the user to choose an appropriate sigma value for each individual image. Based on the results in the previous chapter, the author argues that going for a sigma value in the interval between three and four will, in most cases, lead to a fairly good segmentation.

5.1 Future work

5.1.1 Automatic Scale Selection

There are many directions in which to go to improve the current algorithm. One of the most important ones is to implement an automatic scale selection that works, as was elaborated upon earlier in this section. However, there are also other fields of image analysis methods that are yet unexplored, but may hold potential. One of the most promising (but also most difficult and time-consuming to implement) is that of a Deep Learning approach.

5.1.2 Deep Learning approach

Implementing a neural network, such as a Convolutional Neural Network, to segment the images may at first seem like a very desirable direction to go for. However, implementing a neural network (NN) for this task may prove to be very time consuming, and potentially expensive too. One of the largest obstacles in implementing a NN is the labelling of the data.

In Supervised Machine Learning, an objective is to learn a mapping $\mathbf{f}_\psi : \mathbf{x} \in^P \longrightarrow y$, from observations x to the target y, using a dataset of $\mathbf{D} = \mathbf{x}_i, \mathbf{y}_{ii=1,...N}$ of a finite size N (e.g. for image classification, translation etc). Unfortunately, due to the constraints of dimensionality, high-dimensional inputs (images) and complex models (deep learning) require that one uses very large data sets (millions of pairs $(\mathbf{x}, \mathbf{y})$). In other words, if you have labeled data, that means your data is marked up, or annotated, to show the *target*, which is the answer you want your machine learning model to predict. In general, data labeling can refer to tasks that include data tagging, annotation, classification, moderation, transcription, or processing. *From: [CloudFactory, 2020].*

Due to the process of labelling large amounts of data being a very tedious, repetitive and time-consuming process, most companies outsource the data-labelling process to third-party data-labeling services, such as *Cloudfactory* for example [CloudFactoryWeb, 2020]. In fact, Analyst firm Cognilytica recently stated that "*fully 80% of AI project time is spent on gathering, organizing, and labeling data.*" [CloudFactory, 2020].

To do this oneself, in practice, one would first need to have thousands of images all showing approximately the same thing (drone images for example, taken at a specific height over a specific kind of environment containing oysters), as was done for this project. Currently, no pre-trained neural networks exist that contain data for this specific kind of classification task. Secondly, one would have to label the data, which involves manually drawing polygons on every area of the image and color coding what is what in each image (allocate each region to a user-specified class), to tell the computer what to look for. And then do that for about a thousand images, each of which has dimensions of approximately 5000x3500 pixels. After this, it is possible to perform a number of data augmentation tasks to synthetically increase the label data set. This could possibly yield a label data set of a couple of thousand images, enough to train a small, primitive neural network.

To shorten the labelling process, it may also be possible to do it automatically somehow, such as using Semantic Segmentation or K-means clustering to partition each image into a class, and input that as a labelled image. The aforementioned approach was attempted for this project, but it did not yield any feasible results.

It is not certain whether a neural network like the above could outperform the already existing image analysis methods that have been used so far in this project, primarily due to the very small size of the network. But, if one managed to get a proper set of labelled data for training the neural network, there are two neural network architectures that comes to mind in regards to segmenting the oysters:

1. The DCAN (Deep Convolutional Auto-encoder Network) Architecture

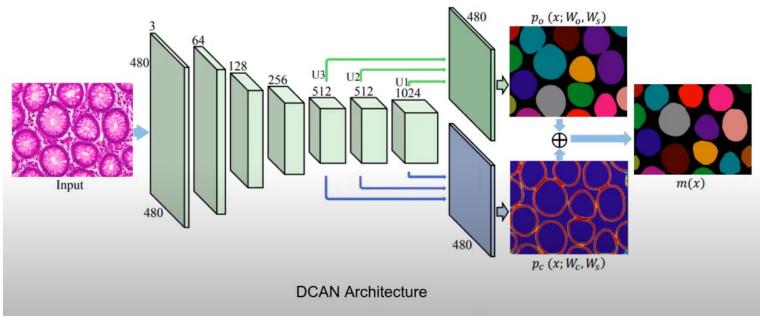


Figure 49: The DCAN Architecture

This architecture has 2 decoders, each based on a different problem. The above example is used for segmenting cells, which is a 'blob detection problem' similar to that of detecting oysters. In the above example, one decoder learns the segmentation of the cells, while the other decoder is learning the edge of the cell. Once these two segmentations are obtained, the edge can be subtracted from the cell, yielding a result (m(x)), in-between those two segmentations. This way, one will get an optimum count of the cell, while still detecting the most correct boundary of the cell.

2. The Mask R-CNN (Recurrent Convolutional Neural Network) Architecture

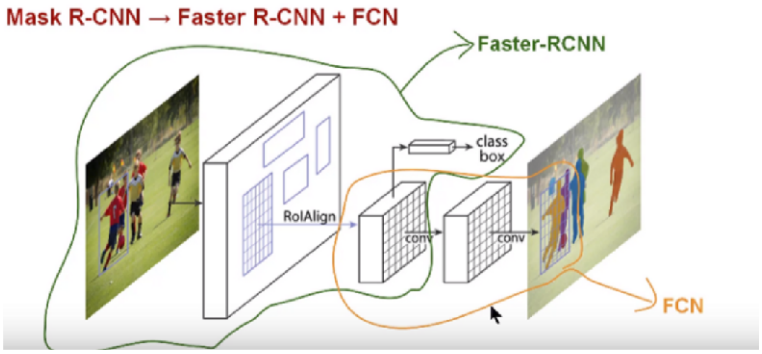


Figure 50: The Mask R-CNN Architecture, using Instance Segmentation

A normal RCNN network proposes a bunch of bounding boxes (or region proposals) in the image, and see if any of them correspond to an image, through a process called *selective search*. This selective search looks at the image, through windows of varying sizes, and for each image tries to cluster together adjacent pixels based on texture, color, or intensity to identify different objects. Then, R-CNN warps the region to a standard square size, and passes it through the Recurrent Convolutional Neural Network. In the final layer of the network, R-CNN adds a Support Vector Machine (SVM), to classify whether this is an object, and if that is the case, which object it is. As indicated above, the Mask R-CNN takes this a step further, by combining the Faster R-CNN (A modified R-CNN framework for better performance), plus a Fully Connected Network (FCN) [Science, 2017].This allows Instance Segmentation to come into play, by providing a pixel-wise mask for each object in the image. The reason for this is that some bounding boxes may otherwise overlap each other, and confuse the algorithm. The Instance Segmentation and bounding boxes are combined using a method called Roi Align. Using this approach, R-CNN obtains both the pixel-wise locations, as well as the coordinates of the bounding box of each object in the image, and ends up obtaining very precise segmentations.

Considering the fact that the Mask R-CNN approach is based on Instance Segmentation, which is used thoroughly in this project, it would be a natural next step to go to at the current stage. The above architecture is often used for self-driving cars, but in the example shown in Figure 50, it has been trained on Instance Segmentation of humans (which pixels are humans, and how many humans are there). Changing it to an Instance Segmentation of oysters instead would only require a small modification to the framework.

5.1.3 Customize the camera further

As mentioned in previous sections, one of the larger obstacles for the project was the effects of sun glare in the images. This can be avoided by installing a circular polarized filter on the camera. If this could be done for the MicaSense RedEdge-MX camera it would reduce a lot of the distortions seen, likely leading to a much better segmentation of the oysters when submerged.

6 Conclusion

The goal of the project has been two-fold. 1) To develop good Image Analysis-based algorithms to segment the oysters from the background in the drone images, both in the optical and multi-spectral regard, and give an estimate of the amount of oysters present in each image frame. And 2) to demonstrate whether or not a Micasense RedEdge-MX multi-spectral camera can out-compete a normal, optical camera, for photography and subsequent segmentation and estimation of oysters through the water surface.

Regarding the first objective, one Supervised Learning algorithm named the Quadratic Maximum Likelihood classification was developed for segmentation of oysters, as well as two Unsupervised Learning algorithms named the K-means clustering, and the Gaussian Mixture Model. These three methods were tested on various pictures and evaluated for their performance. Afterwards, a Semantic Segmentation and an Instance Segmentation algorithm was developed for the same purpose, which managed to give a good estimate of the number of oysters present in an image, as long as the conditions of the environment were not too difficult.

To pursue the second objective, a payload consisting of both a multi-spectral (Micasense RedEdge-MX) and an optical (Go-Pro) camera was built, and mounted on a DJI M-600 drone. This drone was then brought to Nykøbing Mors in northern Jutland, where several sets of images were captured above the oyster beds. The images were captured simultaneously in each camera, under the same conditions, allowing a good comparison between the cameras to be made. The images were subsequently analysed with the earlier mentioned Image Analysis methods, to get an estimate of the performance of both cameras. Interesting results were achieved as seen in Figure 44 and Figure 45, which showed that oysters are highlighted and stand out much more clearly in the green band of the MicaSense camera, especially after computing a histogram equalization of the image. Unfortunately, the metal quadrant present in the image also became much more bright, and was confused by the algorithm as being multiple oysters as well. However, ignoring the points directly on the quadrant and looking inside, the green band of the multi-spectral camera seem to detect more oysters than in the Go-Pro version of the image. It should be noted that the detection depends on the chosen value of sigma (standard deviation of the Gaussian smoothing kernel), and that this value was different between the two photos. Based on the results seen in Figure 45, it is clear, however, that a multi-spectral camera holds potential for mapping underwater environments like these, and may even out-compete a normal, optical camera, given the right amount of post processing, the right value of gain and exposure time (ISO) of the camera, and custom configurations (like installing a circular polarized filter).

Overall, the best approach to segmenting oysters would be to use the Gaussian Mixture Model (see Figure 26) to get a good overview of what is going on in the environment, followed by an Instance Segmentation with a user-set value of sigma, found by trial and error, to segment and count the oysters in each image.

Whole bibliography

Lindeberg, T. (1996). Scale-space: A framework for handling image structures at multiple scales. *Conference: Proc. CERN School of Computing*.

Reef2Reef. (2016). *Penetration depths at different wavelengths 2*. https://www.reef2reef.com/threads/confuse-about-intensity-and-spectrum.231543/page-2 (accessed: 26.03.2020)

Lai, D. (2017). *K-means clustering with visualization tool*. https://dunglai.github.io/2017/06/01/k-means/

Science, T. D. (2017). *Instance segmentation with mask r-cnn*. https://towardsdatascience.com/instance-segmentation-with-mask-r-cnn-6e5c4132030b

CloudFactory. (2020). *Data-labeling-guide*. https://www.cloudfactory.com/data-labeling-guide

CloudFactoryWeb. (2020). *Cloudfactory*. https://www.cloudfactory.com/

Datascience.com. (2020). *What is the difference between semantic segmentation, object detection and instance segmentation?* https://datascience.stackexchange.com/questions/52015/what-is-the-difference-between-semantic-segmentation-object-detection-and-insta (accessed: 16.06.2020)

NIELSEN-BOBBIT, J. (2020). *Eat the best oysters in the world in denmark*. https://www.scandinaviastandard.com/eat-the-best-oysters-in-the-world-in-denmark/

Stubgaard, K. (2020). *Dansk skaldyrcenter*. https://www.skaldyrcenter.aqua.dtu.dk/om-os

Support, M. (2020a). *Light sensors the basics dls and sunshine sensor*. https://support.micasense.com/hc/en-us/articles/115002782008-Light-Sensors-the-Basics-DLS-and-Sunshine-Sensor- (accessed: 23.03.2020)

Support, M. (2020b). *Micasense camera and dls sensor*. https://www.micasense.com/rededge-mx (accessed: 23.03.2020)

Support, M. (2020c). *Micasense camera spectral bands*. https://static1.squarespace.com/static/579a34a98419c24fcccb6be1/t/5bd37ba4e4966ba286463f42/1540842422179/REMX+Trifold.pdf (accessed: 26.03.2020)

Support, M. (2020d). *What does dls sensor do for my data*. https://support.micasense.com/hc/en-us/articles/219901327-What-does-RedEdge-s-Downwelling-Light-Sensor-do-for-my-data (accessed: 23.03.2020)

Timothy Bralower, D. B. (2020). *Em spectrum light intensity*. https://www.e-education.psu.edu/earth103/node/1006 (accessed: 26.03.2020)

what-when-how. (2020). *Blob analysis (introduction to video and image processing) part 1*. http://what-when-how.com/introduction-to-video-and-image-processing/blob-analysis-introduction-to-video-and-image-processing-part-1/

Appendix

6.1 Elaboration of Measures of distance, Bayes Theorem and the Mahalanobis distance

In machine learning, the concept of distance and similarity play a crucial role. For example, if a computer learning algorithm wants to differentiate between a picture of a cat and a dog, it will have to compare the image to what a cat and dog ought to look like. One of the most common ways to do this is with a distance metric, which is a function of two observations, x and y, such that its value is large when they are every dissimilar and small when they are very similar. In addition to this, it must obey some further rules, which will not be covered here. A common way to define this is with the magnitude of the difference of the observations, x - y, called a *norm* in vector space. The most common one of these is the well known euclidean norm, however, in this case, we shall make use of the *Mahalanobis Distance* instead for similarity measurement. Suppose we are given a covariance matrix Σ, estimated from a dataset. Then the *Mahalanobis Distance* can be defined as:

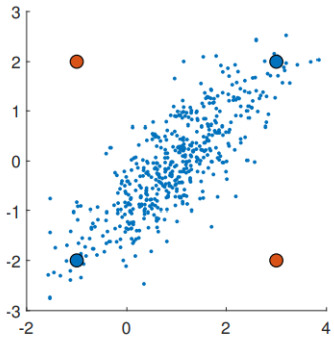


Figure 51: Simple 2D dataset to illustrate the Mahalanobis distance. The Mahalanobis distance is 13 between the red points but only 4.15 between the blue points.

$$d_M(x, y) = \sqrt{(x-y)^T \Sigma^{-1} (x-y)} \tag{2}$$

Notice that if $\Sigma = I$ the Mahalanobis Distance reduces to the Euclidean distance. Roughly said, what the Mahalanobis Distance takes into account is that the distance between two points should be lower when the points lie within the point cloud of the dataset. For instance in figure 51, the distance between the two red points according to the Mahalanobis distance is 13 but only 4.15 between the blue points, however in both cases the Euclidean distance is roughly 5.65. This indicates greater similarity for the two blue points.

6.1.1 Basic probability theory in Machine Learning

Even though most people are familiar with probabilities, they are often confused and intermingled with other concepts from statistics, and will therefore shortly be introduced anew here. We will consider binary propositions such as A, B or C. The probability of something will be denoted with P, so that the probability that A is true or that A and B are both true are written as

$$P(A) \qquad\qquad P(AB) \tag{3}$$

Furthermore, the probability that A is true given C is true, or that A and B are both true given C may be written as the *conditional probability*:

$$P(A|C) \qquad\qquad P(AB|C) \tag{4}$$

All probabilities are numbers between 0 and 1, with P(A) = 0 corresponding to A being false with 100% certainty and P(A) = 1 to A being true with 100% certainty. Finally, there are only 2 important rules in probability theory:
The sum rule: $P(A|C) + P(\bar{A}|C) = 1$
The product rule: $P(AB|C) = P(B|AC)P(A|C)$,
with $\bar{A}$ denoting *not* A, i.e. A is *not* true.

6.1.2 Bayes Rule

Bayes rule is generally written as

$$P(A_j|B) = \frac{P(B|A_j)P(A_j)}{\sum_i P(B|A_i)P(A_i)} \tag{5}$$

Which in our case would become

$$P(\omega_c|\mathbf{X}) = \frac{P(\mathbf{X}|\omega_c)P(\omega_c)}{\sum_{j=1}^{C} P(\mathbf{X}|\omega_j)P(\omega_j)} \tag{6}$$

The left side of this equation is thus the probability of finding class ω_c, given an observation $\mathbf{X}$.
Then, using Bayes' classifier, we would choose the maximum of $P(\omega_c|X)$, which is known as MAP (maximizing the posterior (or a posteriori) probability).

6.1.3 The quadratic maximum likelihood classifier, L

The classifier that will be used is known as the quadratix maximum likelihood classifier, $\iota_c(\mathbf{X})$:

$$\iota_c(\mathbf{X}) \sim lnP(\omega_c) - \frac{1}{2}ln|\mathbf{\Sigma}_c| - \frac{1}{2}(\mathbf{X} - \boldsymbol{\mu}_c)^T\mathbf{\Sigma}_c^{-1}(\mathbf{X} - \boldsymbol{\mu}_c) \tag{7}$$

where $P(\omega_c)$ denotes the prior (a priori) probability, and the last term, $-\frac{1}{2}(\mathbf{X} - \boldsymbol{\mu}_c)^T\mathbf{\Sigma}_c^{-1}(\mathbf{X} - \boldsymbol{\mu}_c)$ represents the Mahalanobis Distance.

6.1.4 K-means Algorithm Workflow

The K-means algorithm attempts to identify K clusters, using an euclidean distance-, and center-based approach. It takes an arbitrary data set and then clusters the data observations into K groups or clusters. A cluster in this context is thus

defined as a group of observations where the distance between observations within that cluster is smaller relative to the distance between observations outside of this cluster. The approach is as follows for a chosen K amount of clusters.

- First, each μ_k (center of the cluster) is initialized at a random location.
- Then each of the observations in the feature space is assigned to the nearest μ_k, and now belongs to this cluster. The region belonging to each cluster is often color coded.
- Then the location of each μ_k is updated to be the *mean* of the points assigned to it.
- The two previous steps is repeated until the location of μ_k doesn't change anymore.

While the K-means algorithm is a quite simple and efficient clustering algorithm, it is sensitive to both outliers and also by simple scaling of one coordinate while keeping the others fixed. In other words, it can only find round/circular clusters. For this reason, one usually standardizes the data before using it. Secondly, the cluster the K-means converges to may also depend on the initialization, making it important to consider how to initialize the data. The K-means algorithm can easily be implemented in Matlab using the function *kmeans*.

6.1.5 Gaussian Mixture Model workflow

The Gaussian Mixture Model is a good way to get around the limitations of the K-means algorithm. It uses a density-estimating approach, with the goal to describe the probability distribution that a given set of observations X has originated from. So for every cluster, it is going to learn a Gaussian distribution that explains what that cluster looks like. In general, the GMM model is more flexible than K-means, as it has the ability to account for clusters of different shapes, and it also tells the user how probable the data is, which will make it easier for the user to choose the K amount of clusters. In addition to this, it is also much less sensitive to data scaling.

The GMM model makes use of the multivariate normal distribution as a building block to create a more flexible distribution. At first, clustering the images with 8 clusters returns a classified image that can be seen in Figure 26. Each image only has 3 clusters, one for each channel of the RGB-spectrum.

1. Via Bayes' rule: $P(\omega_c|x_i) = K \cdot P(x_i|\omega_c)P(\omega_c)$ with $\frac{1}{K} = \sum_{j=1}^{C} P(x_i|\omega_j)P(\omega_j)$
2. For GMM, one has that $\mu_{ic} = P(\omega_c|x_i)$ with $\sum_{c=1}^{C} \mu_{ic} = 1$.
3. Then calculate $P(\omega_c) = \frac{1}{N}\sum_{i=1}^{N} \mu_{ic}$ being the mixing proportion of a class c, as well as $\mu_{ic} = P(\omega_c|x_i)$ where $\sum_{c=1}^{C} u_{ic} = 1$, $\mu_c = \frac{1}{NP(\omega_c}\sum_{i=1}^{N} \mu_{ic}x_i$ and $\sum_c = \frac{1}{NP(\omega_c}\sum_{i=1}^{N} \mu_{ic}(x_i - \mu_c)(x_i - \mu_c)^T$

4. μ_c and $\sum_c$ define $P(x_i|\omega_c)$ which, together with $P(\omega_c)$ via. Bayes' rule gives a new $\mu_{ic} = P(\omega_c|x_i)$, which in turn yields a new $P(\omega_c)$: Iterate

5. Usage of the Expectation Maximum (EM) algorithm:
a. Calculate $P(\omega_c), \mu_c, \sum_c$
b. Calculate $P(\omega_c|x_i)$ in Bayes' rule

The final GMM model, with use of the EM algorithm, is defined as

$$max \sum_{i=1}^{N} ln \sum_{c=1}^{C} P(\omega_c) P(X_i|\omega_c) \tag{8}$$

The GMM model can easily be implemented in Matlab using the *fitgmdist* and *posterior* algorithm from the Machine Learning Toolbox, as was also illustrated in Figure 25.

picture depending on the characteristics and environment of each picture, and be determined automatically by the algorithm. However, after thorough research, it proved to be a more difficult task to implement than first assumed, and finally it was deemed to be outside the scope of this project.

Another method that was attempted was that of histogram matching. By matching each new image to the histogram of the reference image (here, image No. 15, as illustrated in Figure 46), and then smoothing the new image according to the sigma value that was the optimal choice for the reference image, this sigma value should stay the same value, as it smoothes according to one particular histogram, and thus it should also be the optimal choice for the new, matched images. However, this proved not to be the case at all, and in fact, the matched histograms would introduce distortion into the image, in the shape of white blobs, as is seen illustrated in Figure 47 and in Figure 48, especially in the upper-right section of the figure. These white blobs might be mistaken by the algorithm as oysters, which would cause more damage than good.

Figure 46: Reference image (image No. 15)

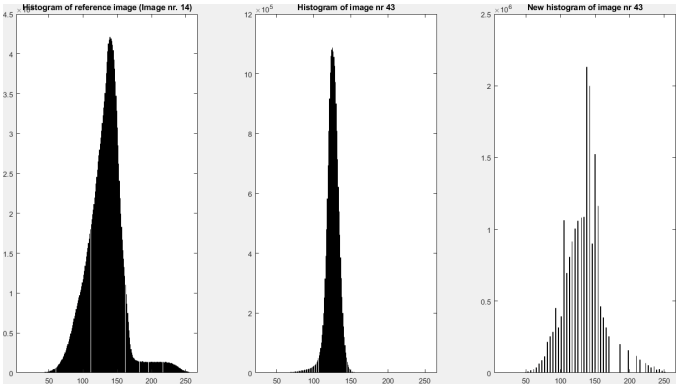


Figure 47: Histogram of the reference image (left), and before and after histogram of the image in question (middle and right)

Appendix 3.2: Student report: Methods to estimate oyster coverage by using different algorithms

May 4, 2021
DTU Space

Oyster Project

DTU Space
National Space Institute

Contents

1 Introduction

The aim of this part of the project has been primarily testing different methods to estimate the oyster coverage in the images captured from UAVs. Several different algorithms were tested to segment oyster beds in the images. It was tested whether the methods optimal for detection of eelgrass will also be optimal for oysters.

1.1 Previous Work

This work is a continuation of a report *Automatic detection and estimation of submerged oysters from drone images* by Jonathan Gundorph. Whereas his work focused on the possibilities of very accurate segmentation of every single oyster in the image, this work has been directed towards estimating the area covered by oysters.

1.2 Supervised vs. Unsupervised learning

There is no doubt that using supervised learning, in particular Convolutional Neural Networks (CNN) will be an interesting direction for the further research and algorithm development. However, taking into consideration, the limited time and limited amount of data, it was decided to focus mostly on image pre-processing and testing unsupervised learning algorithms (e.g Kmeans, Otsu) to identify difficulties and influence of flight altitude, weather and light conditions.

1.3 Multispectral Camera

It has been tested that multispectral camera is not an optimal sensor for monitoring underwater environment. Longer wavelengths cannot penetrate the water surface. Figures 1, 2 and 3 shows the comparison of the same image of an eelgrass bed taken by Micasense RedEdge in RGB, Band 4 and Band 5.

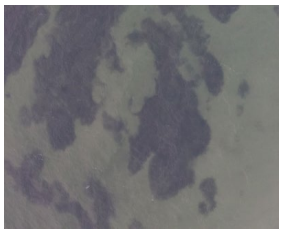


Figure 1: Original Image, Brondby, 40 m

Figure 2: Band 4

Figure 3: Band 5

1.4 Difficulties with identifying oysters

Segmenting oysters in images captured from above the water surface poses certain challenges. The identified diffuclties include:

- Sea surface
- Small contrast between background and oysters
- Different colors of oysters
- oysters covered by sand and algaes

1.5 Datasets

All the conclusions are based on a dataset collected in Lysen Bredning. Images were captured by DJI Phantom flying on altitude of 10 and 15 meters. Images chosen for the tests were thought to represent different environmental conditions. Figures 4 and 5 show the sample images.

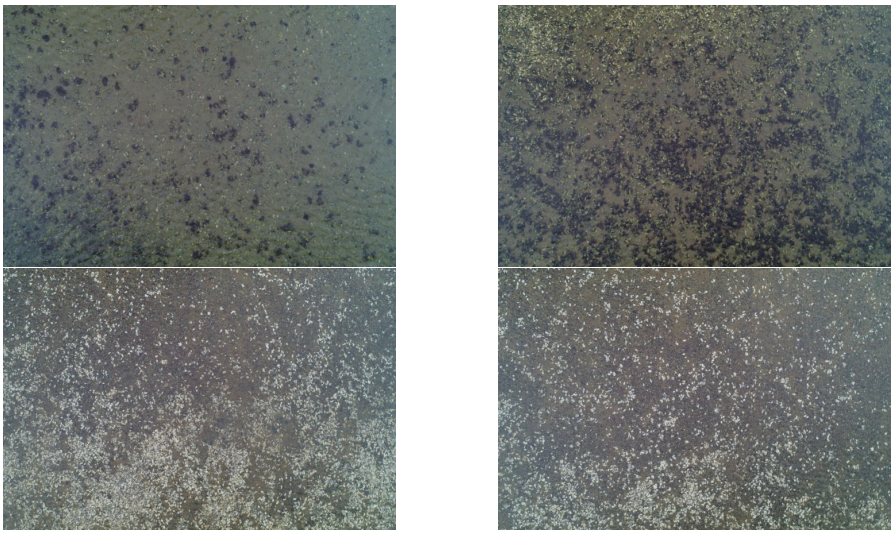


Figure 4: Sample Images captured at 10 m flight altitude

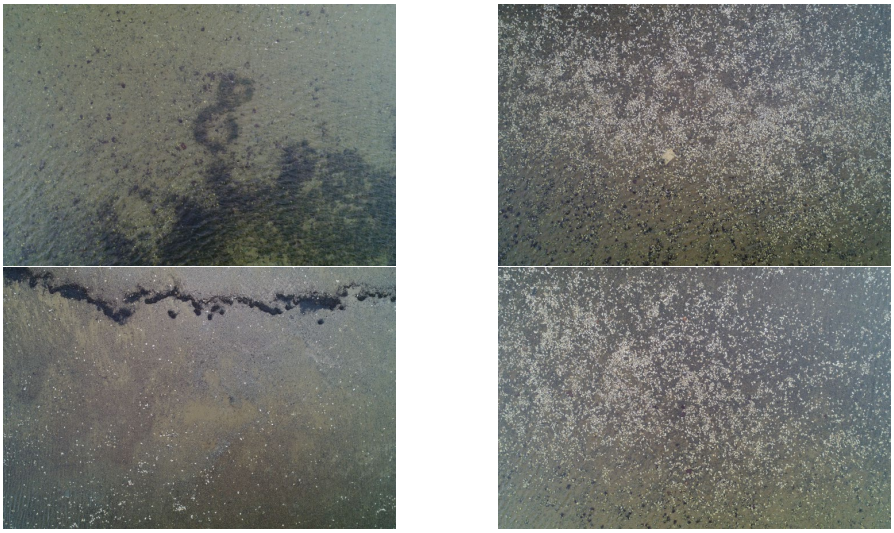


Figure 5: Sample images captured at 15 m flight altitude

2 Image Pre-Processing

It has been noticed that a very important part of working with images is pre-processing and trying to erase the effect of light conditions and uneven water surface before segmentation. It was thought that improving the contrast between oyster bed and its background as well as sharpening the edges should improve the results.

2.1 Filtration in Frequency domain

Three main types of filters in frequency domain have been tested in the project:

- Low pass filter
- high pass filter
- notch filters

Frequency spectrum of an image can provide interesting information about the noise in the images. It was hoped that the small roughness of the water surface in presence of wind will be visible in the spectrum as a repetitive spectral noise, which can be erased by notch filters. The assumption proved to be incorrect, which is shown in Figure 6 The use of high pass filter stops the lower frequencies and enhances the noise in the images, therefore it was not used in the further analysis. Low pass filter proves to improve visibility underneath the water surface and significantly enhances contrast, which can be seen in Figure 7. On the other hand, as individual oysters are rather small, using lowpass filter might blur the contours and decrease the segmentation.

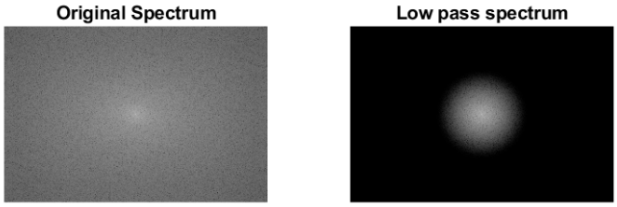


Figure 6: Image Spectrum

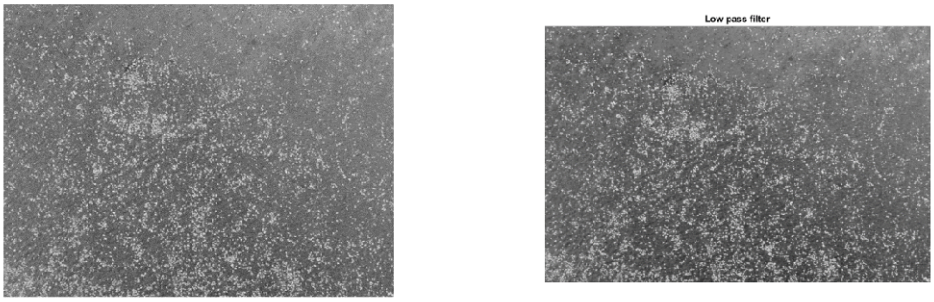


Figure 7: Left: Original Image, Right: Filtered Image

2.2 Dehazing Algorithm

Dehazing algortihms are typically used to reduce atmospheric haze in an RGB image. It was noticed that the water surface is similar in its properties to the atmospheric haze as it reduces the contrast, blurs the edges and makes the image seem 'foggy'. The algorithm used in this project is based on a dark channel

prior, which is based on the observation that unhazy images of outdoor scenes usually contain some pixels that have low signal in one or more color channels.

Figures 8 and 9 show the original and dehazed image. The improvement is very well visible in Figure 9.

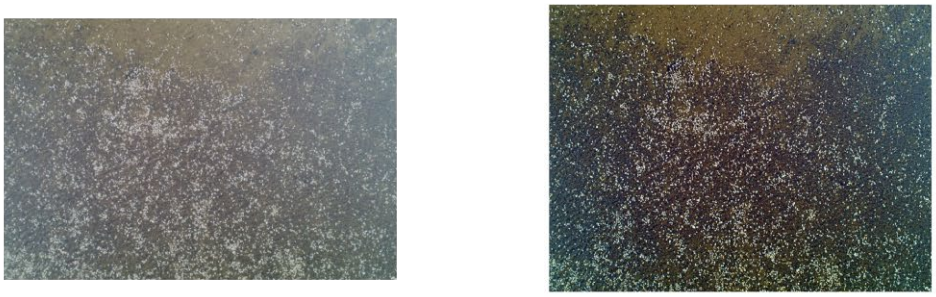


Figure 8: Left: Original Image, Right: Filtered Image

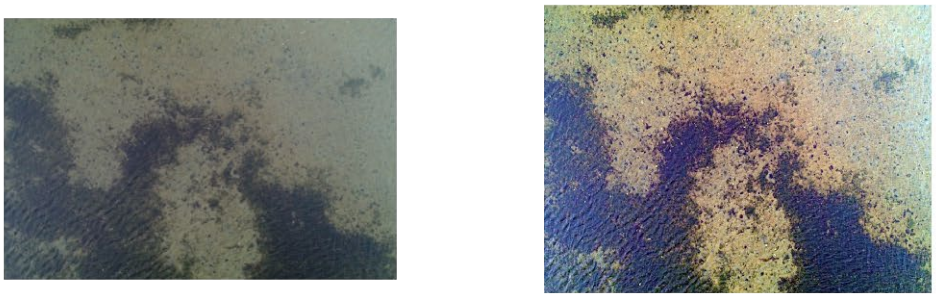


Figure 9: Left: Original Image, Right: Filtered Image

3 Methodology

White oysters are much easier to recognize in the image than darker ones. Unfortunately, the developed algorithms are able only to recognize and segment white oysters coverage.

3.1 Color Spaces

Oysters has proven to be the most visible in YCbCr color space. Working on the value of luminance instead of intensities of gray increaes segmentation results.

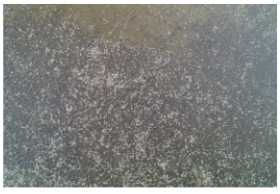


Figure 10: Original Image, 15 m

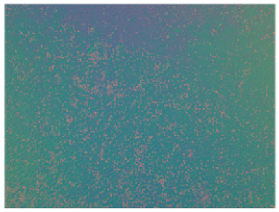


Figure 11: Image in YCbCr color space

Figure 12: Luminance band

3.2 Canny Edge detection

Canny edge detection with further morphological operations has proven to be a promising method. It does not segment all the oysters correctly, but it proves that edge and line detection algorithms might be worth further investigation.

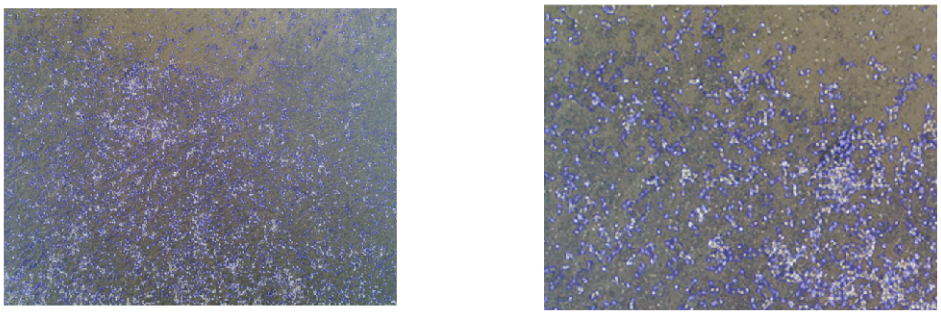


Figure 13: Left: Full size image, Right: Zoom in

3.3 Otsu Binarization

Otsu algorithm is a simple algorithm which automatically finds a binarization threshold based on the histogram of the intensity of gray. It is very simple to use and it works well for images where white mussels are significantly outstanding in the image. It was noticed that the algorithm fails in presence of algaes. Figure 14 shows the sample results using Otsu binarization algorithm.

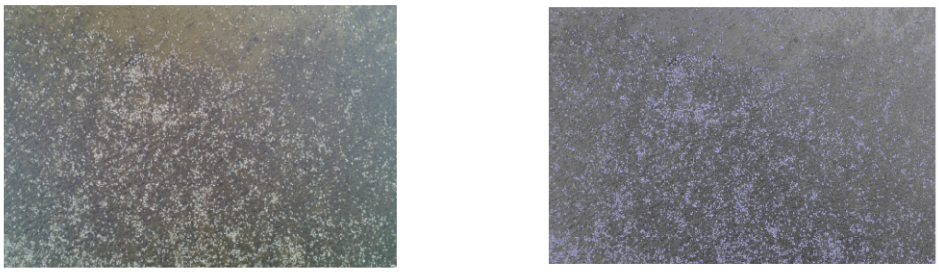


Figure 14: Left: Original Image, Right: Binarized Image

3.4 Segmentation with K-means algorithm

Kmeans algorithm is commonly used for image segmentation. It is an unsupervised learning technique which assigns the pixels to the cluster with the nearest mean. An advantage of using Kmeans is a possibility to choose the number of clusters (therefore it can be used when other objects are captured in the image) and ability to combine different channels (e.g RGB with texture and saturation). Kmeans algorithm has already been used in Jonathan's report, however it was thought that an attempt to minimize the number of clusters and have more accurate results is needed.

Kmeans algorithm has been tested with both 2 and 3 clusters and the sample results are shown in Figure 15.

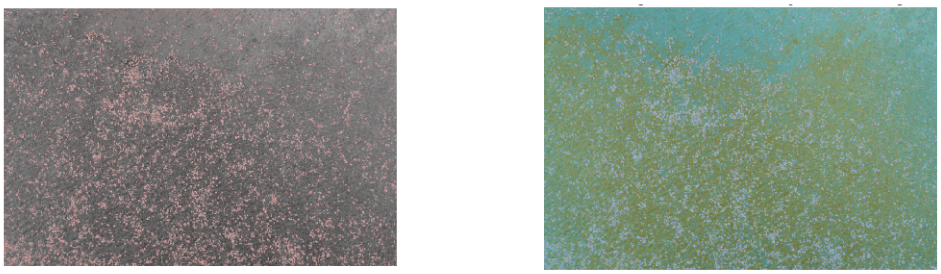


Figure 15: Left: 2 clusters, Right: 3 clusters

4 Results

4.1 Influence of the flight altitude

In general, the higher the flight altitude, the less details are visible in the images, which might decrease the segmentation results. Available datasets did not provide enough information to draw additional conclusions, however based on the eelgrasss project, it is thought that single oysters might be even harder to recognize if the flight altitude is higher than 40 meters.

4.2 Algaes

Algaes appear in the images either as a separate, dense object **??** or as smaller clumps partly covering the oysters **??**. None of the algorithms have been successful in segmenting an algaes bed and oysters beds in the same. Even though the dehazing algorithm increases the visibility of algaes quite significantly (8), they still cannot be successfully distinguished from dark oysters. Figures 16 and 17 show Kmeans segmentation results of luminance after dehazing the images. Oysters and light sand end up in the same cluster, therefore it was thought that the solution might be a bigger number of clusters.

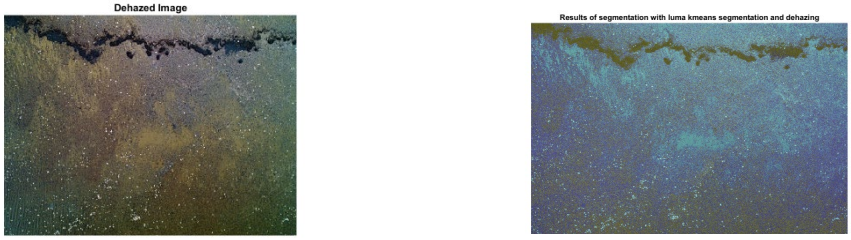


Figure 16: Left: Dehazed Image, 15 m, Right: Segmented Image

Figure 17: Left: Dehazed Image, 15 m, Right: Segmented Image

Algorithm has worked very well with clustering white oysters partly covered by algaes. It has to be taken into account though that the calculated coverage area will be smaller as only parts of the oysters are visible. However, it seems that it is not possible to estimate the coverage of oysters and algaes and oysters at the same time as are covered by algaes has not been segmented accurately. Figure 18 shows the Kmeans segmentation results of luminance on the dehazed image.

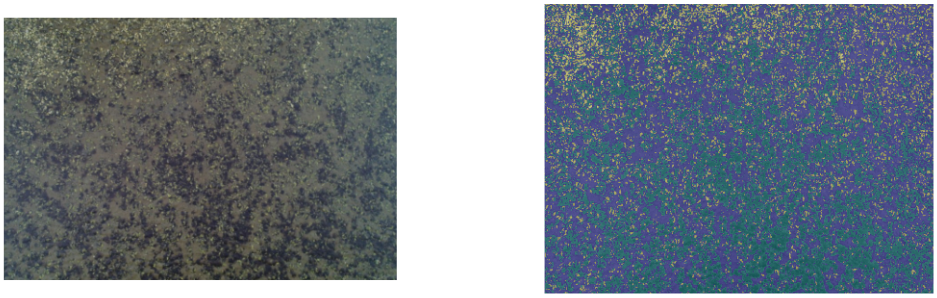


Figure 18: Left: Dehazed Image, 15 m, Right: Segmented Image

4.3 Dense oysters' beds

Dense beds are being segmented more accurately than single oysters as the *imsegkmeans* algorithm in Matlab takes into account also localization of the pixels. Both Otsu binarization and Kmeans segmentation of luminance after dehazing seem to successfully cluster oysters in Figures 19 and 20.

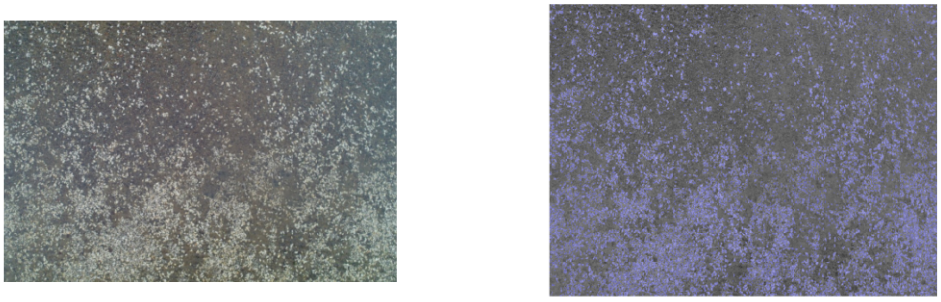


Figure 19: Left: Original Image, 10 m, Right: Otsu binarization

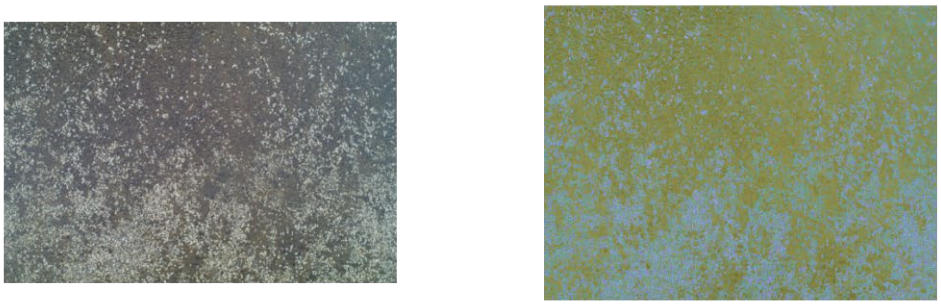


Figure 20: Left: Original Image, 10 m, Right: Kmeans segmentation

4.4 Dark sand

Figure 21 shows an interesting result of Kmeans segmentation after dehazing in presence of the dark sand. Light sand and oysters have been segmented to the same cluster.

Figure 21: Left: Original Image, 15 m, Right: Kmeans segmentation

5 Conclusions and Recommendations

Segmentation of oyster beds has proven to be challenging. Tested algorithms and pre-processing techniques gives an indication what might be the direction of further research and which algorithms will not yield satisfactory results.

Due to the fact that segmentation has proven to be dependent on the environmental changes (light and weather conditions, surrounding vegetation) it is recommended to capture similar images in the same are during different seasons.

Possible further pre-processing should focus on neutralizing different lighting conditions. It is thought that Retinex algorithm might prove to be useful.

As for further development of segmentation algorithm, it is recommended to look into supervised learning. It requires much more work to create and annotate a representative dataset, but we could observe that certain characteristics of oysters (e.g similar values in saturation) are the same in different images and might be used as an indication of oysters' presence.

Moreover, good multispectral dataset needs to be acquired to test the possibilities of using e.g. NDVI to segment oysters on the shore. It is thought that oysters being living organisms, might be recognizable on the NDVI maps.

Appendix 4.1: Genetic tables

Glossary of terms used throughout the report, including the acronyms used.

Term	Acronym	Definition
Locus/Loci		A general term to refer to a location in the DNA
Single Nucleotide Polymorphism	SNP	A base pair (C, T, G, A) in the DNA where there is variation within individuals of a population
Linkage Disequilibrium	LD	Non-random association of two loci within the DNA, located or not close-by
Wright's F_{ST}	F_{ST}	Levels of genetic differentiation between two populations. E.g. F_{ST}=0, no differentiation; F_{ST} =1, two distinct isolated populations
Effective population size	N_e	The number of individuals reproducing/genetically contributing to the next generation
Wright's F_{IS}	F_{IS}	Inbreeding coefficient of individuals in relation to the subpopulation
Nei's G_{ST}	G_{ST}	Measure of differentiation between population that is used to calculate relative migration between them
Observed Heterozygosity	H_e	Proportion of heterozygous loci observed in the DNA of an individual, averaged per population
Allele richness	AR	The average number of variants per marker

Table 1: Size measurements (length) per location for the corresponding age/size class.

Location	Size class	Min. length (mm)	Max. length (mm)
Agger Tange	0	9.47	18.44
	1	50.6	74
	3	120.74	168
Branden	1	20	60.12
	3	130.01	195
Struer	0	8.19	38.8
	2	100.23	102
	3	121.27	143.02
Nykøbing Bugt	0	23.72	35.35
	1	76.58	89.61
	3	130	185
Sallingsund	0	14.26	46.63
	1	66.75	118.95
	3	130.92	165
Dråby	0	14.95	36.96
	1	40.39	68.54
	3	140.29	189
Nissum	3	90.15	113.37
Løgstør	2-3	94.86	139.39
	4	142.2	153.99
Isefjord	NA	NA	NA
Lysen offshore	1	70	95
	2	105	145
	3	150	195
Hals	1	50	95
	2-3	100	135
Lysen	0	30	45
	1	50	65
	1-2	70	85
	2-3	90	115
	4	120	165
Netherlands	0	46.56	58.99
	1	60.51	98.08
	2	102.55	135.59
Sweden	1	60	90
Wadden Sea	0	21.37	35.99
	0-1	36.82	49.92
	1-2	70.27	89.58
Norway	1	85	99
	2	100	172

Table 2: results of the relatedness analysis on the real data and simulated data for each sample location used for the study. Acronyms: Standard Deviation (SD); Standard Error (SE); Confidence Interval (CI).

Location	Type of data	No. dryad (Real)	Mean Relatedness	SD	Median	SE	Mean Relatedness CI	Mean CI 25%	Mean CI 97.5%
Agger Tange	Real	5253	-0.0104	0.1408	-0.0206	0.0019	-0.0019	-0.2704	0.2665
Branden	Real	5050	-0.0103	0.1365	-0.0175	0.0019	-0.0021	-0.2615	0.2572
Dråby	Real	2278	-0.0155	0.1327	-0.0192	0.0028	-0.0073	-0.2658	0.2511
Hals	Real	1891	-0.0171	0.1384	-0.0239	0.0032	-0.0095	-0.2719	0.2530
Isefjord	Real	378	-0.0393	0.1407	-0.0492	0.0072	-0.0305	-0.3023	0.2413
Sallingsund	Real	2628	-0.0145	0.1415	-0.0233	0.0028	-0.0064	-0.2749	0.2621
Løgstør	Real	780	-0.0270	0.1441	-0.0364	0.0052	-0.0202	-0.2945	0.2542
Lysen	Real	6903	-0.0088	0.1407	-0.0161	0.0017	-0.0007	-0.2655	0.2641
Lysen Offshore	Real	1378	-0.0198	0.1418	-0.0293	0.0038	-0.0124	-0.2702	0.2454
Netherlands	Real	1128	-0.0224	0.1368	-0.0307	0.0041	-0.0092	-0.2747	0.2563
Nissum	Real	1711	-0.0180	0.1385	-0.0237	0.0033	-0.0510	-0.3109	0.2090
Norway	Real	153	-0.0626	0.1206	-0.0746	0.0098	-0.0127	-0.2754	0.2500
Nykøbing Bugt	Real	1830	-0.0170	0.1452	-0.0224	0.0034	-0.0107	-0.2853	0.2639
Struer	Real	3828	-0.0122	0.1423	-0.0147	0.0023	-0.0049	-0.2799	0.2701
Sweden	Real	1128	-0.0227	0.1387	-0.0266	0.0041	-0.0139	-0.2855	0.2577
Wadden Sea	Real	8646	-0.0080	0.1388	-0.0185	0.0015	-0.0001	-0.2654	0.2651
Agger Tange	Simulations	499500	-0.0010	0.1307	-0.0079	0.0002	0.0057	-0.2424	0.2539
Branden	Simulations	499500	-0.0010	0.1294	-0.0082	0.0002	0.0062	-0.2402	0.2526
Dråby	Simulations	499500	-0.0010	0.1292	-0.0089	0.0002	0.0062	-0.2399	0.2523
Hals	Simulations	499500	-0.0010	0.1318	-0.0077	0.0002	0.0050	-0.2469	0.2569
Isefjord	Simulations	499500	-0.0010	0.1375	-0.0079	0.0002	0.0054	-0.2561	0.2669
Sallingsund	Simulations	499500	-0.0010	0.1338	-0.0077	0.0002	0.0053	-0.2501	0.2607
Løgstør	Simulations	499500	-0.0010	0.1362	-0.0070	0.0002	0.0046	-0.2540	0.2633

Lysen	Simulations	499500	-0.0010	0.1310	-0.0073	0.0002	0.0053	-0.2449	0.2555
Lysen Offshore	Simulations	499500	-0.0010	0.1303	-0.0085	0.0002	0.0068	-0.2394	0.2530
Netherlands	Simulations	499500	-0.0010	0.1308	-0.0090	0.0002	0.0061	-0.2423	0.2545
Nissum	Simulations	499500	-0.0010	0.1396	-0.0108	0.0002	0.0058	-0.2494	0.2611
Norway	Simulations	499500	-0.0010	0.1324	-0.0077	0.0002	0.0055	-0.2466	0.2577
Nykøbing Bugt	Simulations	499500	-0.0010	0.1355	-0.0065	0.0002	0.0044	-0.2543	0.2631
Struer	Simulations	499500	-0.0010	0.1336	-0.0077	0.0002	0.0048	-0.2503	0.2599
Sweden	Simulations	499500	-0.0010	0.1326	-0.0076	0.0002	0.0054	-0.2464	0.2573
Wadden Sea	Simulations	499500	-0.0010	0.1287	-0.0096	0.0002	0.0065	-0.2377	0.2508

Appendix 7.1: Stakeholder interview questions

Name of company: Interview date:

Name of person interviewed: Interviewed by phone by:

Questions for fishers that gave up gigas fishery

1) Hvorfor er du holdt op med at fiske stillehavsøsters?
2) Har du gjort dig nogle positive erfaringer med fiskeri af stillehavsøsters?
3) Hvem afsatte du dine stillehavsøsters til?
4) Var kvaliteten og mængden af stillehavsøstersene tilfredsstillende?
5) Hvilke områder fisker du hovedsageligt i og på hvilke vanddybder?
6) Hvilket redskab og hvor mange ad gange bruger du?
7) Overvejer du at ansøge om fiskeri af stillehavsøsters igen – hvorfor/hvorfor ikke?
8) Er der nogle forhold som skal ændres før du vil overveje at ansøge om fiskeri af stillehavsøsters igen?

Name of company: Interview date:

Name of person interviewed: Interviewed by phone by:

Questions for fishers <u>currently</u> fishing and lands gigas.

1) Hvad er jeres hidtidige erfaringer med fiskeri af stillehavsøsters?
2) Hvad fungerer godt og hvorfor?
3) Hvad fungerer mindre godt, hvor kan forholdene forbedres?
4) Hvem afsætter du dine stillehavsøsters til?
5) Hvordan er kvaliteten af stillehavsøstersene?
6) Hvilke områder fisker du hovedsageligt i og på hvilke vanddybder?
7) Hvilket redskab og hvor mange ad gange bruger du?
8) Hvordan ser du fremtidsmulighederne for udvikling af stillehavsøstersfiskeri? Er der nogle.
9) Hvor vigtigt er stillehavsøsters for dit fiskeri?

Name of company: Interview date:

Name of person interviewed: Interviewed by phone by:

Questions for industry/traders

1) Hvad er jeres hidtidige erfaringer fra salg og forarbejdning af stillehavsøsters?
2) Hvem køber I stillehavsøsters fra?
3) Hvad fungerer godt og hvorfor?
4) Hvad fungerer mindre godt, hvor forholdene forbedres?
5) Hvor afsættes stillehavsøstersene til? Og hvilke produkter afsættes?
6) Hvor store mængder afsætter I årligt? Og er der nogen begrænsninger for jeres salg?
7) Hvordan er kvaliteten og priserne for stillehavsøsters?
8) Hvordan ser I fremtidsmulighederne for udvikling af stillehavsøstersfiskeri og er der nogen fordringer?
9) Hvor vigtigt er stillehavsøsters for jeres virksomhed – indtægt, brand etc.?
10) Hvad er jeres erfaringer i forhold til depoter – overlevelse og kvalitet?

Appendix 8.1: Tables mini-dredge assessment

Table 1. Mini-dredge mean catch by weight (g/m^2 dredged ±SE, N=18) in the dense and offshore areas. * Total by-catch is only live organisms, excluding rocks and shells.

	Total Catch/Bio-mass (g/m^2 ±SE)	Pacific Oysters (g/m^2 ±SE)		By-catch (g/m^2 ±SE)					
Catch		Live	Dead	Total*	Vertebrates	Bivalves	Invertebrates	Algae	Shells
Dense	1 579 (±140)	118.7 (±21.7)	-	19.9 (±3.6)	0.02(±0.07)	6.0 (±1.0)	2.1 (±0.4)	11.9 (±2.7)	1 072.5 (±157.4)
Offshore	822 (±109)	479.1 (±94.4)	-	61.1 (±8.6)	0	37.5 (±5.4)	0.6 (±0.2)	23.1 (±4.3)	47.9 (±19.5)

Table 2. Mini-dredge mean catch (number/m^2 dredged ±SE, N = 18) in the dense and offshore areas.

	Pacific Oysters (number/m^2 ±SE)					By-catch (number/m^2 ±SE)				
Catch	Live	Dead	Ind.	Clumps	Weight (g) per oyster	Vertebrates	Bivalves	Invertebrates	Algae	Shells
Dense	0.86 (±0.1)	10.8 (±1.3)	0.58 (±0.1)	0.28 (±0.1)	145.7 (±12.7)	0.004 (±0.004)	4.8 (±0.9)	0.9 (±0.2)	-	-
Offshore	2.08 (±0.4)	0.2 (±0.1)	1.41 (±0.2)	0.67 (±0.2)	216.5 (± 8.1)	0	3.0 (±0.4)	0.1 (±0.1)	-	-

Table 3. Quadrat samples: Species richness (S, number of species), abundance (number of individuals), Shannon's diversity (*H*) and evenness (*E*) indices. A total of 6 species were identified, 5 in the dense population and 6 in the offshore population. Abundance was higher at the dense (N = 481) than at the offshore population (N = 67).

Quadrat Site	Population	Treatment	S	N	*H*	*E*
I-2	Dense	Impact	2	13	0.271189	0.391244
I-3	Dense	Impact	1	11	0	1
I-6	Dense	Impact	2	7	0.59827	0.863121
I-8	Dense	Impact	3	33	0.645196	0.587283
I-10	Dense	Impact	3	18	0.854415	0.777722
I-12	Dense	Impact	3	13	0.535961	0.487853
I-13	Dense	Impact	1	13	0	1
I-14	Dense	Impact	3	19	0.536665	0.488493
I-15	Dense	Impact	2	31	0.142506	0.205593
C-19	Dense	Control	3	53	0.645325	0.5874
C-20	Dense	Control	3	27	0.683739	0.622366
C-21	Dense	Control	3	11	0.93477	0.850864
C-22	Dense	Control	2	30	0.610864	0.881291
C-23	Dense	Control	2	23	0.178845	0.258019
C-24	Dense	Control	3	13	0.858741	0.78166
C-25	Dense	Control	3	27	0.914622	0.832525
C-26	Dense	Control	1	91	0	1
C-27	Dense	Control	4	48	0.821256	0.592411
C-28	Offshore	Control	2	4	0.562335	0.811278
C-29	Offshore	Control	1	1	0	1
C-30	Offshore	Control	3	5	0.950271	0.864974
C-31	Offshore	Control	4	5	1.316787	0.949861
C-32	Offshore	Control	0	0	0	
C-33	Offshore	Control	3	8	0.900256	0.819448
C-34	Offshore	Control	0	0	0	
C-35	Offshore	Control	2	2	0.693147	1
C-36	Offshore	Control	1	2	0	1
I-37	Offshore	Impact	1	1	0	1
I-38	Offshore	Impact	1	5	0	1
I-39	Offshore	Impact	1	3	0	1
I-40	Offshore	Impact	1	1	0	1
I-41	Offshore	Impact	1	1	0	1
I-42	Offshore	Impact	1	3	0	1
I-43	Offshore	Impact	2	7	0.59827	0.863121
I-44	Offshore	Impact	1	5	0	1
I-45	Offshore	Impact	4	14	0.754997	0.544615

Table 4. Quadrat samples: Species richness (S, number of species), abundance (number of individuals), Shannon's diversity (*H*) and evenness (*E*) indices. A total of 55 species were identified, 44 in in the reef/dense population and 39 in the offshore population. Abundance was higher at the reef/dense (N = 3279) than at the offshore population (N = 787).

Core Site	Population	Treatment	S	N	*H*	*E*
Site 01	Reef=Dense	Impact	19	164	2.393919	0.813031
Site 04	Reef=Dense	Impact	14	142	2.084983	0.790048
Site 05	Reef=Dense	Impact	12	86	1.972851	0.793934
Site 07	Reef=Dense	Impact	14	96	2.175183	0.824227
Site 09	Reef=Dense	Impact	14	195	2.046378	0.77542
Site 11	Reef=Dense	Impact	13	108	2.179207	0.84961
Site 16	Reef=Dense	Impact	17	115	2.268109	0.800543
Site 17	Reef=Dense	Impact	11	94	1.887749	0.787253
Site 18	Reef=Dense	Impact	17	161	2.155846	0.760919
Site 43	Reef=Dense	Control	16	223	2.016901	0.727443
Site 46	Reef=Dense	Control	15	201	1.956279	0.722394
Site 47	Reef=Dense	Control	17	189	2.428579	0.857182
Site 50	Reef=Dense	Control	17	233	1.765946	0.623302
Site 50b	Reef=Dense	Control	18	85	2.38499	0.82515
Site 51	Reef=Dense	Control	18	206	2.004137	0.693384
Site 52	Reef=Dense	Control	17	262	2.035871	0.718573
Site 53	Reef=Dense	Control	14	163	1.99881	0.757396
Site 54	Reef=Dense	Control	20	556	1.874846	0.625839
Site 55	Offshore	Control	11	79	1.956851	0.81607
Site 56	Offshore	Control	6	37	1.463133	0.81659
Site 57	Offshore	Control	15	48	2.449985	0.904705
Site 58	Offshore	Control	9	31	1.920454	0.874036
Site 59	Offshore	Control	9	26	2.118272	0.964067
Site 60	Offshore	Control	6	40	0.854253	0.476767
Site 61	Offshore	Control	14	134	1.521575	0.57656
Site 62	Offshore	Control	12	71	1.923498	0.774073
Site 63	Offshore	Control	10	26	1.957888	0.8503
Site 64	Offshore	Impact	8	16	1.808046	0.869486
Site 65	Offshore	Impact	9	19	2.068885	0.94159
Site 66	Offshore	Impact	9	22	1.893284	0.861671
Site 67	Offshore	Impact	8	28	1.807682	0.869311
Site 68	Offshore	Impact	13	41	2.121785	0.827223
Site 69	Offshore	Impact	8	22	1.873965	0.901186
Site 70	Offshore	Impact	10	29	2.087942	0.906782
Site 71	Offshore	Impact	16	91	2.306529	0.831904
Site 72	Offshore	Impact	9	27	1.951529	0.888179

Appendix 8.2: Tables floating escavator assessment

Table 1. Excavator catches by weight (kg/m^2 or g/m^2 ±SE, N=3) in the low, medium and high-density areas. * Total by-catch is only live organisms, excluding rocks and shells.

	Total Catch (kg/m^2)	Pacific Oysters (kg/m^2)		By-catch (g/m^2 ±SE)					Shells (kg/m^2)	Stones (kg/m^2)
Density		Live	Dead	Total*	Fish	Bivalves	Other Invertebrates	Algae		
Low	9.4 (±2.6)	0.45 (±0.2)	4.13 (±1.1)	40 (±3)	1.4 (±0.8)	8.1 (±0.4)	7.0 (±3.1)	23.7 (±11.8)	2.3 (±0.4)	1.9 (±1.6)
Medium	33.9 (±1.7)	0.41 (±0.1)	9.17 (±2.1)	1,565 (±661)	0.7 (±0.5)	7.2 (±0.7)	140.5 (±90.6)	1,374 (±579)	8,7 (±7.5)	15.4 (±7.7)
High	57.1 (±3.7)	7.31 (±2.9)	20.53 (±1.7)	4,318 (±916)	4.4 (±0.4)	231 (±144)	765.9 (±198.8)	3,227 (±628)	50.7 (±3.3)	0.1 (±0.1)

Table 2. Excavator catches by number (number/m^2 ±SE, N=3) in the low, medium and high density areas. * Total by-catch is only live organisms, excluding rocks and shells.

	Pacific Oysters (kg/m^2)		By-catch (g/m^2 ±SE)				
Density	Live	Dead	Total*	Fish	Bivalves	Other Invertebrates	
Low	2.77 (±1.25)		3.8 (±1.6)	0.33 (±0.19)	0.78 (±0.44)	2.67 (±1.17)	
Medium	2.44 (±0.29)	116.7 (±22.4)	34.1 (±14.1)	0.67 (±0.51)	7.23 (±0.72)	26.24 (±13.92)	
High	28.33 (±7.06)	170.2 (±38.7)	160.4 (±41.1)	4.42 (±0.42)	5.25 (±3.12)	142.2 (±40.28)	

Table 3. Excavator catches: Species richness (S, number of species), abundance (number of individuals/m^2), Shannon's diversity (*H*) and evenness (*E*) indices. A total of 13 species were identified. Abundance was higher at the high than in medium and low density areas (N = 566, 110, and 20, respectively).

Plot	Density	Treatment	Time	S	N	*H*	*E*
H4	High	Impact	Before	7	248	1.18	0.61
H5	High	Impact	Before	5	96	1.13	0.70
H6	High	Impact	Before	7	223	1.21	0.62
M7	Medium	Impact	Before	5	23	1.29	0.80
M8	Medium	Impact	Before	5	65	0.79	0.49
M9	Medium	Impact	Before	8	22	1.44	0.69
L1	Low	Impact	Before	5	5	1.32	0.82
L2	Low	Impact	Before	10	12	1.80	0.78
L3	Low	Impact	Before	4	3	1.21	0.88

Table 4. Control sites before impact: Species richness (S, number of species), abundance (number of individuals/m^2), Shannon's diversity (*H*) and evenness (*E*) indices. A total of 6 species were identified. Abundance was higher at the high and medium densities (N = 992 and 848, respectively) than at the low density area (N = 48).

Plot	Density	Treatment	Time	S	N	*H*	*E*
CH1	High	Control	Before	6	320	1.37	0.77
CH1	High	Control	Before	0	0	0.00	
CH2	High	Control	Before	3	176	0.99	0.91
CH2	High	Control	Before	0	0	0.00	
CH3	High	Control	Before	1	208	0.00	
CH3	High	Control	Before	0	0	0.00	
CH4	High	Control	Before	4	288	1.12	0.81
CH4	High	Control	Before	0	0	0.00	
CM1	Medium	Control	Before	3	240	0.63	0.57
CM1	Medium	Control	Before	0	0	0.00	
CM2	Medium	Control	Before	1	96	0.00	1.00
CM2	Medium	Control	Before	0	0	0.00	
CM3	Medium	Control	Before	2	96	0.69	1.00
CM3	Medium	Control	Before	0	0	0.00	
CM4	Medium	Control	Before	3	416	0.96	0.87
CM4	Medium	Control	Before	0	0	0.00	
CL1	Low	Control	Before	0	0	0.00	
CL1	Low	Control	Before	0	0	0.00	
CL2	Low	Control	Before	1	16	0.00	1.00
CL2	Low	Control	Before	0	0	0.00	
CL3	Low	Control	Before	0	0	0.00	
CL3	Low	Control	Before	0	0	0.00	
CL4	Low	Control	Before	2	32	0.69	1.00
CL4	Low	Control	Before	0	0.00	0.00	

Table 5. Control and impact sites 22 moths after impact: Species richness (S, number of species), abundance (number of individuals/m^2), Shannon's diversity (*H*) and evenness (*E*) indices. A total of 16 species were identified. Abundance was higher at the control sites (N = 2800 and 1184) than at the impact sites (N = 768 and 400), except for low density areas due to the high abundance of amphipods (N = 1872) that accounted for 80% of abundance in impact sites (N = 2336) but only 56% (N = 512) in control sites (N = 912).

Plot	Density	Treatment	Time	S	N	Shannon	Eveness
CH1	High	Control	After	6	256	1.440235	0.80381
CH1	High	Control	After	5	288	1.541833	0.957995
CH2	High	Control	After	6	384	1.365307	0.761992
CH2	High	Control	After	9	800	1.496298	0.680995
CH3	High	Control	After	10	560	1.820509	0.790637
CH3	High	Control	After	10	512	1.67936	0.729337
H4	High	Impact	After	4	240	0.953271	0.68764
H4	High	Impact	After	3	304	0.409459	0.372705
H5	High	Impact	After	1	48	0	
H5	High	Impact	After	0	0	0	
H6	High	Impact	After	5	96	1.262084	0.784177
H6	High	Impact	After	2	80	0.500402	0.721928
CM1	Medium	Control	After	4	256	1.143275	0.824699
CM1	Medium	Control	After	3	96	0.867563	0.78969
CM2	Medium	Control	After	5	256	0.908909	0.564737
CM2	Medium	Control	After	2	208	0.690186	0.995727
CM3	Medium	Control	After	6	272	0.972745	0.542899
CM3	Medium	Control	After	2	96	0.693147	1
M7	Medium	Impact	After	2	64	0.562335	0.811278
M7	Medium	Impact	After	2	224	0.257319	0.371232
M8	Medium	Impact	After	0	0	0	
M8	Medium	Impact	After	1	16	0	
M9	Medium	Impact	After	1	16	0	
M9	Medium	Impact	After	1	80	0	
CL1	Low	Control	After	2	128	0.562335	0.811278
CL1	Low	Control	After	1	112	0	
CL2	Low	Control	After	6	480	0.70591	0.393976
CL2	Low	Control	After	1	16	0	
CL3	Low	Control	After	1	160	0	
CL3	Low	Control	After	1	16	0	
L1	Low	Impact	After	0	0	0	
L1	Low	Impact	After	1	144	0	
L2	Low	Impact	After	4	784	0.15885	0.114586
L2	Low	Impact	After	2	272	0.251772	0.363231
L3	Low	Impact	After	4	240	0.546741	0.39439
L3	Low	Impact	After	6	896	0.357022	0.199258

Table 6. SIMPER dissimilarity analyses at control sites before and 22 months after impact in high (top), medium (middle) and low density areas (bottom).

Overall Average Dissimilarity: 79.03 Taxon	Av. dissim	Contrib. %	Cumulative %	Mean After	Mean Before
Worms	12.04	15.23	15.23	3.94	0
M. edulis	10.82	13.7	28.92	4.56	1.36
C. gigas	10.27	13	41.92	3.92	0.841
L. littorea	10.04	12.71	54.63	3.93	1.95
Palaemon sp. or C. crangon	6.858	8.677	63.31	2.48	0
Pomatoschistus minutus	6.167	7.803	71.11	2.22	0
C. fornicata	5.212	6.595	77.71	1.59	1.41
N. reticulata	4.201	5.315	83.02	1.12	0.874
Isopoda	2.819	3.566	86.59	0.944	0
Metridium senile	2.23	2.822	89.41	0.944	0
C. maenas	2.031	2.57	91.98	0.583	0.437
Amphipoda	1.86	2.354	94.33	0.788	0
Patella vulgata/T testudinalis	1.134	1.435	95.77	0.472	0
M. arenaria	1.115	1.411	97.18	0.472	0
H. panicea	1.115	1.411	98.59	0.472	0
Leptochiton sp.	1.115	1.411	100	0.472	0
Overall Average Dissimilarity: 77.28 Taxon	Av. dissim	Contrib. %	Cumulative %	Mean After	Mean Before
L. littorea	19.99	25.87	25.87	4.36	2.4
Worms	19.9	25.75	51.61	2.96	0
M. edulis	11.06	14.3	65.92	1.75	0.959
C. gigas	10.45	13.52	79.44	1.89	0
Pomatoschistus minutus	4.309	5.575	85.01	0.944	0
N. reticulata	3.898	5.043	90.05	0	0.841
Amphipoda	3.525	4.561	94.61	0.847	0
C. fornicata	2.197	2.843	97.46	0	0.591
M. arenaria	1.965	2.543	100	0.472	0
Overall Average Dissimilarity: 99.58 Taxon	Av. dissim	Contrib. %	Cumulative %	Mean After	Mean Before
Amphipoda	47.46	47.67	47.67	2.75	0
L. littorea	21.36	21.45	69.12	1.37	0
Worms	8.653	8.69	77.81	0.762	0
C. gigas	7.535	7.567	85.38	0	0.708
M. edulis	4.77	4.79	90.17	0.472	0.354
Palaemon sp or C. crangon	3.174	3.187	93.35	0.732	0
Pomatoschistus minutus	2.525	2.536	95.89	0.583	0
C. maenas	2.046	2.055	97.95	0.472	0
Isopoda	2.046	2.055	100	0.472	0

Table 7. SIMPER dissimilarity analyses 22 months after excavator impact between control and impact sites in high (top), medium (middle) and low density areas (bottom).

Overall Average Dissimilarity: 73.03					
Taxon	Av. dissim	Contrib. %	Cumulative %	Mean Control	Mean Impact
M. edulis	11.64	15.94	15.94	4.56	0.472
Worms	10.01	13.7	29.64	3.94	0.472
C gigas	8.802	12.05	41.7	3.92	0.944
L. littorea	6.985	9.565	51.26	3.93	3.06
Palaemon sp. or C. crangon	6.119	8.379	59.64	2.48	0.472
Pomatoschistus minutus	5.745	7.868	67.51	2.22	0
N. reticulata	5.089	6.968	74.48	1.12	1.77
C. fornicata	4.065	5.566	80.04	1.59	0
Isopoda	3.004	4.113	84.16	0.944	0.472
C. maenas	2.337	3.2	87.36	0.583	0.472
Metridium senile	2.111	2.891	90.25	0.944	0
Amphipoda	1.761	2.412	92.66	0.788	0
C. edule	1.119	1.533	94.19	0	0.472
Patella vulgata/T testudinalis	1.073	1.469	95.66	0.472	0
M. arenaria	1.056	1.446	97.11	0.472	0
H. panicea	1.056	1.446	98.55	0.472	0
Leptochiton sp.	1.056	1.445	100	0.472	0
Overall Average Dissimilarity: 67.25	Av. dissim	Contrib. %	Cumulative %	Mean Control	Mean Impact
Worms	17.58	26.14	26.14	2.96	0.472
L. littorea	14.24	21.18	47.32	4.36	2.74
C. gigas	10.21	15.19	62.5	1.89	0
M. edulis	10	14.87	77.38	1.75	0
Pomatoschistus minutus	4.268	6.347	83.72	0.944	0
Amphipoda	3.504	5.211	88.93	0.847	0
Isopoda	3.183	4.733	93.67	0	0.472
C. edule	2.306	3.428	97.09	0	0.472
M. arenaria	1.954	2.905	100	0.472	0
Overall Average Dissimilarity: 73.69					
Taxon	Av. dissim	Contrib. %	Cumulative %	Mean Control	Mean Impact
Amphipoda	23.89	32.42	32.42	2.75	3.93
L. littorea	14.28	19.38	51.8	1.37	0.829
Worms	8.571	11.63	63.43	0.762	1.32
Pomatoschistus minutus	8.498	11.53	74.97	0.583	1.64
Isopoda	6.362	8.633	83.6	0.472	1.42
Palaemon sp. or C. crangon	5.671	7.696	91.29	0.732	1.05
C. maenas	3.307	4.488	95.78	0.472	0.472
H. panicea	1.612	2.188	97.97	0	0.472
M. edulis	1.496	2.03	100	0.472	0
C. gigas	0	0	100	0	0

Appendix 9.1: Gastronomic possibilities of large Pacific oysters

Authors: Katla Hrund Björnsdóttir and Roberto Flore from DTU SkyLab FoodLab

Several experiments were conducted both regarding opening and cooking large wild pacific oysters (*C. gigas*) (Figure 1) at DTU Skylab's FoodLab in March and April of 2019. The over-all conclusion of the experiments was that by freezing the large oysters in the shell they are easily opened with conventional shucking methods. Since the large oysters are not eaten raw or whole, freezing them did not cause a negative effect on the texture, although more research needs to be done here. The larger individuals contained algae in their digestive system which gave them an unpleasant look and texture for eating raw, but the algae gave the oysters a sweet fresh flavor that could be desirable in food additives and/or distilled beverages. Further research is needed to optimize recipes and pathogen research needs to be conducted.

Figure 1 An example of oyster clusters used in the experiments. To the left and right side of the photos, in centre an oyster of a size commonly served raw.

The oysters were delivered to DTU Skylab on 22.03.19, alive and cooled in a closed foam box with wet paper to keep humid. All experiments, aside from the opening of fresh oysters, were performed with oysters that had previously been frozen un-shucked, thawed in a refrigerator at 5°C over night, shucked and cleaned at the same day as experiments were started.

Opening of fresh live oysters

At the same day as the oysters were delivered to Skylab an experiment was made on opening the oysters using a table vice and shucking knife. By applying pressure on the ventral and dorsal parts of the oyster it was possible to force the shell open enough to then use traditional shucking methods (Figure 2). This method however fractured and grinded the shell causing parts of the shell to get into the oyster which was difficult to clean out.

Figure 2 A vice pressuring the ventral and dorsal parts of a single oyster in a cluster for opening.

Cooking whole un-processed oysters

Un-shucked oysters were steamed in a vacuum closed bag for 8 minutes at 80°C. After steaming the oysters were immediately shucked and tasted for flavour and consistency. The taste of the plane meat was sweet with a strong algae flavour.
The oysters had many algae in their digesting system (Figure 3), which gave them an unpleasant look and consistency.

Figure 3 Algae in the digestion system of the oysters after cooking.

Cooking marinated whole oysters

Oysters were poached in a vacuum sealed bag at 80 °C for 3 minutes, immediately cooled down in an ice bath and marinated with a 4% marinade (6% sugar + 2.5% salt + 1.5% citric acid) for 30 minutes 5°C. They were then dehydrated at 45°C for 3.5 hours and then roasted at 130°C for 10 minutes.
The flavour was sweet but as with the un-processed oysters their digestive system was full of algae giving a large section of the oyster an unpleasing look and consistency for presentation (Figure 3).

Freeze drying

Four frozen oysters were freeze dried using BUCHI Lyovapor® L-200 and grinded to powder using a thermomixer from Vorwerk® (Figure 4)

Figure 4 Freeze dried oysters after grinding.

Drying whole oysters

Both plain oysters and oysters marinated with 4% marinade for 30 minutes and over-night at 5°C and, with 8% and 12% marinade for 30 minutes at 5°C and dried at 75°C for 24 hours using Excalibur® Food Dehydrator. The total weight loss of the oysters was then calculated (Table 1). After drying they were grinded to powder using a thermomixer from Vorwerk® (Figure 5).

Figure 5 The powders made from drying and grinding the oysters to powder. From the left: freeze dried, plain, 4% marinade, 8% marinade, & 12% marinade

Table 1 Weight loss of whole oysters from drying.

Method	Start weight	Final weight	Weight loss %
Plain 1	305	41.5	88.5
Plain 2	240	35	85.4
4% marinating	295	52	82.4
4% marinating over night	246	43.5	82.3
8% marinating	304	57	81.3
12% marinating	295	59.5	79.8
Average	280.8	48.1	83.3

Drying minced oysters

Oysters were minced using a Dynamic® master whisk and spread in a plastic container with three different thickness layers and dried overnight at 75°C and the total weight loss of the oysters calculated (Table 2). The final texture was a crisp plate of minced oysters (Figure 6).

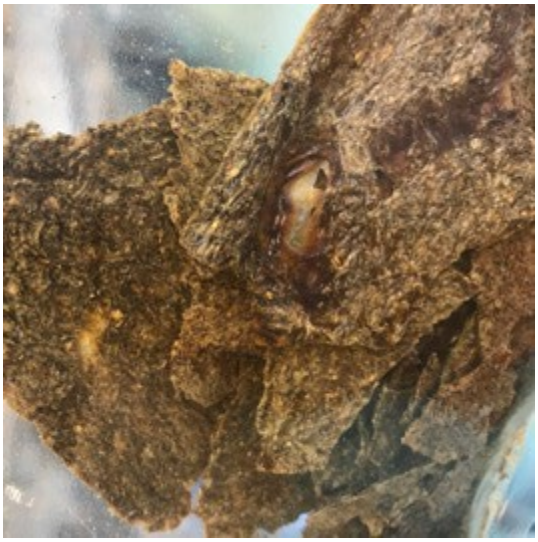


Figure 6 Dried, minced oysters.

Table 2 Weight loss of minced oysters from drying.

Method	Start weight	Final weight	Weight loss %
Plain #1	200	26	87.0
Plain #2	300.5	39.5	86.9
Plain #3	350	44	87.4
Average	283.5	36.5	87.1

Distilling oyster alcohol

Approx. 100 gr of oysters, both whole and minced, were placed in a separate 500 mL 45% alcohol solution at 5°C for 48 hours. After the 48 hours the alcohol was distilled using BUCHI Rotavapor® R-300 rotary evaporator. The final solution where the whole oysters were used had a distinct salt-ocean flavour whereas the solutions from the minced oysters had a distinct algae flavour.