



# Automated Change Detection in Satellite Imagery

**Thesis Report**

October 9, 2018

*Dirk Jan Kok*

Faculty of Aerospace Engineering



Sources cover image:

Satellite image: Sentinel-2A, ESA, 2016

TU Delft Logo: Delft University of Technology, 2018

# Automated Change Detection in Satellite Imagery

---

(ACDet)

By

D.J. Kok

in partial fulfilment of the requirements for the degree of

**Master of Science**  
in Aerospace Engineering

at the Delft University of Technology,  
to be defended publicly on October 9, 2018

Supervisors:	Dr. ir. W. van der Wal Dr. R.W. van Swol	TU Delft NLR
Thesis committee:	Dr. ir. E.J.O Schrama Dr. S.L.M Lhermitte	TU Delft TU Delft

An electronic version of this thesis is available at <http://repository.tudelft.nl/>



## Preface

This project was performed at the Netherlands Aerospace Centre (NLR) in Amsterdam at the department of Aerospace Systems, ISR and Space Utilisation (ASIS). The department focuses at the identification and development of knowledge necessary for the implementation of space technology in innovative space systems and applications.

Earth observation is one of the focus points of the department. The significant increase in the quantity and quality of remote sensing information over the past decade has lifted the possible applications of earth observation. To leverage the full potential of the available data interdisciplinary methods are sought to drive innovation.

This research is a conceptual study into the use of machine learning for remote sensing purposes to come to an automated and flexible change detection system that will not only reduce the human work load, but also increase the speed and quality of detection.

Although the research has been performed as a standalone project, it has close relation to on-going projects at NLR, and will be of value to future projects. Solely open source data as well as open source software has been used as a proof of concept of the capabilities of open source data.

## Acknowledgements

This research is the last phase of my Masters programme at Delft University of Technology. I had the pleasure to follow both the bachelor and master programme at the faculty of aerospace engineering. By defending my thesis on the 9<sup>th</sup> October 2018, an amazing period of my life will have come to an end. I would like to thank all people, friends, family, and teachers who have made my student time as great and successful as it was, as well as Delft, an amazing place to live as a student.

In my thesis research I have had extensive guidance from my supervisor at NLR, Rob van Swol, and my supervisor at the TU Delft, Wouter van der Wal. I would like to thank them both for their input and enthusiasm that have made this research more interesting and fun than I would have ever imagined and have it finished on schedule.

## Abstract

A reduction in system and launch costs and an increase in hardware capabilities has caused an immense increase in the number of satellites, both government and commercial, observing the earth. With that the quantity of earth observation data has grown exponentially combined with spatial and temporal resolutions never seen before.

The sheer amount of data required to look outside the field of earth observation for methods to harness the full potential of it. Human analysis of satellite imagery is simply not possible anymore at this point. The rapid developments in computing power, digital storage, and machine learning algorithms have given an opportunity to set the next step in earth observation. With an interdisciplinary approach this research focused on the question how machine learning with the input of time series data can be used to come to an accurate, flexible and most of all automated change detection system for remote sensing.

A novel approach of combining a multi-layer perceptron neural network with graph based image segmentation provides a highly accurate object detection method, consistent throughout a time series of images. With the use of a running average on Sentinel-2A and Sentinel-2B imagery insignificant and temporary changes are minimized and the focus is put on detecting lasting changes in objects on a scale matching the 10x10 m spatial resolution of the imagery. The satellite data is combined with independent Information on the same area from the Top10NL and BasisRegistratie Gewaspercelen databases to an aggregate of 10 elementary classes on which the neural network is trained for detection.

The system is able to flag significant changes without predefining the object type. An area is qualified as changed when both the object to which it is assigned and its class have changed. Where the disappearance of an existing or appearance of a new object is considered as object change. Maintaining strict criteria limits the amount of information shown on the resulting change map. A satellite image analyst can thus monitor large areas. The analyst can focus on assessing the changes flagged by the system rather than detecting them, increasing speed and quality of the task.

The newly constructed island the Marker Wadden is used to validate the system performance. The construction perfectly fit the timeline of the imagery used in the project, and the official project planning allowed for validation of the change detection system with an external data source.

## Contents

Preface .....	4
Acknowledgements.....	5
Abstract.....	6
Contents.....	7
1 Introduction .....	9
2 Research Goals.....	11
3 Methodology.....	13
3.1.1 Classification .....	13
3.1.2 Change detection .....	13
3.2 Classification .....	14
3.2.1 Imagery .....	14
3.2.2 Labelling.....	18
3.2.3 Polygon Overlay .....	20
3.2.4 Training and Verification Set.....	21
3.2.5 Classifier .....	22
3.3 Feature Engineering.....	27
3.3.1 Standardization .....	31
3.4 Segmentation.....	32
3.5 Classification Cleaning.....	36
3.6 Change Detection.....	38
3.6.1 Overlay .....	38
3.6.2 Segment Matching.....	39
3.6.3 Change Map .....	40
3.6.4 Filtering .....	42
4 Verification.....	43
5 Experimental Set-Up .....	50
5.1 Marker Wadden Test-Case .....	50
5.2 Data Source.....	51
5.3 Set-Up .....	51
6 Validation & Results.....	52
6.1 Marker Wadden Validation per Construction Phase.....	54
6.1.1 Phase 0 .....	55
6.1.2 Phase 1 .....	55

6.1.3	Phase 2 .....	56
6.1.4	Phase 3 .....	57
6.1.5	Phase 4 .....	57
6.1.6	Phase 5 .....	58
6.1.7	Phase 6 .....	59
6.1.8	Phase 7 .....	60
6.2	Time Series Change .....	61
6.3	Large Area Map .....	62
6.4	Error Sources .....	64
7	Conclusions & Discussion .....	66
8	Recommendations .....	69
8.1	Detailed Improvements .....	69
8.2	Global Improvements .....	69
	Bibliography .....	70
	List of Figures .....	75
	List of Tables .....	78
	List of Abbreviations .....	79
	A – Feature Distributions .....	80
	B - Images .....	91
	C - System Specification .....	92



## 1 Introduction

Earth observation was amongst the first applications of space technology, and still plays an important role. As often is the case one of the first rewards came from its military / political use (European Patent Office, 2016). Satellites provide an ‘eye in the sky’ that allows inspection of foreign territory (Beckers, et al., 2015). For over fifty years it offered the capability to detect and inspect military infrastructure (ESA, 2009). In the early days of remote sensing the ‘simple’ task of registering infrastructure was the main objective. This was performed by human inspection of the images (Baumann, 2009). With the increase of both spatial and temporal resolution the capabilities have expanded to monitoring areas of interest with a temporal resolution of a single day (Planet, 2018) for the detection of change in infrastructure. Despite the advanced technology providing the satellite imagery, the detection of change is still to a large extent a human task. The exponential increase in image data has now overwhelmed the capacity of human inspection. A need for automated inspection has become pressing (Zhang, 2010).

At the mean time computing power has also grown exponentially and, equally important, new analytical techniques including statistical procedures and self-learning algorithms geared to large data analysis have been developed (Zhu, 2017). The combination allows for automated analysis of large satellite image data bases. The first automation procedures still worked under human supervision and diminished the load of routine tasks for these human analysts. At present this automation process has reached a level where the aim has become to reduce the human input to a minimum by developing artificial intelligence (AI) systems that as much as possible take over the human task of interpretation (Zhang, 2010).

The detection of bunker improvements and new structures in the Baltic Sea exclave of Kaliningrad (Tucker, 2018) forms a recent example of change detection using a combination of machine learning and human input. During a three-month period images were taken of a known Russian munition storage facility in Baltiysk, near the Polish border. The noted changes were of a detail of detecting the levelling of bunkers and the reinforcement of berms around them. It is interesting to see news with these details out in the open. It shows how earth observation is not only moving towards automated analysis but also from a restricted environment into the private and open domain. It is becoming open source. This is emphasized by the fact that the analysis was performed by Planet, a private satellite company. The analysis was performed using human-machine integrated location-based analytics according to Matt Hall, a senior geospatial analyst at the company 3GIMBALS which provides similar services. An automated machine learning program detected structural improvements, the berm production and presence of a rail-road. Human analysis provided interpretations of these findings, namely that they point to an upgrade of the installation to becoming a nuclear facility. It confirmed the presence of nuclear weapons in the Russian enclave close to Western Europe. Theoretically the specific combination of changes could be linked to a conclusion “nuclear facility” in an automated fashion but considering the severity of the implications human input is required finally.

The news highlights the power of automated change detection while at the same time showing room for improvement. In the example the observed area was a known military site. This limited the scope of the search to that small area. An analysis of large areas, without prior knowledge of where to focus remains a challenge. Next to that the algorithm was set to detect specific construction

types, such as railroads. The detection of specific types of construction is a rapidly developing application in the field of artificial intelligence. The common approach is to use large quantities of examples (data) to train the machine learning algorithm into recognizing specific objects and then apply the trained algorithm to unknown data (Heaton, 2012).

A next step will be to develop algorithms that detect any changes that occur in large flag areas without prior focus on a specific type of object or a specific location. For strategic military purposes and for causes like climate control automated change detection can be an extremely valuable tool especially when it can be applied on a large and ultimately global scale. An essential element of future algorithms will be the ability to observe changes without pre-set limits of area or object type of interest. Such algorithms will be limited only by the information present in the input data. There is currently no automated system with such capabilities that can take over the human analysis part in earth observation. Arguably complete removal of human input from the analysis procedure may not be the goal. Useful would be to have an automated program which flags significant changes on a large area map. This will help to monitor for instance changes in large sea areas. Such an algorithm applied to the South China Sea could have improved or even speeded up detecting the construction of islands and military infrastructure on them by China. China uses these Spratley Island in a geopolitical dispute to claim the South China Sea (Macias, 2018). Currently the changes were detected at a stage where it was a fait accompli. Early detection of ongoing work could have supported efforts to halt the development.

The present Automated Change Detection (ACDet) project aims to provide an algorithm for automated detection of changes of which the nature is not predetermined on images that cover half of The Netherlands. This area was chosen as it includes the proof of concept event, construction of the Marker Wadden island, and can be handled with the limited computing power of a laptop computer. Application of the algorithm to a larger area will just require more computing power. Images taken by the Sentinel-2A and Sentinel-2B satellites are used. The images have a spatial resolution of 10x10 meter and a temporal resolution that vary between several days up to a month because images with over 50% cloud coverage are filtered out. This combination of resolutions allows detecting long-term changes with a higher sensitivity and accuracy for large infrastructure. An automated change detection system as researched in this project can give a considerable boost in the detection rate and speed as well as significantly reducing the man-hours needed to do so.

The construction of the Marker Wadden island was chosen as test case as it was performed precisely in the time-frame of the series of images available for this research. Furthermore, the official project planning provides a unique independent source for validation of the changes picked up by the ACDet system.

This project was aimed at answering the following research question:

“How can **machine learning** be used with the input of **time series data** to come to an accurate, flexible and **automated change detection** system for remote sensing?”

## 2 Research Goals

The main research question of the project was translated into the goal to produce an artificial intelligence program for automated detection of changes in satellite images (ACDet). From a review of the literature an outline was constructed for how such a program should work and possible bottlenecks were identified leading to the following five sub-questions.

### ***How to minimize the relation between the system performance and the quality of the input data?***

The accuracy of the ACDet system will depend to a large extent on the quality of the spectral input data, the labelling data used for training and the method of classification chosen (Alpaydin, 2004). The input data is an external limiting factor which can only be mitigated to the extent of choosing the best available source. Manipulation lies in the choice of the method for classification which also has a large influence. As all common methods have strengths and weaknesses there is no 'best method' (Tewkesbury, Comber, Tate, Lamb, & Fisher, 2015). For this project a comprehensive feature engineering approach is applied where spectral bands combined with mathematical indices are used to classify individual pixels, and errors in individual pixels are averaged out by combining pixels into kernels. This to lift the performance of the classification procedure beyond the limitations of single pixel based classification. A novel technique of combining segmental information is implemented to homogenize and further improve the pixel classification.

### ***What procedure can identify objects correctly and consistently in a time series of images?***

Correctly identifying classes on pixel level has its limits when it comes to applicability. The true value lies in identifying objects. Technically stated this comes down to grouping pixels which is usually done by image segmentation or clustering. All of the methods in the wide range available have their limitations. Standard methods such as edge detection and threshold segmentation are not able to cope with shifting gradient within an object (Yuheng & Hao, 2017). While other methods such as k-means or DBSCAN clustering have proven in this research unable to identify objects similar as how an operator would outline them on a satellite image. A relatively new method of graph-based image segmentation developed by Felzenszwalb and Huttenlocher in 2004 is implemented. It shows a unique ability in segmenting images similar to the human eye. Furthermore, to gain a consistent segmentation in a time series of images the novel approach of using a running average is applied.

### ***How to minimize the noise in images caused by seasons, local weather, and temporary presence of objects in an image?***

The satellite image data of the Earth's surface are subject to influences of season and local weather. When the images are used for change detection these influences are a challenge (Zhu, 2017). A second element that proved difficult is the temporary presence of objects in a picture, since the aim is to detect lasting change. To overcome these difficulties the novel approach of using a running average in a time series of images is proposed. A temporary disturbance visible in a single image can be averaged out.

***Existing change detection methods are used to detect change in two consecutive images. How to expand change detection to a time series of images?***

Images are usually analysed for change using a bi-temporal approach (Singh, 1989). When analysing time series to detect changes over a longer period of time the method used must take into account the time element of gradual changes. The approach proposed consists of multiple elements addressing this issue. The running average applied on the data set deals with the element of time. Next to that single time-point changes are added up into a cumulative change map to visualize the changes over a complete time series in a single change map.

***How to detect change of importance without pre-setting what type of object or change to look for?***

Detection and change are two very broad terms. Most present day AI-detection algorithms are trained towards specific pre-defined objects. These algorithms have proven very successful. Usually they use a convolutional neural network (CNN) to process the data. Consecutive images are fully processed by the CNN. Specific objects that are detected in a given image are then compared to the objects found in the previous or following picture to detect appearance or disappearance of objects. An example is automated road and building detection by the company CrowdAI. The CNN developed by CrowdAI is extremely good in mapping road and buildings in terms of accuracy and reliability and tracks the changes of these objects in time.

What current methods fail to achieve is to give a reliable change detection output without pre-setting the type of object one is looking for: general change detection. In this research the aim is to provide a satellite image analyst (further referred to as operator) with an output showing change flagged by the system as important, but without pre-setting a specific type of object or change and thereby filtering out information that might have been of strategic importance.

The system is set up to provide a large area map with a minimal number of flagged change items, enabling the operator to monitor a large area of interest and focus on assessing specific areas flagged by the system. In principle, by lowering the thresholds for what the algorithm considers to be a true change, the number of change flags can be increased but at the cost of a lower reliability of each flag and a higher work-load for the operator.

## 3 Methodology

ACDet is designed to automatically detect changes in time-series of satellite images. For proof of principle it was tested in this project if ACDet can detect the emergence of the Marker Wadden island in a time series of satellite images of the northern part of The Netherlands. Outside this area ACDet will also detect changes but these were not verified in this project. This methodology section describes the techniques for satellite image analysis and change detection that are applied in ACDet, the reasons for incorporating specifically these techniques and the comparison of the satellite data to independent data sources describing the infrastructure of the same surface area which is needed to put correct names on the items that ACDet detects in satellite images. Finally for as proof of concept the change maps produced by ACDet are compared to the planning of the Marker Wadden island development. Overall, ACDet has two tasks: classification and change detection.

### 3.1.1 Classification

Classification means that ACDet must correctly identify all items that are present on a specific satellite image. Change detection means that ACDet must correctly detect changes that occur in individual pixels (change of label) and consequently changes that occur in specific items made up of one or more pixels (appearance, disappearance, growth, shrinkage) that are present in the images in a time series of the same area.

The satellite images used in this project are obtained from the Sentinel L2A data base and consist of 10980X10980 individual pixels. Each pixel covers an area of 10X10 m and is described by geolocation data and multiple wavelength bands, 9 of which are used in this project (see Table 3.1). ACDet places individual pixels or, when this improves the accuracy, groups of pixels called segments into classes that are characterized by a specific set of waveband information. When the members of these classes are projected on the satellite image the classes will be distributed all over this so-called class map. To put meaning to the resulting classes the class map is then compared to independent data bases in which all items present in the surface area that is covered by the satellite images have already been classified, Top10NL and BRP. Inherently the number of classes that ACDet uses is limited to the 10 classes provided in these databases: 'water inland', 'sand', 'heath', 'grass', 'forest', 'building', 'road', 'farmland', 'sea' and 'dune'.

### 3.1.2 Change detection

The next step is to compare class maps of the same area obtained at different points in time in order to detect where and when the labels of pixels change and to translate these changes into appearance, disappearance, growth or shrinkage of specific items. The techniques that are applied in ACDet for this change detection are described in the second part of the methodology section.

## 3.2 Classification

Figure 3.1 displays an overview of the classification procedure per image. To come to a pixel level classification there is a sequence of essential steps (the orange blocks). Additional techniques are applied to enhance the result of the classification.

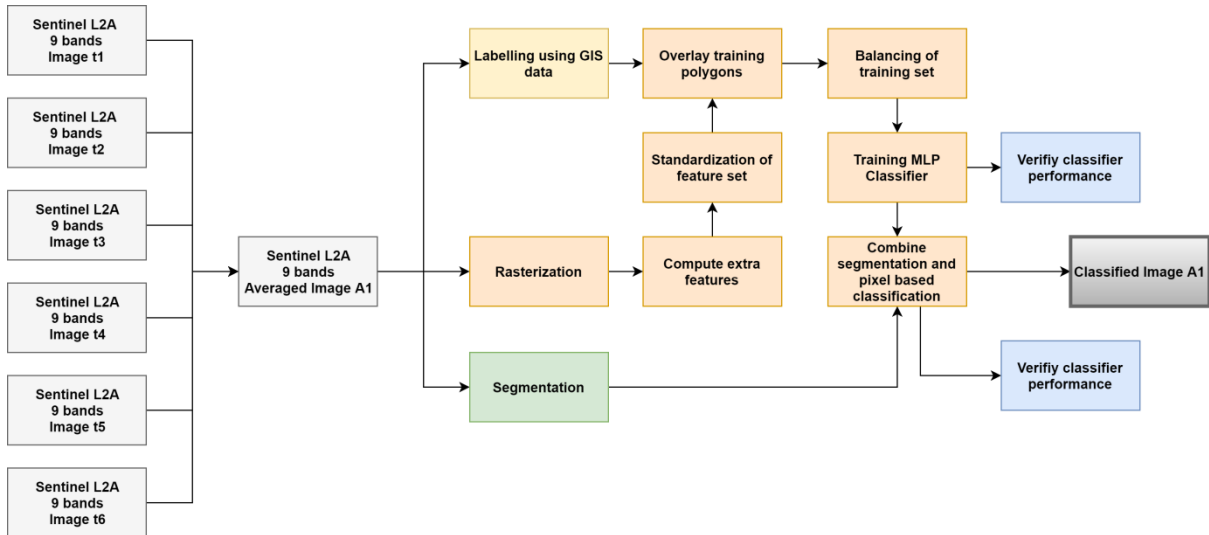


Figure 3.1: Classification Procedure

### 3.2.1 Imagery

As input ACDet uses imagery from the Sentinel-2A and Sentinel-2B satellites. Launched in respectively June 2015 (A) and March 2017 (B), the satellites fly a sun-synchronous orbit at an altitude of 786 km. With a multispectral imager on-board and a geometric revisiting time of only 5 days, the Sentinel2 satellites provide accurate and regularly updated imagery of the Earth (Kramer, 2018). The product used in this project is level-2A (L2A) imagery. The images have been corrected for geo-location and atmospherically corrected. Additionally a images with over 50% cloud coverage have been left out (ESA, 2018).

To detect and mark changes ACDet uses a combination of pixel-based classification, providing class information, and segmentation, providing geometric information. Pixel-based classification is sensitive to noise produced by temporary events in consecutive pictures (see below). Since the aim is to have ACDet identify lasting changes, the noise of temporary changes is filtered out by the segmentation procedure and by applying a moving average procedure as explained later.

Using single images to track change has several drawbacks. The classification of (a set of) pixels can reflect either a long-lasting characteristic or a temporary change. For example a semi-trailer of a standard 18.75 m length (RDW, 2012) parked on a parking lot may be visible in one individual image of a time series. In this image it may be classified as ‘building’, due to its similarity with commercial buildings seen from above. The change detection method must be able to distinguish this temporary change from the long term change linked to (dis-)appearance of a building. A second limitation of using single imagery is the influence of day specific effects. Although the Sentinel 2 satellites fly a sun-synchronous orbit there will be fluctuations in the lighting of the recordings due to local weather at the time of day. Again, ACDet must not interpret class changes that occur due to for instance differences in lighting as real changes of items.

This noise that may occur in single images can be reduced by not using as input for ACDet the single images but averages of multiple images that are taken close together in time. ACDet therefore averages the values for the spectral bands of 6 consecutive images into a new image and uses this as input for change detection. Singular events that take place on one image are removed, and for that pixel the average is taken of the remaining five images. When an event occurs in two of the six images, one image is removed and the second compromised image is included into the average calculation where it will contribute 20% of the end–result. Thus the longer the unwanted event lasts, the higher its impact on the classification procedure will be. The tipping point is when four images contain the change, one is removed, and the three remaining images contribute more than 50% to the average causing a change in classification. The noise reduction is improved further by shifting the set of six images one image at a time, a running average. Figure 3.2 shows the construction scheme for the running average. By not merely taking an averaged image, but also having them overlap, the consistency between pictures is maximized and non-lasting changes are filtered out.

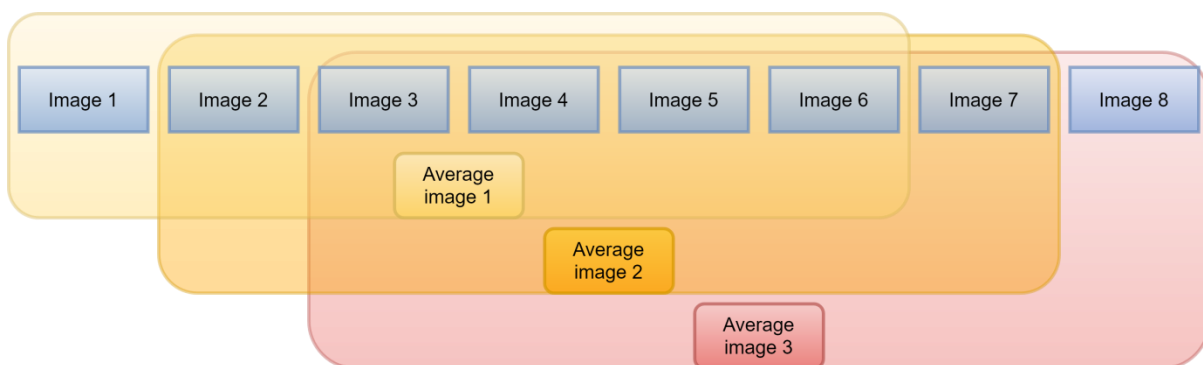


Figure 3.2: Construction of Running Average

Figure 3.3 to Figure 3.8 show 6 images being averaged into a single image (Figure 3.9). In the images there are two marked areas. The yellow marking outlines a harbour with ships in it. On every image there is a different number of ships and at different positions. This is a clear example of a temporary change that may be interesting when you want to monitor ship movements but that is not of interest for the purpose of detecting lasting changes. The averaged image shows ships only at the locations where there is an image in every pixel. Namely at the main dockings. This provides the desired information that there is a harbour with ships, but not the daily variation.

The second marked area, marked in red, outlines a grass field. Grass fields have the tendency to show a large fluctuation in spectral appearance due to weather effects and maintenance such as mowing. In Figure 3.8 the field has turned yellow, most probably due to draught, while being lashing green in the other 5 images. This temporary change is filtered out in Figure 3.9, providing a more consistent and clear input for ACDet.

The pictures have been stretched using identical values to make them visually comparable. It is clear there are noticeable colour variations between the images. These are manageable, because of the continuous training of the MLP classifier, but it is favourable for the performance of the system to minimize the colour changes. Which is also accomplished by using an averaged image.



Figure 3.3: Image 1. Sentinel 2A 2015-11-13

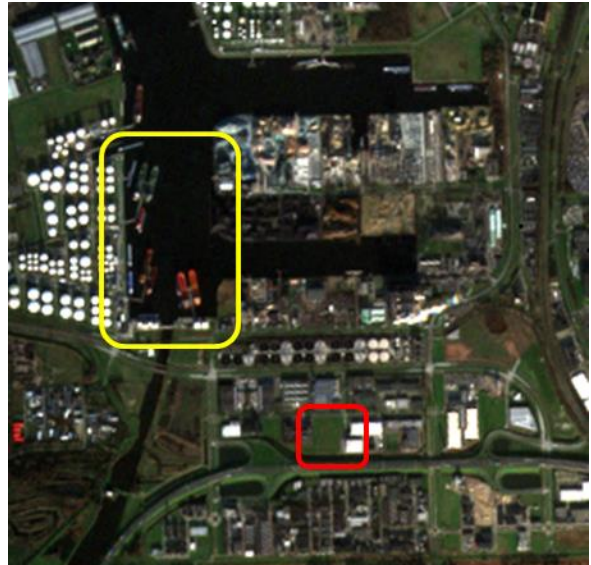


Figure 3.4: Image 2. Sentinel 2A 2015-12-23

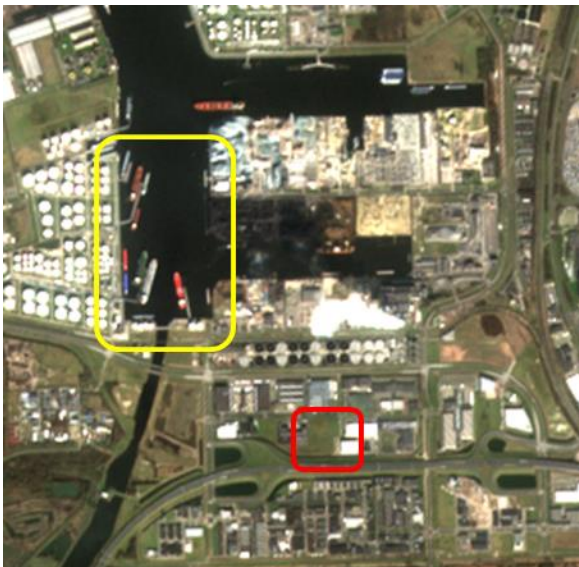


Figure 3.5: Image 3. Sentinel 2A 2016-03-12

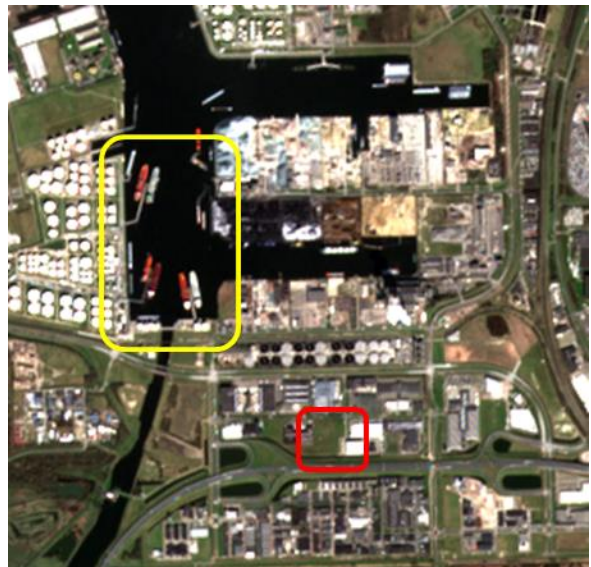


Figure 3.6: Image 4. Sentinel 2A 2016-04-01

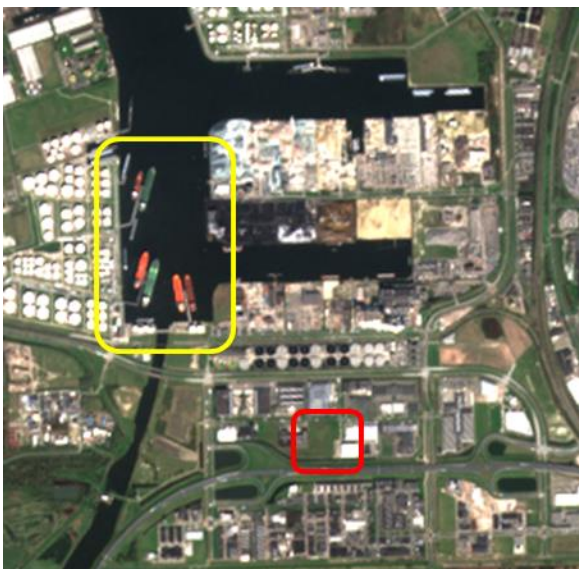


Figure 3.7: Image 5. Sentinel 2A 2016-04-11

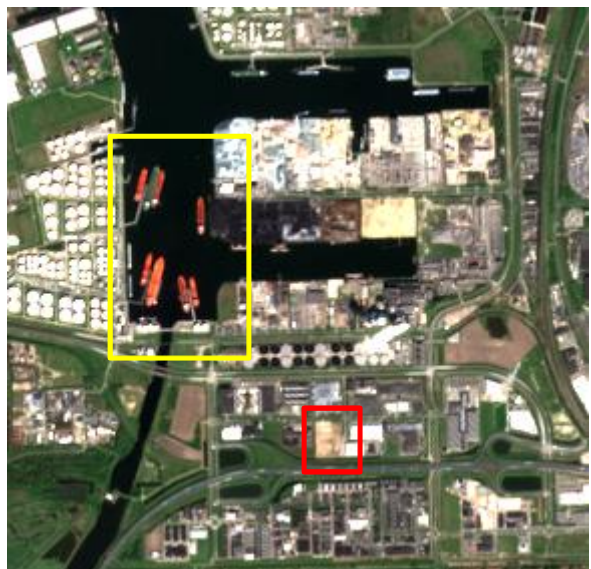


Figure 3.8: Image 6. Sentinel 2A 2016-04-21



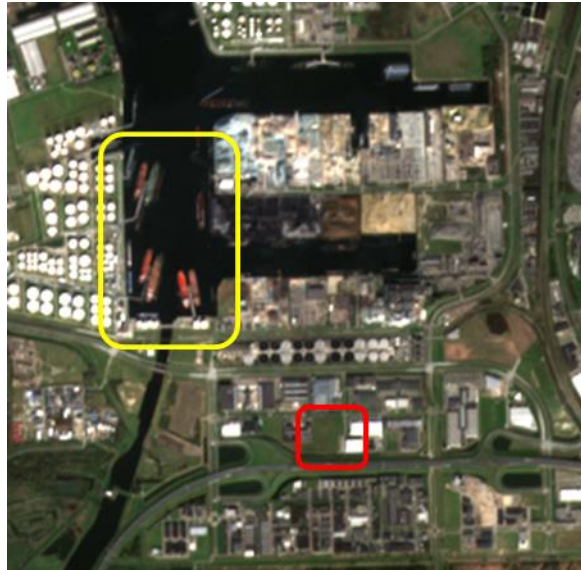


Figure 3.9: Averaged image

A major source of error in the data (the images) is the presence of clouds. Processing the images to a L2A product has already eliminated images with 50% or more cloud coverage, but clouds still present a major source of noise, causing misclassifications. There are many advanced forms of image processing for cloud detection and removal. These techniques are well established, but require significant computer resources and considerable time to implement them correctly. Next to that the commonly used cloud masks detect and remove the clouds, leaving an image partly black, which is not a desirable input of ACDet. As can be noted, using a running average also provides an option to remove clouds.

In the process of averaging the image per spectral band, statistics is used to filter out clouds. In colour coding white is  $\text{rgb}(255, 255, 255)$ . At areas where a cloud is present in one of the images there will be an outlier data point having a higher value. For example, for the R band (feature 2) 5 input pixels have a value of between 50 and 60 and a single pixel gives a value of 255 (a cloud). Using statistics this outlier can be eliminated. The 80<sup>th</sup> percentile is computed per pixel, per band. A single outlier is filtered out and the average is computed using the remaining 5 input values. If 2 or more pixels contain a cloud at exactly the same location, still only 1 cloud pixel will be removed. The cloud will therefore remain visible, but more vaguely compared to the original clouded image. The advantage of this method over more sophisticated methods, such as cloud removal by the use of a CNN, is the simplicity. It is a clear and fast approach in which only one of the six pixels will be removed if it is an outlier. And only if it is an outlier of a higher value, hence a white pixel. A CNN cloud mask is more 'smart' in detecting clouds, but will act like a black box. It is not clear which pixels have been filtered out and on what grounds that happened. As the averaged image is the input for ACDet, such lack of transparency is undesirable. A second advantage is that the mean image will not have removed, black, areas. As mentioned, most methods detect and eliminate clouds, leaving a carved image behind. Figure 3.9 and Figure 3.10 display the mean image, respectively without and with cloud removal using statistics.



Figure 3.10: Mean image without cloud removal



Figure 3.11: Mean image with cloud removal using percentile

### 3.2.2 Labelling

In machine learning labelling of the dataset is considered one of the most essential aspects (Alpaydin, 2004). It has a great influence on both the training and testing performance of the algorithm. For the labelling of satellite imagery with the purpose of detecting infrastructure the most applicable are cartographic data sources. For well mapped countries such as The Netherlands there are multiple open sources available.

To acquire a good labelling set with applicable classes a combination of Top10NL and basisregistratie gewaspercelen (BRP) has been used. Both are open data sources provided by the Dutch cadastre. Table 3.1 depicts the classes and their source. The division of classes is based on the following requirements:

- Is it of distinctive value in this research
- Does it have distinctive features
- Is it of classifiable size given the optical resolution used
- Is there a notable presence in the available satellite image area

Using these requirements, the classes were selected based on availability on datasets. Given that the classification is done on pixel level it is recommended to have classes as elementary as possible. For example sand, water and grass, rather than more high level classes such as soccer field, parking lots or airfield. The more low-level the classes, the better the classification result per pixel. Which can later on be used to distinguish objects with high a precision.

Some classes show a higher seasonal change than others. Farmland and grassland are two of the most difficult classes for this reason. They have a large seasonal change and also show overlap at moments in time. For example, a unit of farmland that is temporarily unused can have grass growing on it. Or the other way around, grassland looking like a dirt field when it has recently been ploughed and reseeded. Since the labelling provided in the independent databases is usually updated once a year, on the basis of information given by the farmers it can lead to mislabelling. To minimize this effect, the BRP for farmland has been used, since it is the most up-to-date. It is significantly more

accurate than Top10NL and OSM. The categories used for farmland are aggregated categories. Application of the large number of categories available in the BRP would not add to our aim of detecting long-lasting changes while it would increase the chance of misclassification and thus unwanted change detection.

**Table 3.1: Labelling Classes**

Class	Description	Source
<b>Water</b>	Surface water inland	– Top10NL <ul style="list-style-type: none"> <li>• Waterdeel polygon <ul style="list-style-type: none"> <li>▪ Meer</li> <li>▪ Plas</li> <li>▪ Waterloop</li> </ul> </li> </ul>
<b>Sand</b>	Sand	– Top10NL <ul style="list-style-type: none"> <li>• Terrein <ul style="list-style-type: none"> <li>▪ Zand</li> </ul> </li> <li>• Waterdeel Polygon <ul style="list-style-type: none"> <li>▪ droogvallend</li> </ul> </li> </ul>
<b>Heath</b>	Heath	– Top10NL <ul style="list-style-type: none"> <li>• Terrein <ul style="list-style-type: none"> <li>▪ Heide</li> </ul> </li> </ul>
<b>Grass</b>	Large areas of grass lands	– Top10NL <ul style="list-style-type: none"> <li>• Terrein <ul style="list-style-type: none"> <li>▪ Grasland</li> </ul> </li> </ul>
<b>Forest</b>	Areas with dense population of trees: forest of different kinds and tree plantations	– Top10NL <ul style="list-style-type: none"> <li>• Terrein <ul style="list-style-type: none"> <li>▪ Bos: naaldbos</li> <li>▪ Bos: loofbos</li> <li>▪ Bos: vriend</li> <li>▪ Bos: gemengd bos</li> </ul> </li> </ul> – BRP <ul style="list-style-type: none"> <li>• Boomkwekerij</li> <li>• Bos</li> </ul>
<b>Building</b>	Aggregated class of commercial en non-commercial buildings	– Top10NL <ul style="list-style-type: none"> <li>• Gebouw Polygon</li> </ul>
<b>Road</b>	Major high ways (A wegen) and highway junctions.	– Top10NL <ul style="list-style-type: none"> <li>• Wegdeel Polygon <ul style="list-style-type: none"> <li>▪ 1: Parkeerplaats</li> <li>▪ 1: autosnelweg</li> <li>▪ 1: hoofdweg</li> </ul> </li> </ul>
<b>Farmland</b>	Combination of aggregated farmland classes from BRP. The aggregated classes were compiled by (Noorbergen, 2018).	– BRP <ul style="list-style-type: none"> <li>• Aardappelen</li> <li>• Bieten</li> <li>• Bloemen</li> <li>• Braakliggend</li> <li>• Groenten</li> <li>• Mais</li> <li>• Overige akkerbouw</li> <li>• Peulvruchten</li> <li>• Uien</li> </ul>
<b>Sea</b>	Sea water	– Top10NL <ul style="list-style-type: none"> <li>• Waterdeel Polygon <ul style="list-style-type: none"> <li>▪ Zee</li> </ul> </li> </ul>
<b>Dune</b>	Sand dunes including vegetation	– Top10NL <ul style="list-style-type: none"> <li>• Terrein <ul style="list-style-type: none"> <li>▪ Duin</li> </ul> </li> </ul>

The labelling databases are processed into polygons using geospatial software. In this research the open source software QGIS (QGIS Development Team, 2018) was used, but other software such as ESRI is also usable. Using the Top10NL and BRP databases published by PDOK the labelling polygons for the region of the Sentinel-L2A image area are extracted and aggregated into the classes described in Table 3.1. For the rasterization of the polygons in Python the polygons are saved in the Spatialite database file format (.sqlite) (Furieri, 2017). The advantage Spatialite has over the more commonly used Shapefile format or GeoJSON format is the file size, and file structure. When it comes to a polygon file with large numbers of polygons, the file size using a Shapefile (SHP) format quickly increases. For the labelling class ‘building’, which consists of a large number of small polygons the shapefile is about 1.5 GB in size saved as 7 separate files. Whereas the same information saved as Spatialite is 29 MB saved as a single file. Saving as a GeoJSON file requires half the memory space compared to a shapefile, but still considerably more than a Spatialite file. Given the high memory use of the change detection algorithm and related data, every possible reduction in RAM-memory is needed to keep the algorithm runnable on equipment less powerful than a supercomputer.



Figure 3.12: Labelling using QGIS

### 3.2.3 Polygon Overlay

Polygons have a different data structure than pixels. A polygon is a plane figure made up of straight edges between vertices, creating a closed form. Since a polygon is mathematically described it is not bound by resolution. It can be enlarged infinitely without becoming blurry. The only element of detail a polygon has is the number of vertices. It is a fundamentally different way of storing information compared to using a raster, where every pixel is fixed and the image cannot be enlarged without becoming blurry.

To rasterize a polygon it has to be overlaid on the image. The geotransformation and projection of the Sentinel-L2A image are used to rasterize the polygon, to convert them to the identical resolution as the image. And to overlay the polygon onto the correct pixels. The polygon will not completely match the raster, thus a rendering method has to be selected in which pixels are considered as part of the polygon. Available rendering methods are:

- The pixel is completely covered by the polygon
- The polygon covers the centroid of the pixel
- The polygon touches the pixel

- The polygon covers the pixel by a specific percentage

The rendering method has an effect on the labelled training set for the classifier. Figure 3.13 shows a 9-pixel kernel of different colours with a polygon overlaid. Option 1, rasterizing a polygon onto the fully covered pixels, has a limiting effect on the training set. In Figure 3.13 this would mean only pixel 5 will be labelled by the polygon. Option two, rendering the polygon onto pixels of which the centroid is covered by the polygon is the most common rendering method. It works under the assumption that if the centroid of the pixel is covered, it catches the essence of the shape in question. Rendering by means of all pixels touched by the pixel is the most elaborate method. It is questionable whether a pixel only slightly covered, alike pixel 1, can be seen as part of the shape. Also it results in significant overlap of polygons close to each other and to pixels receiving a double label. To address this problem option four, a specific percentage of pixel covered, can be used as criteria. However, selecting a specific threshold requires research into what is optimal. Also this option is currently not supported in the GDAL library used.

For the purpose of this project an optimal balance of quantity and quality of pixels labelled is obtained by centroid covering. In areas where polygons are close to each other such as urban areas with building-class polygons there is a limited risk of pixels being double labelled. At the moment there is no researched method on selecting the proper label in those cases. The algorithm treats the last label given as the final label.

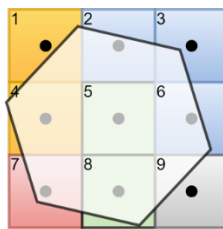


Figure 3.13: Polygon overlay

### 3.2.4 Training and Verification Set

Using the labelling procedure described it is likely the labelled data set will have one or more classes underrepresented in the number of samples compared to other classes. An imbalance in classes present in the training set can have a negative impact on the training time and quality of the classifier, the so called ‘imbalance problem’ (Prati, Batista, & Monard, 2009). More on the sensitivity of the classifier on imbalanced training sets is discussed in the paragraph Classification on page 22.

For an imbalanced dataset  $x$  with minority class:  $x_{min}$ , and majority class  $x_{maj}$ , the balancing ratio is defined as equation 3.1. In balancing the dataset is resampled into a new dataset  $x_{res}$  satisfying  $r_x > r_{x_{res}}$ . As sampling method two commonly used methods are available: random under-sampling and random over-sampling.

$$r_x = \frac{|x_{min}|}{|x_{maj}|} \quad 3.1$$

Random over-sampling is a non-heuristic method based on replicating a random selection of samples from  $x_{min}$  to reach the balancing ratio  $r_{x_{res}}$ . By creating exact copies of samples the random over-sampling method can result in overfitting of the classifier (Prati, Batista, & Monard, 2009). The method is visualized in Figure 3.14.

Random under-sampling is a non-heuristic method based on reducing the samples of  $x_{maj}$  by random selection to reach the balancing ratio  $r_{x_{res}}$ . Unlike over-sampling this method does not have the risk of over-fitting, but it does bear the risk of potentially discarding too much data. Data are important for learning (Prati, Batista, & Monard, 2009). The method is visualized in Figure 3.15.

When a single class has a significant larger presence in the dataset than the other classes overfitting of the classifier can occur. It means that the classifier if over-trained at recognizing the large class and thereby reducing the performance in detecting the other classes. To prevent over-fitting of the classifier random under-sampling is used as balancing method for the training set. The risk of disregarding too many samples is mitigated by ensuring all classes have a solid basis of labelled training pixels.

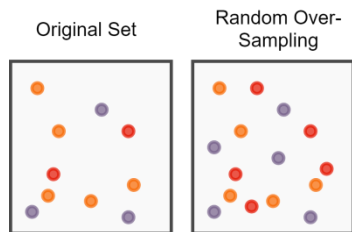


Figure 3.14: Random Over-Sampling



Figure 3.15: Random Under-Sampling

The balanced dataset is to be used for both training and verification of the classifier. To split up the samples into two separate datasets, one for training and one for verification, a method of random splitting is used. By random selection 33% of the samples is set aside for verification of the trained classifier.

### 3.2.5 Classifier

Classification is the statistical problem of assigning categories (classes) to data points, in this case pixels. The algorithm responsible for this is called the classifier. In ACDet the classifier is used to assign classes at pixel level. From this the following requirements are set to the method of classification:

- Handle separate pixels as input
- Multi-class classification
- High accuracy

For multi-class classification with a high accuracy there are two classification methods available: a random forest (RF) or a neural network (NN) (Azure Machine Learning Team, 2015). A random forest is a collection of decision trees. Each tree categorizes a pixel based on a sub-set of features. Every tree uses a different feature sub-set. From the collection of outcomes the majority is taken as final classification result for the input pixel. The advantages of using a random forest are:

- Fast training times
- Decision process is transparent
- Not prone to overfitting
- High accuracy
- Resource

Compared to a RF a NN has longer training times, is more prone to overfitting and the decision process is not transparent. However a NN can achieve higher accuracies, and most importantly it is capable of continuous learning, where a RF is not. Continuous learning is the process of using new inputs (a new picture) to continue training and improve the model. A RF is not capable of continuous learning, meaning the classifier would have to be retrained on every picture. This increases processing time and it makes the programme incapable of handling seasonal change.

Because of the continuous training capabilities a NN is used as classification method in ACDet. The most commonly used and most applicable NN types are a Convolutional Neural Network (CNN) and Multi-Layer Perceptron (MLP) NN. CNN's are most spoken of these days in the field of computer vision and know many applications. From recognizing brain tumours to facial recognition. CNN's are especially well suited for image classification and can achieve superhuman performance (He, Zhang, Ren, & Sun, 2015). The essence and of a CNN is that it trains itself with using merely labelled pictures as input. The disadvantage is that it will act as a black box. The only adjustments that can be made are to the design of the network in terms of number and size of layers. Since the technique is very new there are no studies yet on the fundamental workings of a CNN. Designing and tweaking a CNN network is done solely based on empirical data.

A second point of attention is the significant computing resources a CNN requires for designing and training. Companies with significant resources such as Deep Mind and CrowdAI dedicate all their efforts and resources in designing the optimal CNN for a specific task. The fact that it acts as a black box, the lack of fundamental design theory and the required computing resources makes it an unsuitable technique for this research.

The most basic form of NN is the so called multi-layer perceptron (MLP) network. Just as a CNN the decision process is not transparent, but unlike a CNN it only handles the statistical part of the classification process, making it less of a black box. The working of the system and its integration into ACDet are simple. This leaves more time in the project to focus on the performance of ACDet as a whole by designing the separate elements and not be dependent on a comprehensive CNN, with only parameter tweaking as design tools.

The MLP NN consists of a minimum of three layers: the input layer, hidden layer, and output layer, as visualized in Figure 3.16. The number of nodes in the input layer equals the number of features, in this example three. Each node is connected to every node in the next layer, the hidden layer, by an edge with a certain weight. The number of nodes in the hidden layer is configurable, and unrelated to the number of feature or classes. Next to that the number of hidden layers is adjustable. The nodes in the hidden layer are again connected to each node in the output layer. The output layer consists of a number of nodes equal to the number of classes, in this example 4.

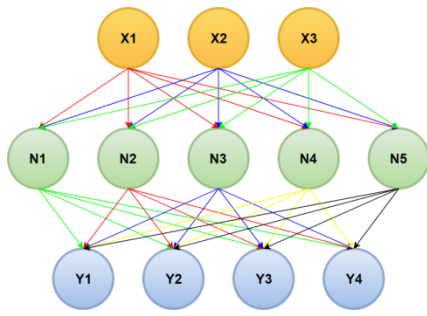


Figure 3.16: Layers and nodes of MLP network

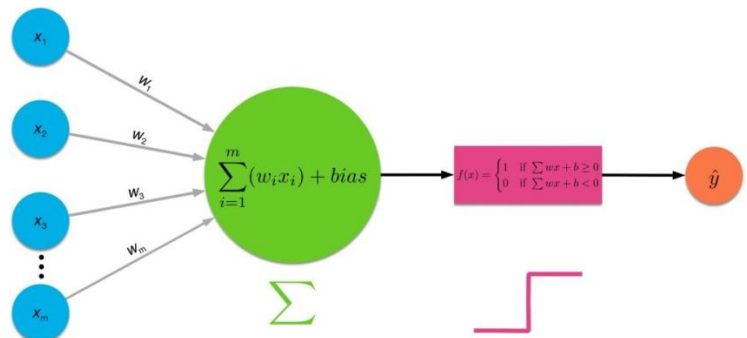


Figure 3.17: Classification procedure of MLP network (Kang, 2018)

Figure 3.17 shows the procedure of transfer and manipulation of a data point running through the network. At each node the inputs are multiplied with the corresponding edge weights and summed including a bias. Using an activation function the output is generated, which is sent to every node of the next layer.

The activation function is at the heart of a node, transforming the inputs to an output. The most common functions used are (SciKit-Learn, 2018, pp. 1873-1878):

- Identity: a no operation function, useful in implementing a linear bottleneck.

$$f(x) = x \tag{3.2}$$

- Logistic: a logistic sigmoid function.

$$f(x) = 1/(1 + \exp(-x)) \tag{3.3}$$

- Tanh: a hyperbolic tan function

$$f(x) = \tanh(x) \tag{3.4}$$

- ReLu: the rectified linear unit function.

$$f(x) = \max(0, x) \tag{3.5}$$

In the newest neural networks the ReLu function is considered most applicable in computer vision. The function, shown in Figure 3.18, uses a zero threshold to activate the signal. Due to its linear, non-saturating form it improves training times by a factor 6, compared to the tanh activation function, when using stochastic gradient descent as training solver (Karpathy, 2018), see Figure 3.19. For neurons involving heavy operations such as exponentials, a ReLu function is much more efficient than either tanh or sigmoid. The thresholds can be set to zero using a matrix, greatly simplifying the process (Karpathy, 2018). A drawback of the ReLu function is its sensitivity. During training when the gradient is too steep, the weight of the edge will be changed significantly, making that that node will not be triggered (activated) at any data point anymore after it.



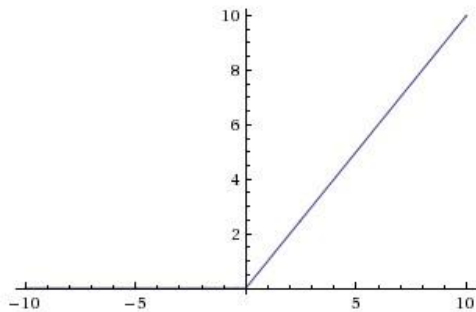


Figure 3.18: ReLu activation function (Karpathy, 2018)

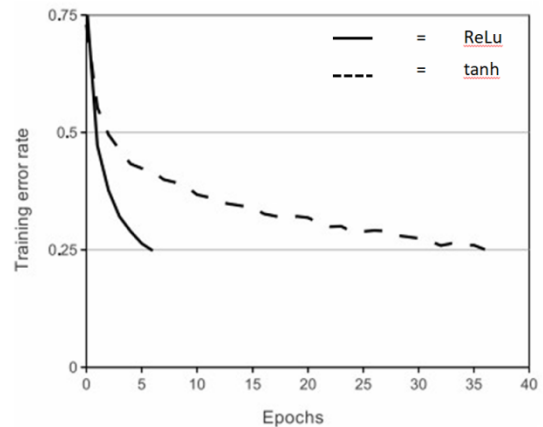


Figure 3.19: Training convergence using stochastic gradient descent (Krizhevsky, Sutskever, & Hinton)

A neural network uses supervised learning to adjust the edge weights. The labelled training set of pixels is classified by the MLP classifier, and the outcome is compared to the pre-defined label. Using an 'adam' solver the weights are adjusted accordingly. It is a computationally efficient first-order gradient descent optimizer of stochastic objective functions. It is especially well suited for large datasets and high-dimensional features spaces (Kingma & Jimmy, 2017).

The representational power of NN is determined by the depth, the number of hidden layers, and the width, the number of nodes per layer. The combination of the two controls the ability to represent complex, higher order, functions, hence the capacity of the network. As each pixel in a layer is connected to every pixel in the next layer, an increase in network depth significantly increases the networks ability to represent complex higher order functions. Optimizing the depth and width of a network remain unsolved problems. Networks design is done purely based on empirical data (Shuxiang & Ling, 2008). In general a single hidden layer has enough solving capability. Next to that an increase in network depth increases the risk of overfitting, as the model tries to find highly complex functions to describe the data. Also, a depth increase significantly increases the training time of the model. Taking into account these factors, a single hidden layer is used in ACDet.

Remains the design of the layer width. Figure 3.20 visualizes the effect of adding additional neurons when describing a two-class data set. A wider network creates more decision regions to describe the data. An increase in width comes with an increased risk in overfitting and increased training time. As rule of thumb the optimal layer width is found to be the mean of the input and output layer widths (Heaton, 2012).

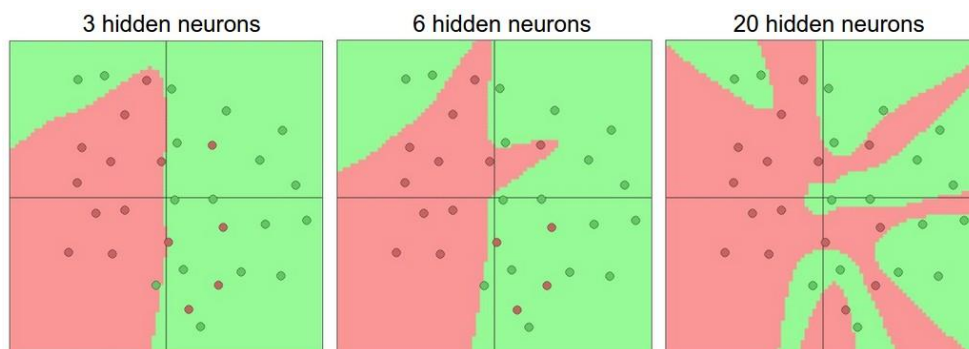


Figure 3.20: NN Layer design - effect of layer size (Karpathy, 2018)

The MLP classifier computes per pixel the probability of it belonging to each class. The highest probability is chosen as the outcome, and thus the classification of that pixel. In certain cases the algorithm has a high uncertainty. When the features of a pixel have a large overlap with two or even more classes the likelihood per class will go down. By default the algorithm is designed to select a class, even with low likelihoods and two classes close to each other in terms of probability. To address this issue of a forced classification in cases of high uncertainty a threshold is applied. When the likelihood of any class is below 50% the pixel will not be categorized into a class. This significantly improves the reliability of the classification system as it is no longer forced to assign a class but at the cost of losing some pixels.

For implementation of the MLP classifier the SciKit Machine Learning Toolbox for Python is used. It is one of the most commonly used Python libraries for machine learning. The high level API enables a fast and powerful implementation of the most commonly used machine learning algorithms.

### 3.3 Feature Engineering

Every pixel is treated as a data point with  $x$  features (characteristics). The first nine features are the spectral bands of the Sentinel-L2A product. Ranging from the visible light spectrum up to the short wave infrared spectrum, the bands give solid feature sets for the recorded area. The exact wavelengths as well as the resolution per band are depicted in Table 3.2.

A direct shortcoming of pixel level classification is the lack of contextual information. One of the effects visible from this is the misclassification of a single pixel in a class homogeneous area. For example, a pixel in the middle of a grass field being classified as farmland when it shows more brown tints. To add contextual information to pixel level classification smoothing kernels are used. The kernel is applied on a single image band. Using kernels on all nine spectral bands would result in pollution of the feature set. Therefore, kernels are applied to the first three spectral bands: blue (490 nm), green (560 nm) and red (665 nm). A combination of 3x3 kernels and 5x5 kernels is used. The 3x3 and 5x5 kernel are visualized in Figure 3.21 and Figure 3.22 respectively. The smaller kernels capture direct contextual information to solve single pixel misclassifications such as the farmland pixel in a grass field. Whereas the larger 5x5 kernels are used to take the area into account in the classification. Which is especially useful for classes that are difficult to detect on pixel level, but stand out as an area. For example, dunes and heath.

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

Figure 3.21: 3x3 Contextual Kernel

1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25
1/25	1/25	1/25	1/25	1/25

Figure 3.22: 5x5 Contextual Kernel

Table 3.2: Features

Feature (f)	Description	Resolution [m]	Equation
0	Sentinel Band 2 - 490 nm (visible)	10 x 10	Na
1	Sentinel Band 3 - 560 nm (visible)	10 x 10	Na
2	Sentinel Band 4 - 665 nm (visible)	10 x 10	Na
3	Sentinel Band 5 - 705 nm (VNIR)	20 x 20	Na
4	Sentinel Band 6 - 740 nm (VNIR)	20 x 20	Na
5	Sentinel Band 7 - 783 nm (VNIR)	20 x 20	Na
6	Sentinel Band 8 - 842 nm (VNIR)	10 x 10	Na
7	Sentinel Band 11 - 1610 nm (SWIR)	20 x 20	Na
8	Sentinel Band 12 - 2190 nm (SWIR)	20 x 20	Na
9	3x3 kernel feature 0	30 x 30	$x_{3x3} = \frac{1}{9}(x_1 + x_2 + \dots + x_9)$
10	3x3 kernel feature 1	30 x 30	$x_{3x3} = \frac{1}{9}(x_1 + x_2 + \dots + x_9)$
11	3x3 kernel feature 2	30 x 30	$x_{3x3} = \frac{1}{9}(x_1 + x_2 + \dots + x_9)$
12	5x5 kernel feature 0	50 x 50	$x_{5x5} = \frac{1}{25}(x_1 + x_2 + \dots + x_{25})$
13	5x5 kernel feature 1	50 x 50	$x_{5x5} = \frac{1}{25}(x_1 + x_2 + \dots + x_{25})$
14	5x5 kernel feature 2	50 x 50	$x_{5x5} = \frac{1}{25}(x_1 + x_2 + \dots + x_{25})$
15	NDWI (Normalized Difference Water Index)	10 x 10	$x_{NDWI} = (B_3 - B_8)/(B_3 + B_8)$
16	NDVI (Normalized Difference Vegetation Index)	10 x 10	$x_{NDVI} = (B_8 - B_4)/(B_8 + B_4)$
17	MNDWI (Modified Normalized Water Difference Index)	20 x 20	$x_{MNDWI} = (B_3 - B_{11})/(B_3 + B_{11})$
18	SAVI (Soil Adjusted Vegetation Index)	10 x 10	$x_{SAVI} = \frac{3}{2} \frac{B_8 - B_4}{B_8 + B_4 + 1/2}$
19	NDVI705 (Red Edge Normalized Difference Vegetation Index)	20 x 20	$x_{NDVI705} = (B_6 - B_5)/(B_6 + B_5)$
20	LAI-SAVI (Leaf Area Index - Soil Adjusted Vegetation Index)	10 x 10	$x_{LAI-SAVI} = \frac{1}{2.4} \ln \left[ 0.371 + \frac{3}{2} \frac{B_8 - B_4}{B_8 + B_4 + 1/2} \right]$

The classifier uses features to distinguish between classes. A good feature is therefore one with a good class distribution. A distinctive feature increases the performance of the classifier. Using the labelled training pixels, the class distribution of a feature in the dataset can be visualized, giving an

insight in the effectiveness of a feature. In Figure 3.23 the class distribution of feature 4 is shown. The figure shows the minimum and maximum data values (the thin line), the standard deviation (the thick line) and the mean (the dot) per class. The standard deviation is of most value to characterize the distinctive value of a feature for a particular class. Inspecting Figure 3.23 it is noticeable that the features ‘sand’ and ‘grass’ have a standard deviation higher than the other features. On the basis of this feature alone the classifier will determine a pixel with a feature 4 value of 3000 nm to be either ‘sand’ or ‘grass’. In combination with other features the certainty per class is adjusted and one class will stand out as most probable.

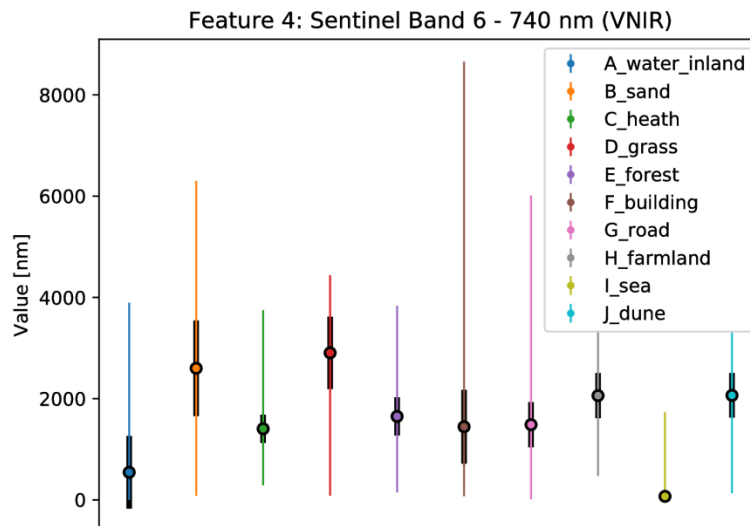


Figure 3.23: Class distribution feature 4

To increase the classification accuracy specific band combinations are used to highlight elements, for example vegetation. These so called indices make very effective features as they are designed to highlight particular classes. From the set of available indices (SINERGISE, 2018) the most effective features for the data set at hand were selected using the feature distribution visualization.

The normalized difference water index (NDWI) (feature 15) is designed for water mapping (Figure 3.24). In the visible to infrared wavelengths water has a low radiation and a high absorbability. The green (Sentinel band 3) and NIR band (Sentinel band 8) are used to highlight water areas, see Equation 3.6. The feature distribution (Appendix 0: Figure 0.16) visualizes the effectiveness of the NDWI in mapping water. Next to that it is a distinctive feature for dunes.

$$x_{NDWI} = (B_3 - B_8) / (B_3 + B_8) \quad 3.6$$



Figure 3.24: NDWI (SINERGISE, 2018)

The normalized difference vegetation index (NDVI) (feature 16) is an effective index for mapping green vegetation (Figure 3.25) by normalizing chlorophyll absorption in the red wavelength (Sentinel band 4) and green leaf scattering in the NIR wavelength (SINERGISE, 2018), see Equation 3.7. The feature distribution (Appendix 0: Figure 0.17) underlines the distinctive value of the NDVI in highlighting the classes ‘heath’, ‘grass’, and ‘forest’.

$$x_{NDVI} = (B_8 - B_4)/(B_8 + B_4) \quad 3.7$$

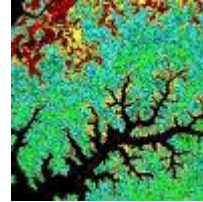


Figure 3.25: NDVI (SINERGISE, 2018)

A variation on the NDWI is the modified normalized difference water index (MNDWI) (feature 17), visualized in Figure 3.26. The NIR band (Sentinel band 8) used in the NDWI is replaced by the middle infrared band (Sentinel band 11) and the green band (Sentinel band 3) is substituted for the red band (Sentinel band 4), see Equation 3.8. As a result the MNDWI enhances open water and reduces the noise from built-up land, vegetation and soil (Xu, 2006). The class distribution of the MNDWI plotted in Figure 0.18 (Appendix 0) highlights the classes ‘water inland’ and ‘sea’ the same as the NDWI does, but it also shows distinctive values for the classes ‘heath’, ‘grass’, and ‘forest’. Whereas these classes are overlapping in distribution for the NDWI.

$$x_{MNDWI} = (B_3 - B_{11})/(B_3 + B_{11}) \quad 3.8$$



Figure 3.26: MNDWI (SINERGISE, 2018)

The NDVI index utilizes the red and NIR wavelengths to map canopy reflectance. Although it is found to map vegetation very well there are limitations due to atmospheric influences and soil substrate differences. As a result the NDVI shows unstable results in mapping areas of dense vegetation. The soil adjusted vegetation index (SAVI) (feature 18) is an alteration on the NDVI, minimizing the influence of soil brightness from spectral vegetation (Huete, 1988).

$$x_{SAVI} = \frac{3}{2} \frac{B_8 - B_4}{B_8 + B_4 + 1/2} \quad 3.9$$

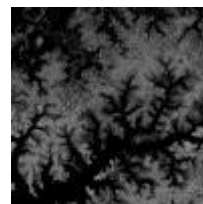


Figure 3.27: SAVI (SINERGISE, 2018)

The red edge normalized difference vegetation index (NDVI705) (feature 19) has been developed as a measure for the health of vegetation and cover quality. It is a narrowband index using the ratio of NIR and red-edge wavelengths as depicted in Equation 3.10. Compared to the NDVI the NDVI705 is less sensitive to viewing angle (Verrelst, Koetz, Kneubühler, & Schaepman, 2006). For dikes, dunes and heath cover quality is determined by the presence of healthy grass, dry grass and bare soil. Dense green grass cover effects the reflectance spectra by showing an order of magnitude increase in the NIR wavelength, an increase in the red wavelength and a decrease in the red-edge wavelength. Bare soil effects the complete reflectance spectrum. The combination of bare soil and

lush green vegetation makes the NDVI705 an ideal indicator for cover quality (Cundill, der van Werff, & der van Meijde, 2015). And thus an ideal feature for combined classes such as ‘heath’ and ‘dune’.

$$x_{NDVI705} = (B_6 - B_5)/(B_6 + B_5) \quad 3.10$$

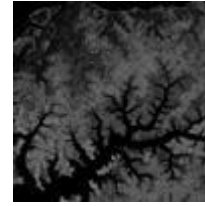


Figure 3.28: NDVI705 (SINERGISE, 2018)

The final index used in the pixel classification process is the leaf area index – soil adjusted vegetation index (LAI-SAVI) (feature 20). The leaf area index is the projected leaf area per unit of ground surface. It has been found to closely correlate to vegetation productivity and evapotranspiration (the sum of evaporation and plant transpiration from land and water into the atmosphere). At a regional level the LAI can be accurately estimated by the SAVI (Xavier & Vettorazzi, 2004), using Equation 3.11.

$$x_{LAI-SAVI} = \frac{1}{2.4} \ln \left[ 0.371 + \frac{3}{2} \frac{B_8 - B_4}{B_8 + B_4 + 1/2} \right] \quad 3.11$$

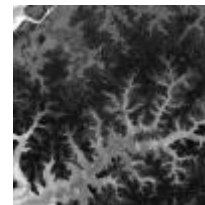


Figure 3.29: LAI-SAVI (SINERGISE, 2018)

### 3.3.1 Standardization

The MLP classifier is sensitive for absolute feature values. It does not incorporate scaling of a sort by itself, so the feature set has to be scaled before being used as input. For development purposes and to ensure compatibility with the classifier, the standardization method is chosen from the selection of methods provided by the pre-processing toolbox of the SciKit-Learn library.

Based on the data set at hand the standardization method has to be insensitive to outliers, preserve class balance and preserve feature distribution. The most compliant technique is a robust scaler. The centering and scaling statistics is based on percentiles, alike the cloud removal approach in the image averaging. It makes that the scaler is not affected by marginal outliers, even if they appear in large numbers (Raghav, Lemaitre, & Unterthiner, 2017). The capability to handle outliers is of importance to have the ACDet program as automated and flexible as possible. If ACDet were to be over sensitive to outliers, non-useful features or other irregularities, it would dramatically decrease the general applicability of the programme.

Preserving feature distribution is an important property. A flattened feature space would make it difficult for the classifier to distinguish between classes, making the accuracy of ACDet go down. Figure 3.30 shows a data set including outliers with values significantly higher than the average data. On the right plot the robust scaler has removed the outliers, while preserving the original feature distribution. The resulting range of features is significantly higher using the robust scaler compared to other scaling methods such as normalization, min-max-scaling, and max-absolute scaling.

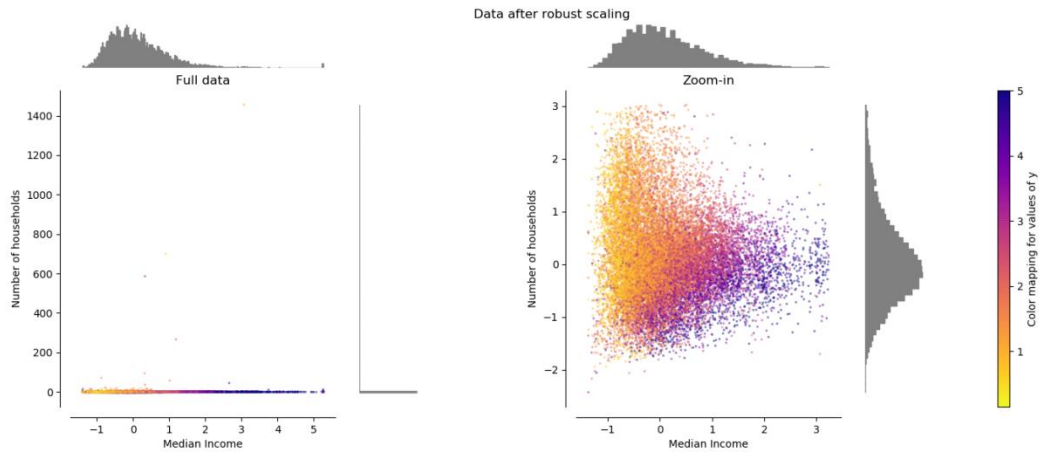


Figure 3.30: Robust Scaler (Raghav, Lemaitre, & Unterthiner, 2017)

In Figure 3.31 the effect is shown of using a robust scaler standardization on feature 4, the same feature as previously shown in Figure 3.23. Comparing the graphs underlines the ability of the robust scaling method to preserve the original feature distribution.

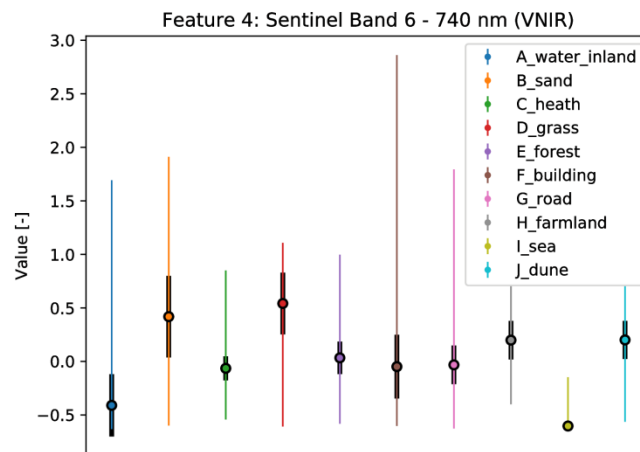


Figure 3.31: Class distribution feature 4 Standardized

### 3.4 Segmentation

To move from a pixel based classification to an object based system the geometry of objects has to be taken into account. A common method is by means of image segmentation. Pixels are grouped based on characteristics, lifting out particular elements in a picture. Most segmentation methods focus on highlighting a specific single element against the background, for example a picture of coins on a sheet (SciKit-Image, 2014). For the application in this research the segmentation method will have to fulfil the following requirements:

1. Capture perceptually important groupings of pixels.
2. Capable of defining relevant regions with a large variation in size. For example, het IJsselmeer is considered a single region, but a building of about 9 pixels should also be marked as a separate region.



3. Provide a consistent output in a time series of images. The segmentation method will be used in this research to determine 'objects' and is it to quantify change, creation, or removal of objects in a time series of images. It is therefore important to have a consistent segmentation output in the time series.
4. Computationally efficient. Segmenting the image is only of the steps in the entire process. Next to that the images at hand are of a significant size (10980\*10980 pixels). A computationally efficient segmentation method is therefore a requirement to keep the program workable.
5. Cope with multi-band images, to leverage the full potential of the Sentinel data.

The definition of a perceptually important region depends on the application of the technology. In this research it is a region that would be marked as a single distinctive region by a human operator analysing the satellite image. Figure 3.32 displays a synthetic image with different regions. An operator would mark the original image to have three regions. Most standard segmentation methods such as edge detection or threshold segmentation (Yuheng & Hao, 2017) will give an incorrect segmentation result of two regions. To return the correct three regions the segmentation method should be able to cope with large variations in intensities and use non-local criteria to distinguish perceptually important edges (Felzenszwalb & Huttenlocher, 2004).

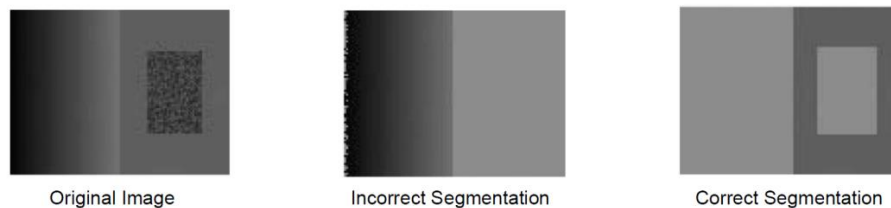


Figure 3.32: Perceptually important regions in an arbitrary grayscale image (Felzenszwalb & Huttenlocher, 2004)

The five most common classes of image segmentation methods are: threshold segmentation, regional growth segmentation, edge detection segmentation, segmentation based on clustering, and segmentation based on weakly supervised learning in CNN (Yuheng & Hao, 2017). As mentioned thresholding and edge detection segmentation are not capable of handling large variations in intensities. Next to that they only work with grayscale images. In the time series the Sentinel images show significant colour variations based on local weather and season at the time of measurement. To leverage the full potential of the data and minimize the seasonal and weather effects the segmentation method should be able to handle multi-band images, which excludes edge detection and thresholding. Segmentation by means of clustering can be done with many different clustering methods. In this research the methods K-means and DBSCAN have been used and analysed on performance. Both were unable to capture perceptually important regions and required computational resources that were not available for this research, thus clustering is deemed unfit as segmentation method. Segmentation based on weakly-supervised learning using a CNN is also not applicable for this research. As described in {section classification method} the approach is too comprehensive, acts as a black-box and requires significant computational resources.

By exclusion of the other four options, this leaves segmentation by regional growth. In this approach neighbouring pixels with similar properties are grouped into regions. The graph-based image

segmentation method developed by Felzenszwalb and Huttenlocher in 2004 is a perfect fit for the change detection problem at hand.

In the graph based segmentation method developed by Felzenszwalb and Huttenlocher the pixels, from here on referred to as vertices, are represented as a 2-dimensional nearest neighbour graph,  $G = (V, E)$ . Each vertex  $v_i \in V$  corresponds to a pixel, and E is a set of undirected edges (lines) between pairs of neighbouring vertices in the Euclidian feature space. Each edge has a weight, where  $w(v_i, v_j)$  is the edge weight between vertices  $v_i$  and  $v_j$ . The weight of an edge is defined as the Euclidian distance between the vertex pair (Equation 3.12). S is the segmentation of G such that  $G' = (V, E')$ . S consists of a set of components (regions) C.

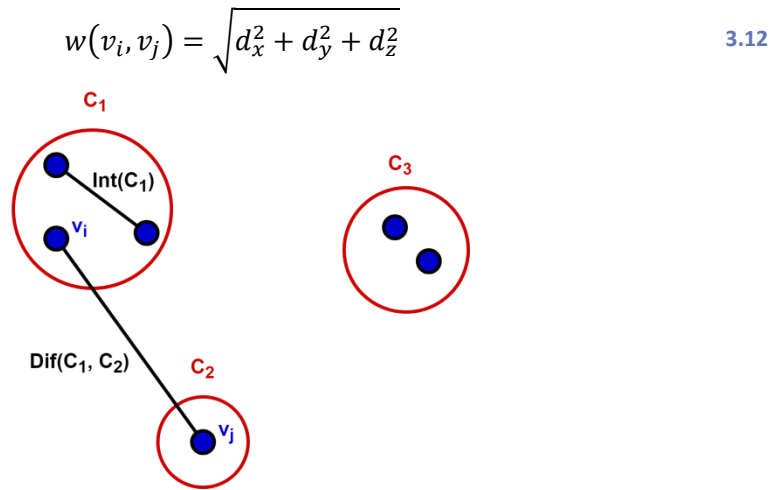


Figure 3.33: Set of components consisting of a single vertex each

Figure 3.33 displays a set of five vertices (V), with edges E, divided into three components (C). Using predicate D (Equation 3.13), the boundaries between components are evaluated. The predicate describes a comparison between the internal difference of a component and the length of the edge to the next component.

$$D(C_1, C_2) = \begin{cases} true & \text{if } Dif(C_1, C_2) > MInt(C_1, C_2) \\ false & \text{otherwise} \end{cases} \tag{3.13}$$

Using a minimum spanning tree (MST) the internal difference is set to be the largest weight (edge) in the MST of a component  $C \subseteq V$ , see Equation 3.14. Combined with the threshold (Equation 3.15) the minimal internal difference is defined as Equation 3.16, where the threshold controls the minimal difference between two components to count as a new component.

$$Int(C) = \max_{e \in MST(C,E)} w(e) \tag{3.14}$$

Where  $w(e)$  is the weight of edge  $e$ .

$$\tau(C) = k / |C| \tag{3.15}$$

Where  $k$  = scale of observation (predefined).

$$MInt(C_1, C_2) = \min(Int(C_1) + \tau(C_1), Int(C_2) + \tau(C_2)) \tag{3.16}$$

Secondly the difference between a pair of components  $C_1, C_2 \subseteq V$  is defined as the minimum weight edge connecting the two components: Equation 3.17. If there is not edge connecting  $C_1$  and  $C_2$  the

$Dif(C_1, C_2)$  is infinite. Looking back at predicate formula D, a vertex is defined as a new component if the external difference is greater than the minimal internal difference. If that is not the case, and the minimum internal difference is greater, the vertex is included in the component  $C_1$ .

$$Dif(C_1, C_2) = \min_{v_i \in C_1, v_j \in C_2, (v_i, v_j) \in E} w(v_i, v_j) \quad 3.17$$

Using the predicate the segmentation algorithm works in the following steps:

**Algorithm 1: Segmentation**

- *Input:*  $G = (V, E)$ , with  $n$  vertices and  $m$  edges.
  - *Predefine:*  $k$
0. Sort( $E$ ) by increasing edge weight into  $a = (e_1, \dots, e_m)$
  1. Initialize segmentation  $S^0$ . Each vertex  $v_i$  is a standalone component.
  2. Loop over  $q$  in range (1,m):
    - a. Edge pair  $e_q = (v_i, v_j)$
    - b. Define components  $S^{q-1}(v_i, v_j) = (C_i^{q-1}, C_j^{1-q})$
    - c. Compute internal differences  $Int(C_i^{q-1})$  and  $Int(C_j^{q-1})$  using equation 3.14.
    - d. Compute thresholds  $\tau(C_i^{q-1})$  and  $\tau(C_j^{q-1})$  using equation 3.15.
    - e. Compute minimal internal difference using equation 3.16.
    - f. Compute the minimum weight difference between the components  $C_i^{q-1}$  and  $C_j^{q-1}$  using equation 3.17.
    - g. Apply predicate (equation 3.13). If true  $C_i^{q-1}$  and  $C_j^{q-1}$  are merged into  $S^q$ , hence the segment grows. Otherwise  $S^q = S^{q-1}$ , hence a new segment is created.
  3. Return final segmentation  $S^m$

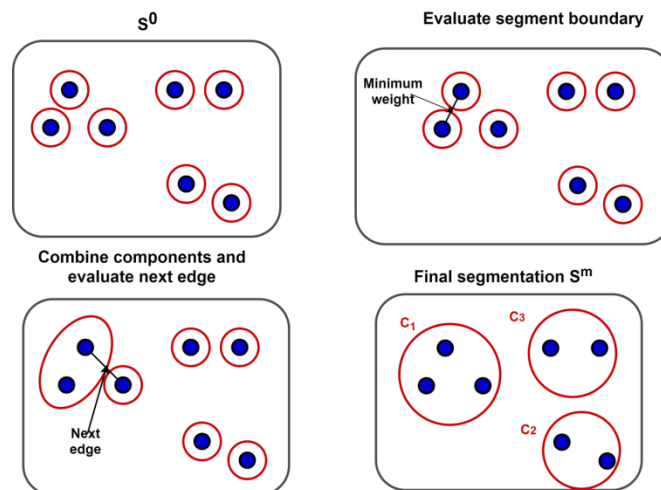


Figure 3.34: Segmentation steps

Figure 3.34 depicts a visualization of the segmentation process. Using local decision criteria, ACDet decides to either merge two components or create a boundary. This is called a greedy process, and it makes the method applicable to large images, as the algorithm works its way through the image,

without having to include the complete image in its computations. A downside of greedy processes is that global properties of the image are not taken into account. The unique element of the method at hand is that it is greedy while still obeying to global, perceptually important properties. This is done using predicate  $D$  as described. The proof that the algorithm produces segments that are neither too coarse, nor too fine is given in the paper by Felzenswalb and Huttenlocher.

The ability to handle multi-band imagery is another unique advantage the Felzenswalb method has over other segmentation methods, as it allows to leverage the full potential of the Sentinel imagery and reduce the seasonal and day effects of an image a time-series of images.

### 3.5 Classification Cleaning

In the pixel-based classification a percentage of the pixels is misclassified while being surrounded by pixels of a uniform class. This happens mainly at areas with a high percentage of polluted pixels, hence urban areas. A pixel is considered polluted when it is on the edge of two or more different classes, resulting in a mixed-colour pixel.

Under the assumption that a segment is homogeneous in class, the segmentation information and pixel based class map can be merged. The segmentation is performed purely on spectral information. Using an Euclidean feature space the data points are grouped based on relative distance, as described in section 3.4 on segmentation. Assuming the classes are not considerably overlapping, which is ensured by the feature engineering, a segment is by definition homogeneous in class.

With a mathematically simple procedure the pixel based class map and segmented image are merged. The pixel classes present in a segment are counted, and the majority class is considered as most applicable class. The pixels belonging to that segment are relabelled to the single class, resulting in a class homogeneous segment.

The process of homogenizing segments is called classification cleaning. It is an automated process to increase the classification, and thereby the performance of ACDet, without the need for extra information input. The fact that it is an automatic procedure with information already present in the system, makes it highly useful.

Figure 3.35 shows the image of a highway traffic junction, encircled in yellow. The traffic junction is an element of considerable size, but due to the low spatial resolution and the geometric shape of the junction there are several mixed pixels. The pixel based class map of the area is shown in Figure 3.36. The junction is recognisable, but the road has shrunk to a narrow line. The water present at both sides creates mixed pixels, which in this case are predominantly water and thus classed as such. Secondly there is a percentage of unclassified (white) pixels. With such a mixed result it is questionable if the system actually interprets it correctly or presents noise that appears correct.

Using the procedure of classification cleaning the image is homogenized using the segmental information, resulting in Figure 3.37. It is a much cleaner output. The marked traffic junction is labelled as unknown by ACDet in this class map. An outcome that appears to be a loss of information, but is actually an improvement as it is a reliable outcome.



Figure 3.35: Original picture M1

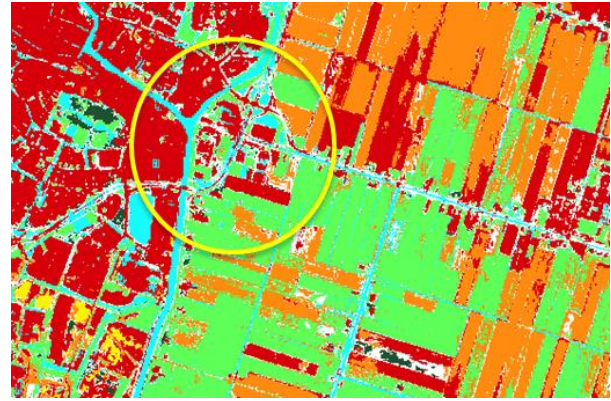


Figure 3.36: Pixel based classification

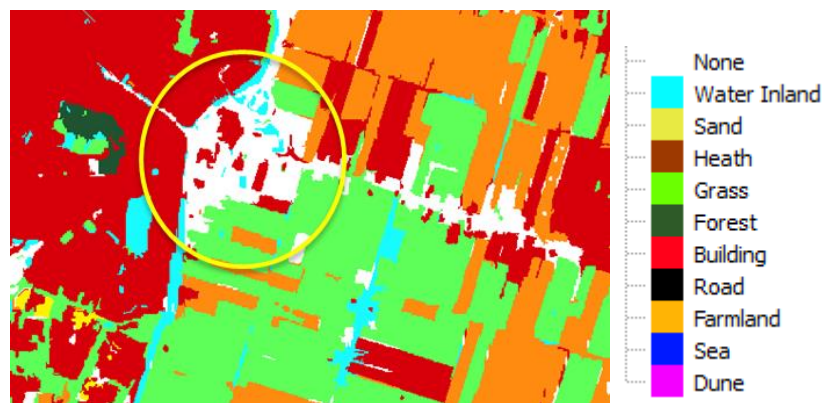


Figure 3.37: Cleaned classification

Relabelling part of the pixels can be considered as a loss of detail. It is however questionable if the detail lost was accurate. The input images have a spatial resolution of 10x10 m per pixel. A considerable size when it comes to detecting urban objects. As a result of the low resolution the urban areas, as well as others, show a great variety in pixel classes. This is due to for example a stroke of grass in a backyard, or the central part of a road. Features that appear as recognisable to an operator, but which are actually too small to be seen on the image, given the resolution. These small features create polluted pixels. The fact that an operator is still able to distinguish between a building and the backyard is due to external information, not present in the picture. The operator knows he or she is looking at a house in a city and the pixel is slightly green. The chance of that green spot being a stroke of grass is significant due to the extra information the operator has. This additional information is however not present in the image or the ACDet system. Solely based on the spectral information it would be impossible to distinguish between the building and the backyard due to the low resolution. It is undesirable to have the system try this. Also because the training and verification polygons do not have the accuracy to distinguish between the building and a small size backyard. Homogenizing the class map is therefore an improvement in the performance of ACDet and not a loss of detail.

### 3.6 Change Detection

The cleaned class maps provide class information on a pixel level per single image. To detect and quantify change consecutive images in the time series of images are compared. The procedure of change detection is portrayed in Figure 3.38. Two separate elements of change are determined: change in class and change in objects.

The change in class is determined on pixel level by overlaying the cleaned class maps. The change in objects is determined comparing the segmented images and analysing them for differences.

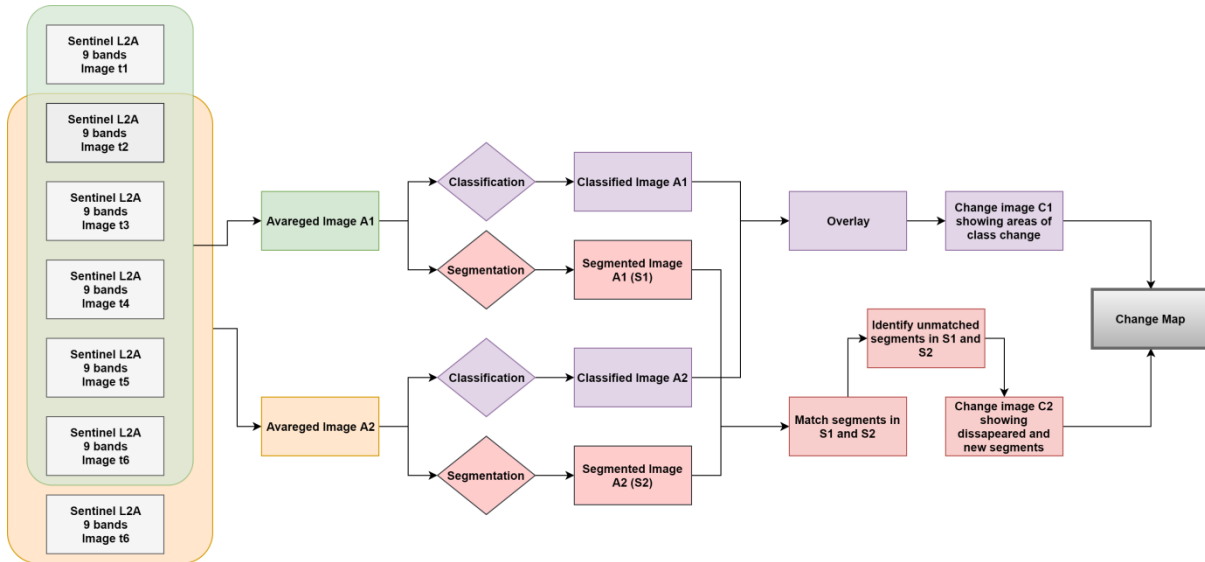


Figure 3.38: Change Detection Procedure

#### 3.6.1 Overlay

In Figure 3.39 the overlay procedure is visualized. The cleaned change maps C1 and C2 are laid over each other to select the raster point that have changed in class. Mathematically it means C1 is subtracted from C2 and all raster point not-equal to zero are given a value of 1. It is a fast and powerful procedure for highlighting class changes.

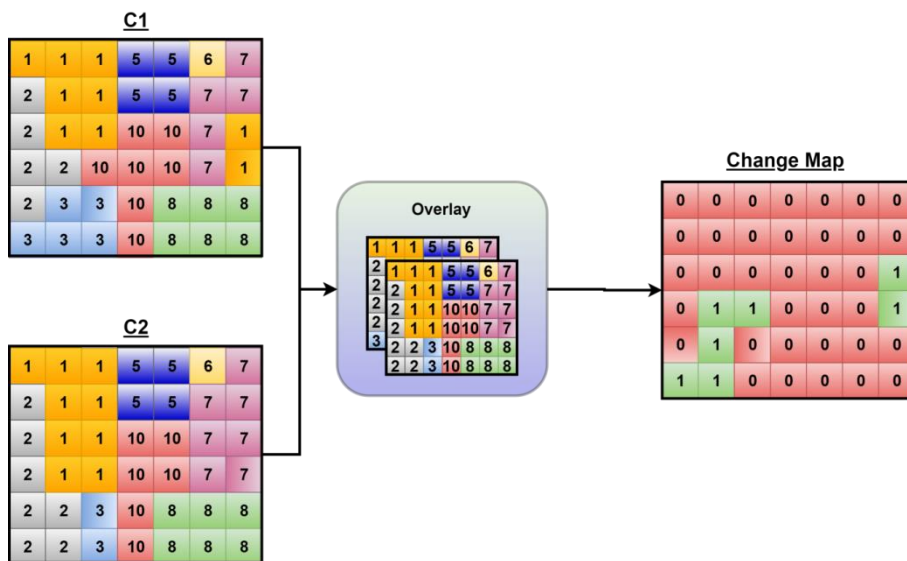


Figure 3.39: Overlay

### 3.6.2 Segment Matching

In the ACDet system a segment is considered an object. Thus the consecutive segmented images S1 and S2 are used to determine the removal of existing objects, and the appearance of new objects. Note that the segmentation procedure, as described in section 3.4, is done purely on spectral information, not involving additional features.

The procedure is visualized in Figure 3.40 and described in pseudo-code in Algorithm 2. The segments in S1 and S2 have unique identifiers, but there is no correlation between the labels of the two. Merely the location of the grid points (pixels) can be used for relating points in S2 and S2. Taking S2 as baseline per segment the corresponding raster points in S1 are selected.

#### Algorithm 2: Segment Matching

1.  $n1$  = number of segments (labels) in S1
2.  $n2$  = number of segments (labels) in S2
3. Initialize  $A = \text{zeros}(n1, n2)$
4. Loop over  $x$  in range  $(0, n1)$ :
  1. Find  $S2[S1==x] \gg$  store as  $b$
  2. Determine unique values in  $b$  and their count
  3. Store count of unique values in  $A[x, \text{unique}] = \text{count}$
5. Loop over  $y$  in range  $(0, n2)$ :
  1.  $\text{Label2} = y$
  2. Select highest value per column of  $A$  and store as  $\text{label1} = \max(A[:, y])$
  3. Clear remainder of corresponding row to avoid double matching:  
 $A[\text{label1}, :] = 0$
  4. Store results in vector  $M = [\text{object\_nr}, \text{label1}, \text{label2}]$
6. Find all unmatched segments S1:  $\text{list}(\text{set}(S1) - \text{set}(M[:, \text{label1}])))$
7. Append unmatched segments S1 to  $M$  with  $\text{label2} = 0$  for those entries

In a table (A) with the rows being the segments of S1 and the columns the segments of S2, the class values of the selected pixels are stored with their count. Repeating this procedure for all values of S2. Table A provides an overview of the class labels of S1 present in each segment of S2. The highest count of class label is considered a match, and thus the corresponding labels are stored in a vector. To avoid double matching the row corresponding to the matched segment S1 is cleared. The matched labels are marked in green in table A in Figure 3.40.

It can occur that a segment in S1 is split up into two segments in S2. For example, a farm field is segmented as a single object in S1 but seen as two objects in S2 because half of the field crops have been planted. This occurred with segment 2 in S1 being split up into segments 12 and 13 in S2. Segments 2 and 12 have been matched and the row is cleared to avoid double matching. As such segment 13 is seen as a new segment, resulting in a value of zero for the corresponding label S1 in table B.

Another possibility is disappearance of segments. The area in S1 covered by segments 3, 4, and 5 is divided into only two segments in S2, namely 14, and 15. As such there is no corresponding segment in S2 to match for segment 5 in S1, which is marked as yellow in table A. Segment 5 is stored as unmatched in table B with a label 2 value of 0.

The results stated in table B are transformed back to a raster of original size. When an object is present in both image 1 and 2 it is interpreted as unchanged. These are the matched segments and the corresponding raster points are assigned a value of 0. Change is defined as either a removed or new object, assigning the values 1 and 3 respectively.

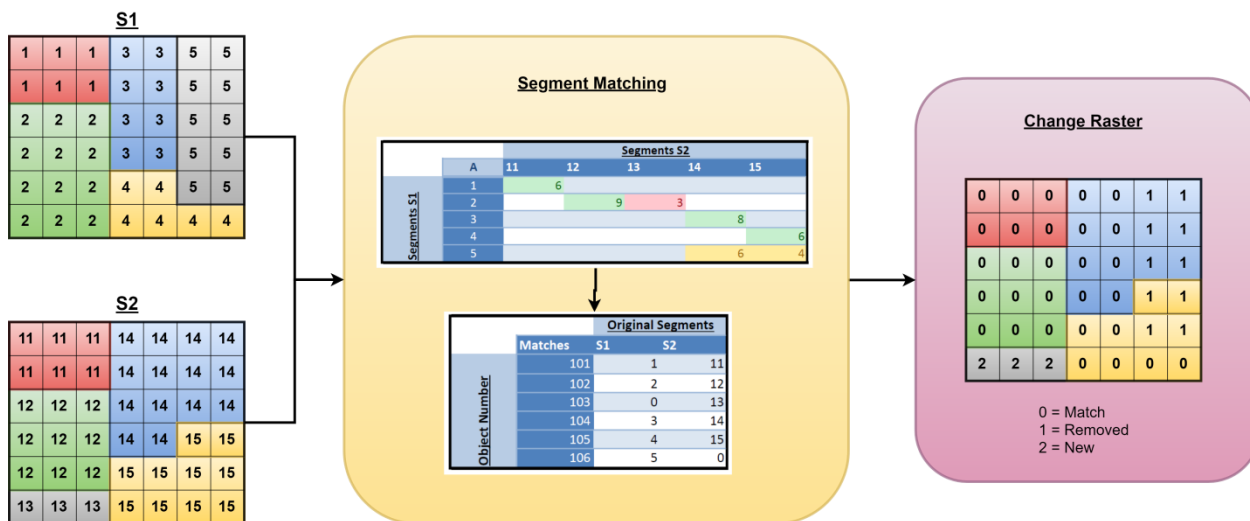


Figure 3.40: Segment Matching Procedure

### 3.6.3 Change Map

The final change map is compiled by combining the change map of class change (overlay) and the change map of object change (segment matching). Change is defined as a point where both the object and the class have changed. The purpose of this research is to provide a solution to operators analysing satellite imagery by providing a change map with markings only there where real change happens. So overall to limit the amount of information presented to the operator, but to increase the significance of the change in the points that are marked.

Using merely the class change map gives a scattered image and too much information. Although the accuracy of ACDet in terms of pixel classification is high there remains an error rate of about 9-11%. Using only the change map of objects is also not a viable option. Despite techniques to limit the variation in segmentation such as averaging the images, there remains a significant variation in segments between two consecutive images.

Limiting the definition of change to the points where both the class and object have changed severely reduces the amount of information, without deleting relevant change. After all the objective is to detect new infrastructure which would mean both a class change and an object change. If ACDet should be used for different purposes the definition of change and thus the relevant code could be altered to the specific needs of that new question.

Figure 3.41 displays a final change map. The outcomes of the overlay and segment matching are combined and change map is multiplied by the class map C2 to provide class information on the changed pixels visible. The white areas are all pixels without change. This is a zoom-in of an area with a relatively large amount of change to give a good representation of the capabilities.





Figure 3.41: Change Map Unfiltered

### 3.6.4 Filtering

The trade-off between providing as much information as possible and filtering out unnecessary information is an ever on-going process. As this research is aimed at detecting infrastructural change and new islands, two types of filtering can be applied without losing valuable insights.

Specific class transitions between C1 and C2 are used to filter, see Table 3.3. Transition from and to none classification (class 0) are filtered out as this is the system response for not being able to identify the object.

Transitions between 'grass' and 'farmland' are left out. Both are highly fluctuating classes due to weather, maintenance and farming. It is a field on its own, and thus left out. Lastly the transitions between the classes 'building' and 'road' are left out. These classes have some overlap and show a high fluctuation in pixel being classified as either or one or the other. This is due to overlap of the classes as well as object size, resulting in mixed pixels that are hard to classify.

Table 3.3: Class Transition Filtering

Class C1	Class C2
0: None	All
All	0: None
4: Grass	8: Farmland
8: Farmland	4: Grass
6: Building	7: Road
7: Road	6: Building

A second form of filtering applied is minimum object size. This research is aimed at automatically displaying structural changes of significant size. Small groups of pixels or even single pixels showing up on the change map are, although they can be accurate, not of value at this moment. A minimum group size of 9 pixels is used as filtering. Where the pixels are required to be connected by a face, as displayed in Figure 3.42, and not merely by an edge.

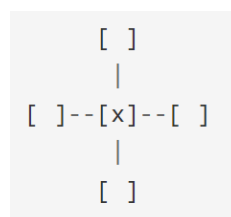


Figure 3.42: Face Connectivity (SciKit-Learn, 2018)

## 4 Verification

Verification of the performance of ACDet is done using labelled verification pixels. Using the procedure described in section ‘Training and Verification Set’, labelling data is used to create a set of labelled pixels. The set is randomly split into a training and verification set with a ratio of 2:1.

After the MLP classifier has completed training on the training set, the verification set is classified and the results are compared with the known labels. The performance of the system is measured in four parameters:

- Precision ( $P$ ): is the positive predictive value of the system. It is the fraction of true positives ( $T_p$ ) among the total of positives:  $T_p + F_p$ , with  $F_p$  being the number of false positives.

$$P = \frac{T_p}{T_p + F_p} \quad 4.1$$

- Recall ( $R$ ): is the sensitivity of the system. It is the fraction true positives out of the total number present in the data set: the true positives + false negatives ( $F_n$ ).

$$R = \frac{T_p}{T_p + F_n} \quad 4.2$$

- F1-Score ( $F1$ ): is the weighted average of the precision and recall. The score ranges from 0 to 1 with 1 being the highest score.

$$F1 = 2 \frac{P \times R}{P + R} \quad 4.3$$

- Percentage of unclassified pixels: is the percentage of pixels below the classification threshold and thus marked as unknown (class 0: None) by the system.

The precision, recall and F1 score are determined per class and per mean image, displayed in Figure 4.1, Figure 4.2, and Figure 4.3 respectively. The overall performance of the system can be concluded as being high with an average precision of 96%, recall of 94% and F1-score of 95%. Analysing in detail the underlying values the class ‘sea’ stands out with both a precision and recall of almost 100% in every image. This can be attributed to the location of the class and the highly distinctive features. A significant portion of misclassification concerns mixed pixels. With the low resolution of 10x10 m a pixel is likely to contain the edge of an object. For sea pixels this problem occurs only at the border with the beach and occasionally for large objects on sea. The averaging process applied in ACDet removes mixed pixels that are surrounded by pure sea pixels, further reducing the extent of the problem. This explains the high identification accuracy. The second reason lies in the distinctive features of water pixels. Water is well recognisable using the spectral bands as well as using the NDWI and MNDWI indices.

‘Farmland’ has the second highest precision with an average of 99%, but the recall with an average of 89% is markedly lower. The latter can be attributed to the overlap of the classes ‘grass’ and ‘farmland’. Both classes contain a stage of bare land. Furthermore, ‘farmland’ also contains stages in which grass is growing at the land. When the crops are on the land, the longest period in the “lifecycle” of ‘farmland’, ‘farmland’ is very distinguishable from grass explaining the high precision. When farmland lies as bare land or has light grass growing on it shares more characteristics with grass land and it may be classified as such. The training and verification labels do not take into

account these time variations as they are updated only once a year and will accordingly still mark the field as ‘farmland’, explaining the recall of 89%. The fact that ‘farmland’ is mistaken for grass by the MLP classifier is proven by the confusion matrix displayed in Table 4.1. The matrix reports the class assigned by the verification labels versus the predicted class by the MLP classifier. A 100% score would thus result in a diagonal matrix. The column farmland has ‘grass’ as the highest number of the alternative classes, meaning that most misclassified ‘farmland’ pixels are categorized as ‘grass’. An option to reduce this misclassification would be to also train ACDet with seasonally timed input data. The class change that occurs knows a seasonal aspect. It occurs due to harvesting which is usually performed in the same period each year. A training set that contains data from each season for multiple years could learn to recognize pixels that turn from ‘farmland’ to ‘grass’ land the same season every year. This approach could not be tested at present due to the lack of enough seasonally defined data.

Table 4.1: Confusion Matrix M6-Uncleaned

		True Class by Verification Labels									
		Water Inland	Sand	Heath	Grass	Forest	Building	Road	Farmland	Sea	Dune
Predicted Class by MLP Classifier	Water Inland	1371953	2512	954	22811	17409	19321	14296	4755	1073	2664
	Sand	458	208690	327	217	356	737	509	405	33	3606
	Heath	253	56	548445	508	19194	1347	1478	997	0	22227
	Grass	34716	177	5763	1485241	93449	6585	35143	22816	7	37916
	Forest	11132	37	34058	48107	1340928	5662	6094	4392	0	35944
	Building	2085	2917	1056	975	3082	421945	71164	9119	4	822
	Road	1767	643	101	1123	1506	15492	134329	3766	74	1001
	Farmland	21391	59463	20967	241231	36536	174096	149998	10372166	2	144659
	Sea	36278	3025	0	0	0	138	170	1	16603457	0
	Dune	2343	15083	13183	2703	26235	784	922	7331	0	726217

The class ‘building’ comes with a moderate average precision of 68% and recall of 80%. This can be attributed to the high percentage of mixed pixels and the wide features space of the class. As explained in section 3.3 on feature engineering buildings have many mixed pixels due the relatively small object size compared to the spatial resolution. And the inaccuracy of the training labels, including areas such as gardens. This effect of ponds and gardens is seen back in the column building of Table 4.1, showing a large number of building pixels being miscategorised as ‘water inland’ or ‘farmland’. Secondly the class shows a large spectral variety in objects. Commercial objects often have flat white or black roofs composed of tar and gravel, while residential homes have pointed roofs constructed of clay or list roof tiles. Added to that are elements such as rooftop gardens and solar panels. The spectral properties of these structures are significantly different, causing the class ‘building’ to lack very distinguishable features.

‘Road’ has proven a difficult class to detect. With an average precision of 25% it is with an arm’s length the worst performing class in terms of precision. The average recall of 80% is better, but still the lowest of all classes. These results are most likely caused by the object size. Roads are generally too narrow to detect with the given spatial resolution. Using higher resolution images would improve the precision here. Next to that a road is not always neatly situated at the middle of a pixel, but partly in several adjacent pixels. On top of that the labelling polygons do not exclude the central reservation of roads with lanes separated by grass and plants. These factors combined make it a difficult class to distinguish. Looking at the column Road in Table 4.1 it is noticeable that it is mostly mistaken for farmland, likely due to the central and side reservations of roads. To limit the effects only major highways have been selected for training and verification.

The majority of the classes ('water inland', 'sand', 'heath', 'grass', 'forest', and 'dune') show good scores of precisions from 72% ('grass') to 92% ('water inland'), recalls from 88% ('forest') to 96% ('sand') and f1-scores from 81% ('dune') to 91% ('water inland'). The scores are not significantly influenced by the thresholding applied, as can be seen in Figure 4.4 with a percentage of unclassified pixels due to thresholding ranging from 3.11% for M7 to 7.03 for M11. These numbers verify the performance of the ACDet concerning pixel based classification.

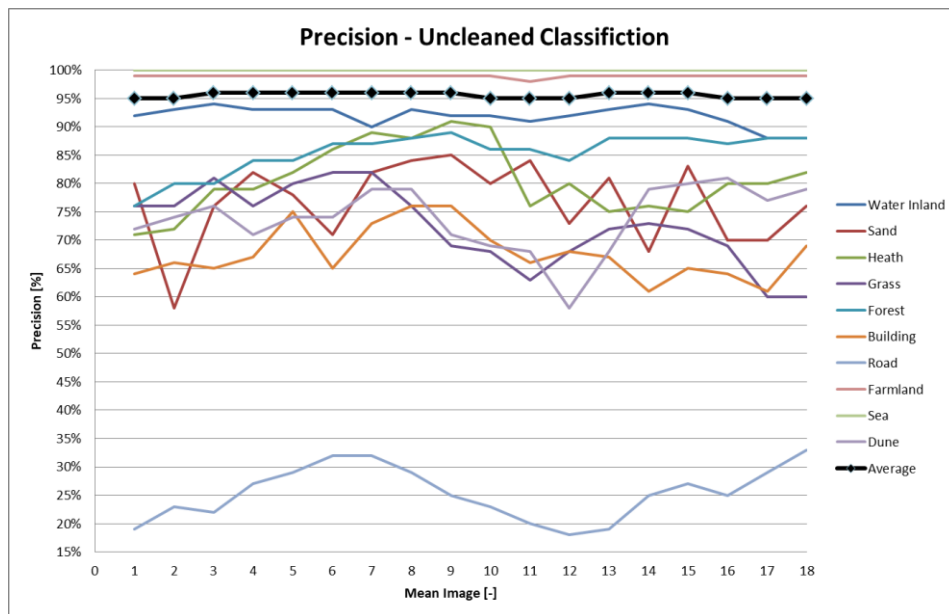


Figure 4.1: Precision of uncleaned classification of mean images

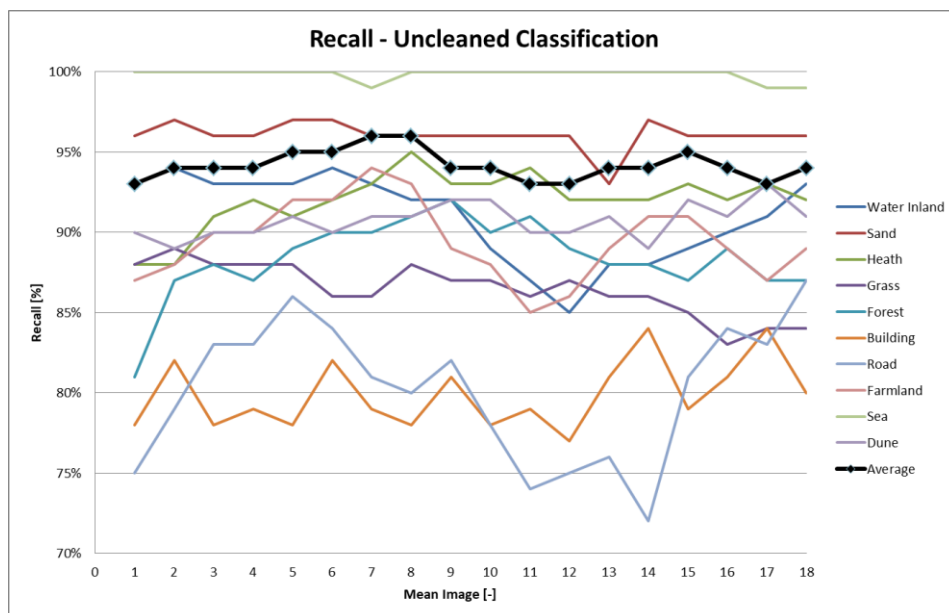


Figure 4.2: Recall of uncleaned classification of mean images

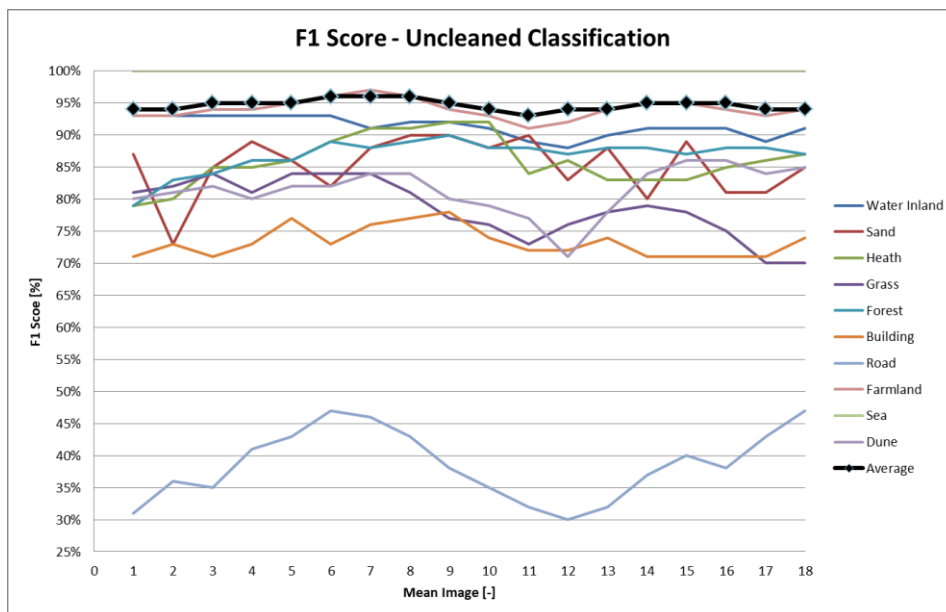


Figure 4.3: F1 Score of uncleaned classification of mean images

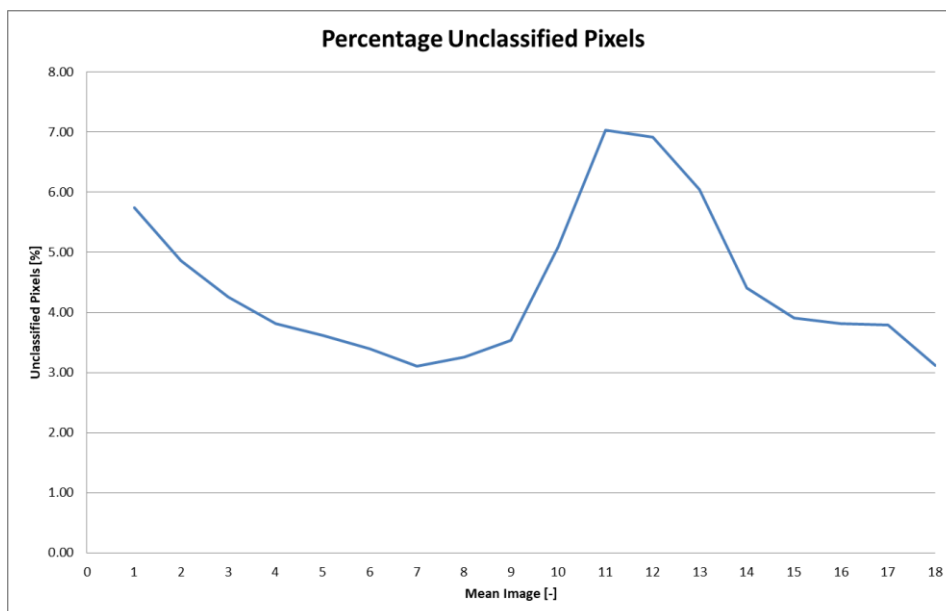


Figure 4.4: Percentage of unclassified pixels per mean image for uncleaned classification

The classification cleaning, as explained in section 3.5, has an effect on the pixel based class map. To test the effects of the method the same statistics for the cleaned class maps as for the uncleaned version are determined: precision, recall, f1-score, percentage of unclassified pixels. To highlight the effect of classification cleaning the difference between the scores for the cleaned and uncleaned class maps are portrayed in Figure 4.5, Figure 4.6, Figure 4.7, and Figure 4.8. The value changes are depicted in percent-point (%-point) per mean image.

Analysing the average for the precision, recall, and f1-score it can be concluded that the overall performance of the system is relatively unchanged. Per class however there are improvements and deteriorations. Improvements are made in the classes ‘sand’, ‘heath’, ‘farmland’, and ‘dune’ with an increase in f1-score of 0.009, 0.020, 0.005, and 0.040 percent-point respectively. The greatest

improvements are made in the difficult classes: 'heath', and 'dune'. These classes present a challenge for the MLP classifier as they are classes mainly recognisable as an area, and less per pixel. The classification cleaning procedure enhances the effect of areas by homogenizing the class map and thus improving the detection performance. 'Heath' and 'dune' show an average improvement in f1-score of 0.020 and 0.04 percent point relatively.

The cleaning process has a negative impact on classes 'water inland', 'grass', 'forest', 'building', and 'road' with a decrease in f1-score of 0.040, 0.041, 0.018, 0.019, and 0.001 percent-point respectively. These are piece by piece classes with a good performance in the uncleaned class map. Making the decrease in performance a minimal impact on the overall performance of the system.

It can also be questioned if the performance measures for the cleaned class map are accurate. The verification labels have a higher resolution than the images. As explained in section 3.5 it is not certain that the details that appear to be visible actually match the spatial resolution of the images. Next to that it is a design choice to value a homogenized class map over a pixel-based one. The emphasis of ACDet lies in detecting objects of scale and not single pixels.

Figure 4.8 plots the difference in 'percentage of unclassified pixels' between an uncleaned and a cleaned class map in percent-point. It shows a reversed scale. A decrease in percentage is an improvement in performance as the number of pixels of which the system is uncertain goes down. On every image an improvement is shown with an average decrease of 1.02 percent point of unclassified pixels. This is a significant improvement showing a decreased sensitivity of the system, making it more reliable as an automated system.

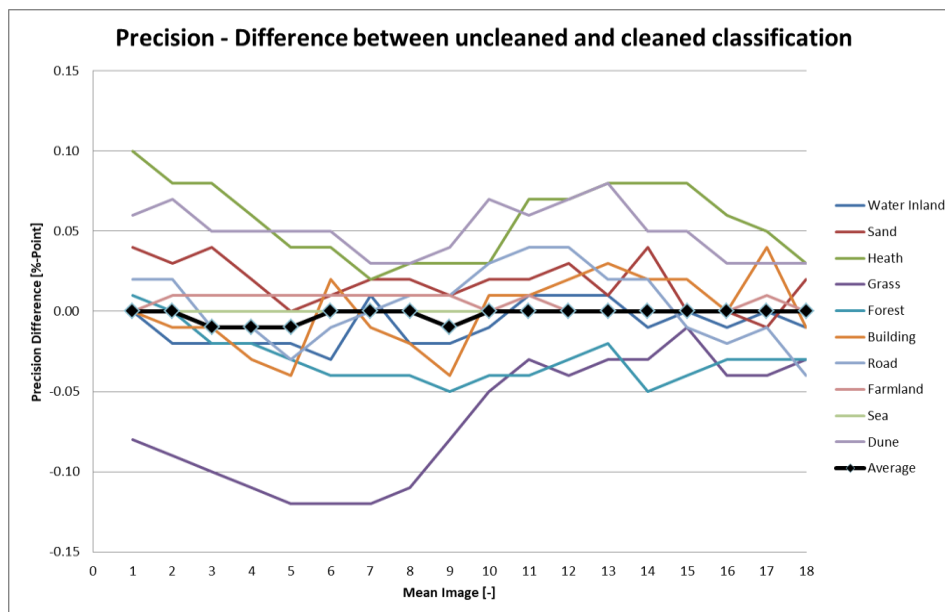


Figure 4.5: Precision difference between uncleaned and cleaned classification

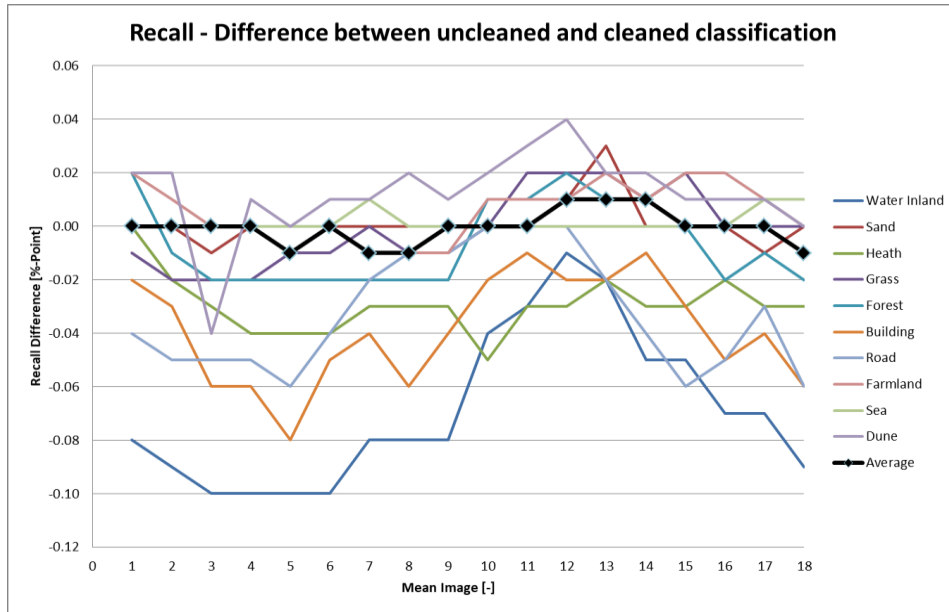


Figure 4.6: Recall difference between uncleaned and cleaned classification

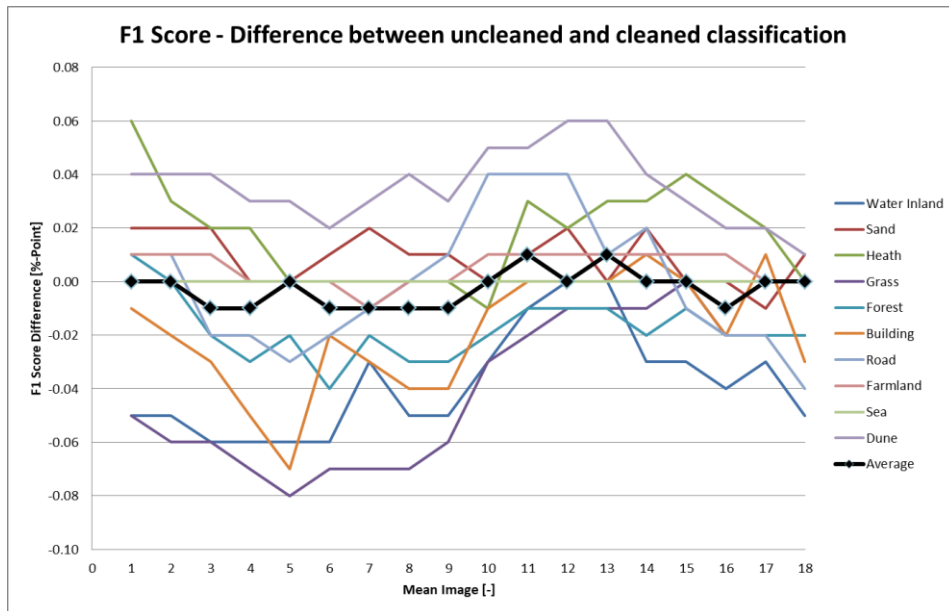


Figure 4.7: F1 Score difference between uncleaned and cleaned classification



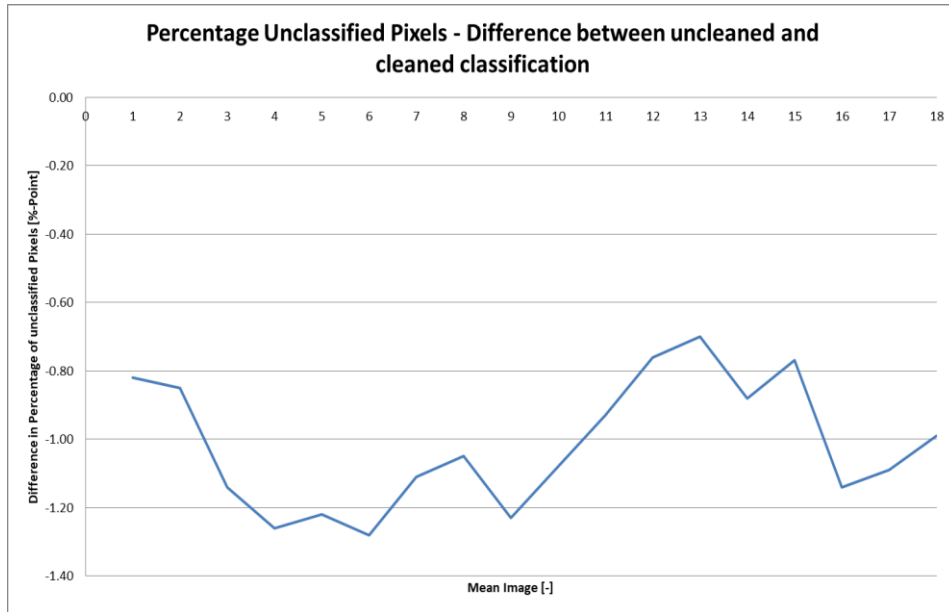


Figure 4.8: Percentage of unclassified pixels difference between uncleaned and cleaned classification

## 5 Experimental Set-Up

This section describes the experimental set-up of the research. As test case for the performance and usability of ACDet the construction of the Marker Wadden island is selected. The input data used in the test case are described in section 5.2 followed by a listing of the specification of the hardware used in the set-up.

### 5.1 Marker Wadden Test-Case

It is a unique development project situated in the Markermeer, to stimulate fauna both under water and above, and underwater vegetation. The project's location is mapped in Figure 5.1.

The project runs from 2016 to 2020, with main construction occurring in the years 2016, 2017 and 2018 (Natuurmonumenten, 2017). The timeline perfectly matches the satellite imagery available for this research. It is a unique test case to have an island newly constructed, with matching timeline and project planning available to verify the system's outcome. For these reasons it was selected as both test and validation case for ACDet.



Figure 5.1: Marker Wadden Project Location (Natuurmonumenten, 2017)

## 5.2 Data Source

As input a time series of images from the Sentinel-2A and Sentinel-2B satellites is used. The time series ranges from November 13, 2015 to May 26, 2018 with a total of 53 images. Note that these are L2A products, meaning pictures with over 50% clouding have already been left out. The images are ranked by visual inspection on cloud coverage using the levels stated in Table 5.1. Images of a level 0, 1, or 2 are used as input for ACDet. The more clouded images are left out to avoid a negative performance impact on the system. The images used for input are listed in Table 0.1 given in Appendix 0. The table also lists per mean image the original input images used for construction (the running average).

Table 5.1: Cloud Levels

Cloud Level	Description
0	Unclouded
1	Marginal regional clouding
2	Marginal overall clouding
3	Considerable regional clouding
4	Considerable overall clouding
5	Heavy overall clouding

## 5.3 Set-Up

For developing the ACDet program and running the test case a regular laptop has been used in combination with an external hard drive for storage. The exact specifications are given in Table 0.1 and Table 0.2. It is important to stress the capability of the program to run on regular hardware and still achieve good performance. Similar applications developed by research institutions of scale or commercial companies usually use computer clusters with performances incomparable to a single laptop.

The amount of RAM and the processor speed put a limitation on the program when it came to the Marker Wadden test case. The change detection procedure of comparing segments is computationally a very heavy task. Therefore a cut-out was made from the original picture. The original picture consists of 10980 x 10980 pixels, covering almost the complete northern half of The Netherlands. The cut-out used during the test set-up is 3645 x 3091 pixels, covering an area of 36.45 km by 30.91 km.

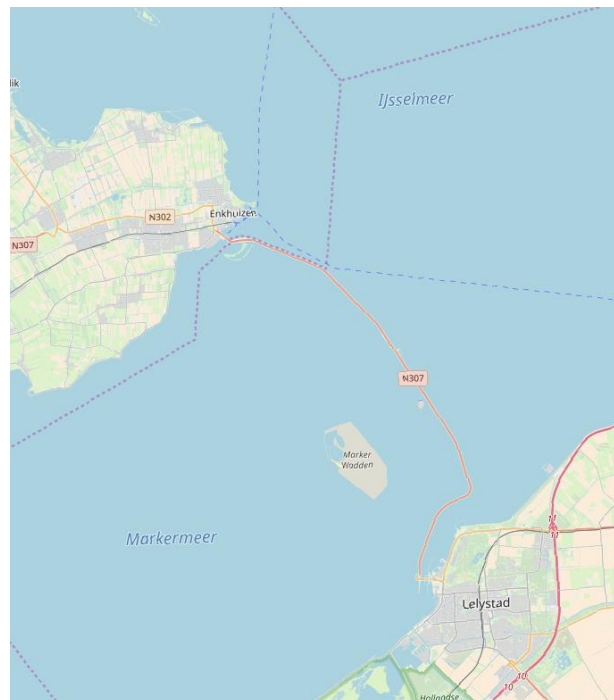


Figure 5.2: Map of cut-out area used for testing (OpenStreetMap, 2018)

## 6 Validation & Results




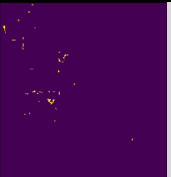
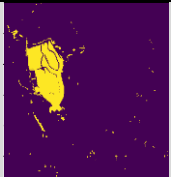



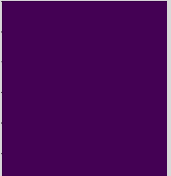
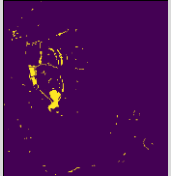


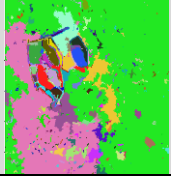

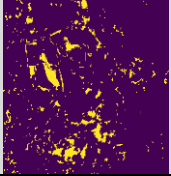

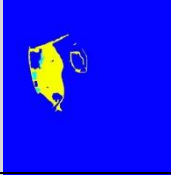
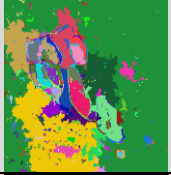
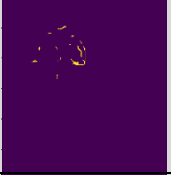
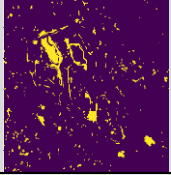
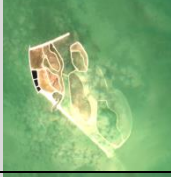
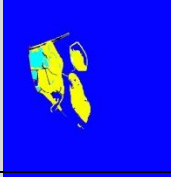
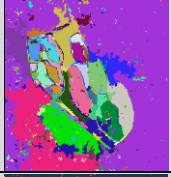
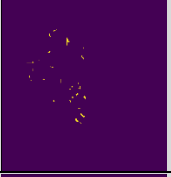
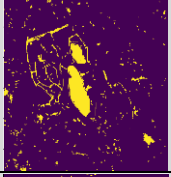

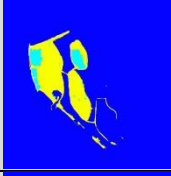
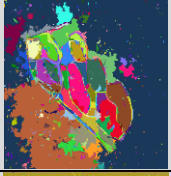
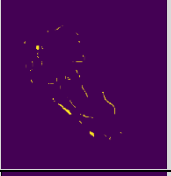
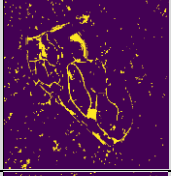

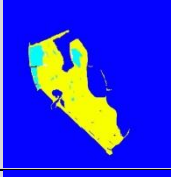
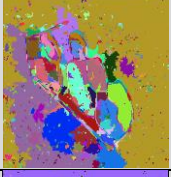
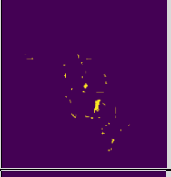
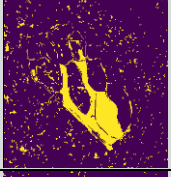

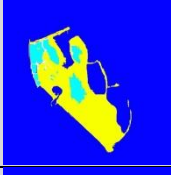
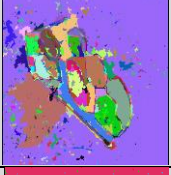
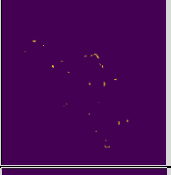
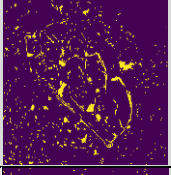

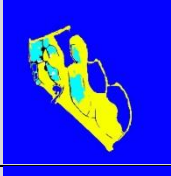
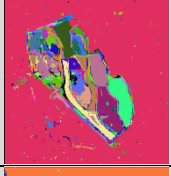
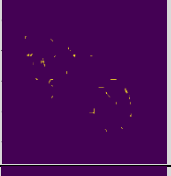
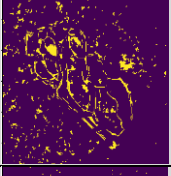


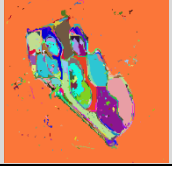
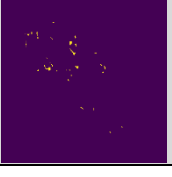
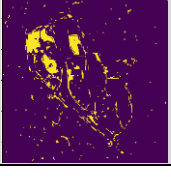
The results of ACDet on the Marker Wadden test case are listed in Table 6.1. The input image is stated with the time range it includes, followed by a cut-out of the image at the location of the Marker Wadden. Column 3 gives the class map, cleaned as described in section 3.5. The objects map shown in column 4 visualize the objects recognized by the system by mean of segmentation. Note that the colouring of the objects is random, due to random object numbering. The colouring in consecutive object maps is not consistent. The distinctive colours are merely used to visualize the different objects detected by the system.

Column 5 harbours the change map. The change map is always the change of the image compared to the previous image. For example CM4 given in row 5 of the table gives the change between M3 and M4. As described an area has to fulfil two requirements to be marked as change: the objects is changed, meaning it has been removed or added, and the class is changed. The map is filtered to remove small insignificant spots as described in section 3.6.4.

The change map shows a very limited number of areas and it may seem as if ACDet overlooks many changes that seem significant to the human eye. To outline that the system does identify all changes, but many are left out due to the criteria set for change a second change map is added in column 6. In this change map an area is marked as changed if either one of the requirements is met. So the class has changed, or the object has changed (removed or added).

Table 6.1: Time Series Results

Image / Period	Mean Image (M)	Class Map Cleaned (CLF)	Objects Map (OM)	Change Map (CM)	Change Map All Changes (CMAC)
1 13/11/2015 - 21/04/2016				N/A	N/A
2 23/12/2015 - 01/05/2016					
3 12/03/2016 - 11/05/2015					
4 01/04/2016 - 08/09/2016					

5 11/04/2016 - 27/03/2017					
6 21/04/2016 - 26/05/2017					
7 01/05/2016 - 29/08/2017					
8 11/05/2016 - 23/09/2017					
9 08/09/2016 - 07/11/2017					
10 27/03/2017 - 17/11/2017					
11 26/05/2017 - 21/12/2017					
12 29/08/2017 - 25/02/2018					
13 23/09/2017 - 02/03/2018					
14 07/11/2017 - 21/04/2018					

15 17/11/2017 - 06/05/2018						
16 17/12/2017 - 11/05/2018						
17 25/02/2018 - 21/05/2018						
18 03/02/2018 - 26/05/2018						

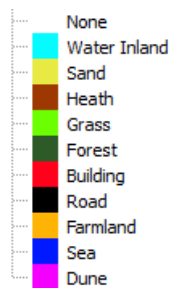


Figure 6.1: Legend for Change Maps

### 6.1 Marker Wadden Validation per Construction Phase

The results are validated by means of the official project planning of the Marker Wadden published by the contractor of the project Arcadis. Following the eight steps described in the project planning the relevant results are analysed for accuracy and validity.

Note that the imagery used is based on a running average. The time periods of the images do not exactly match the time periods listed in the project planning. Secondly it is not certain if the time periods listed in the project planning match reality of when construction of a phase was actually done. Therefore, both the imagery and time planning from the project planning are used to determine the relevant period of a phase.

Per phase a table is given with the project planning information provided by Arcadis and the relevant images are listed, which can be found in Table 6.1. As validation a short comparison is made per construction phase between the project planning and the retrieved results. Figure 6.2 displays the Marker Wadden divided into the different modules referred to in the projection planning.



Figure 6.2: Marker Wadden Modules (Arcadis, 2016)

6.1.1 Phase 0

Table 6.2: Phase 0 Project Planning (Arcadis, 2016)

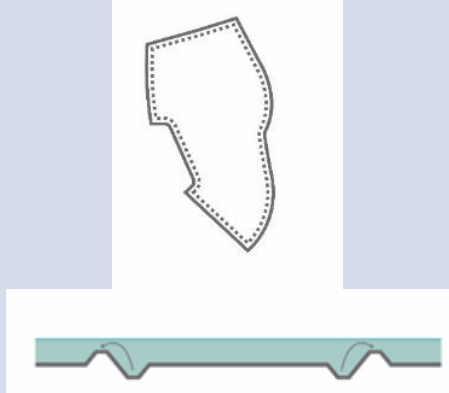
Phase	Description	Period
Preparation for work	<ol style="list-style-type: none"> <li>1. Polling work area</li> <li>2. Application of beaconing for shipping safety</li> </ol>	January – February 2016

Relevant images: 1, 2, and 3

No construction is performed yet. This is reflected in the class maps and change maps correctly.

6.1.2 Phase 1

Table 6.3: Phase 1 Project Planning (Arcadis, 2016)

Phase	Description	Period
	<ol style="list-style-type: none"> <li>1. Design underwater ring quay for modules A and B (side-casting)</li> </ol>	March – April 2016
	<ol style="list-style-type: none"> <li>2. Excavate sand extraction well and realize sludge channel</li> </ol>	April – October 2016

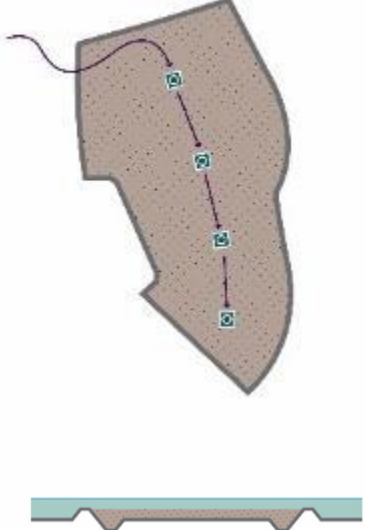
Relevant images: 4

In M4 the underwater ring quay is slightly visible. Extra modules have already started construction as well. Although the ring quay is not fully visible in the image yet, due to the averaging, it appears

correctly on both the class map and objects map. The changes compared to M3 are correctly detected and visualized on CMAC4 (the all change map). CM4 shows only a few changes. This is due to the fact that the small objects visible in OM3 are matched to the objects in OB4 and thus seen as existing segments and thus left out of the change map. This is explained in further detail in section 6.4.

### 6.1.3 Phase 2

Table 6.4: Phase 2 Project Planning (Arcadis, 2016)

Phase	Description	Period
	1. 1 <sup>st</sup> Filling underwater ring quay to level -2m NAP for modules A and B	April – May 2016
	2. Settling first filling	May – July 2016

#### Relevant images: 5 and 6

The underwater ring quay is filled with sand up to slightly below the water level. In M5 and M6 the modules A and B are now clearly visible and classes accordingly in CLF4 and CLF5. Because sand modules still have a layer of water on top of them the MLP classifier has a low certainty on both the classes 'sand' and 'sea'. The areas show features of both classes. In CLF6 the overlap in classes results in a section of module A being classed as unknown by ACDet, meaning the certainty of all classes is below the threshold of 50%.

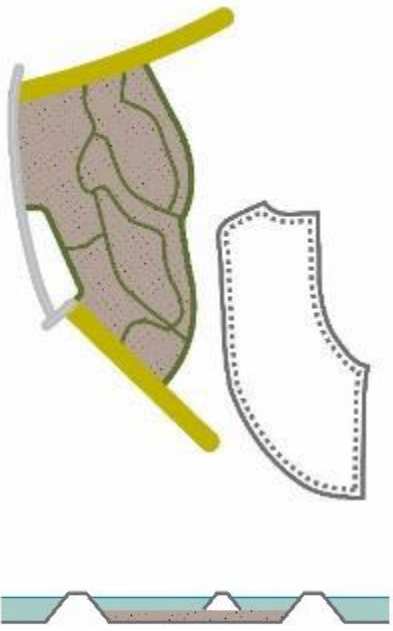
The modules have gone from 'Sea' in CLF4 to 'sand' in CLF5 to again 'sea' in CLF6. The objects (segments) have however not changed. They were already picked up by ACDet in OM4 and remained consistent up to OM6. Since the class has changed but not the object only one of the change criteria is met and thus the changes are not visible on CM5 and CM6, apart from some small segments.

It is important to note that the changes are in fact picked up by the system, as can be seen in CMAC5 and CMAC6, but as explained they are filtered out due to the strict criteria set for significant change. The missing of change due to an entire existing segment changing in class is further explained in section 6.4.



### 6.1.4 Phase 3

Table 6.5: Phase 3 Project Planning (Arcadis, 2016)

Phase	Description	Period
	1. Increase underwater ring quay to above water level as sand dam, stone dam and quay	July 2016
	2. Applying soft inner dikes module A and B	A: May – June 2016 B: August – September 2016
	3. Applying outer harbour (excluding recreational facilities)	July 2016
	4. Renew sand dam and apply ring dike of sand Module C	October 2016

#### Relevant images: 7, 8 and 9

The soft inner dikes are correctly identified as objects in the objects maps OM7, OM8, and OM9. On the CLF9 the dikes are identifiable as black lines. Due to the use of rocks for the strengthening of the dikes, ACDet classifies them as 'road'. The same event is visible on the two outer dikes. Arcadis applied large rocks (basalt blocks) on the outer edges of the sand dams as strengthening and as protection against waves. It is neat to see ACDet pick up these elements, since the line of basalt rocks is not wide enough to cover an entire pixel. Even with mixed pixels the features of basalt are distinguishable enough to classify the rocks.

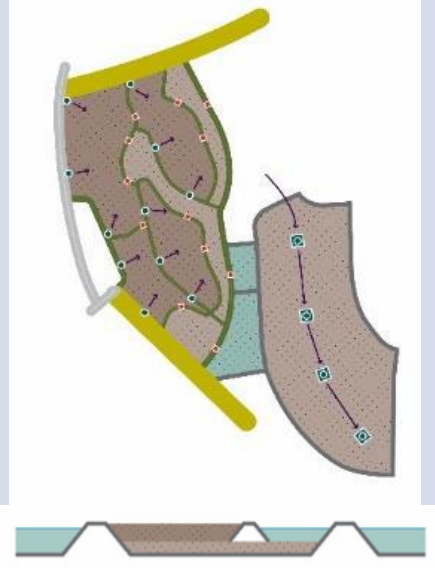
The new module C is shown in CLF9 and CMA9. In CM9 the module is left out since the object was already identified in OM8, but the class change was picked up at CLF9, thus in both cases only one of the change criteria was met.

Analysing the object maps OM7, OM8, and OM9 shows a vast number of segments in the water. This can be attributed to turbulences in the water around the island as can be seen on M7, M8, and M9. The turbulences are most likely due to shallow water, clouds, waves, or a combination of them. This is where the change maps have an advantage over the all change maps. Having the double criteria for change makes that these irregularities are filtered out neatly in CM7 to CM9, while they do appear on CMA7 to CMA9.

### 6.1.5 Phase 4

Table 6.6: Phase 4 Project Planning (Arcadis, 2016)

Phase	Description	Period
-------	-------------	--------

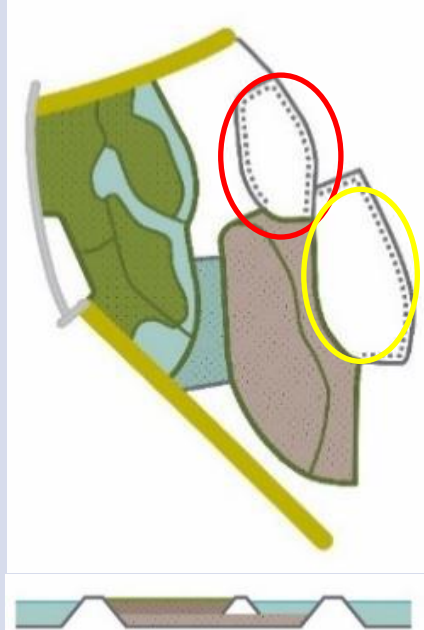
	1. Insert second and third fill modules A and B	July – October 2016
	2. Applying intermediate dammed sheltered bankside + filling	September – October 2016
	3. Application of the intermediate dammed sheltered bank	September – October 2016
	4. 1st filling module C	November – December 2016

**Relevant images: 9**

The application of the intermediate dammed sheltered bankside to connect modules A and C is difficult to detect on M9. Only the upper dam is visible and thus the only part visible on CLF9 and CMAC9. The dam was already detected as an object in OM8 and therefore not seen at CM9.

**6.1.6 Phase 5**

Table 6.7: Phase 5 Project Planning (Arcadis, 2016)

Phase	Description	Period
	1. Consolidation of the marsh of module A and B	March – December 2017
	2. Installing internal dikes module C	December 2016

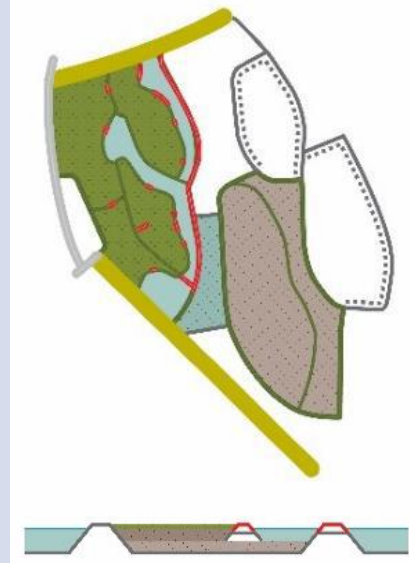
**Relevant images: 6, 7, 8, 9, 10, 11, 12**

The instalment of the red encircled dike of module C has started ahead of the official planning. Already from M6 and on the module has been visible. In OM6 the red encircled module was picked up as a separate object, while on the class map it has not been fully visible until CLF9. It is therefore difficult to relate back to a change map at which the module should have been visible, but looking at CMAC9 the module does appear.

The yellow encircled module was constructed according to schedule and starts to appear visually from M10 and on. On M10 the dike is still very vague and therefore not visible on the class map until CLF11. As an object it has been detected from OM9 and on. Even before the object appears visually. On CM11 the outlines of the dike are visible as well as on CMAC11.

### 6.1.7 Phase 6

Table 6.8: Phase 6 Project Planning (Arcadis, 2016)

Phase	Description	Period
	1. Sliding between and rear quays after development of dense reed vegetation module A and B	Ca. January – March 2018* * Exact date is selected based vegetation development.
	2. Installing a polder reef on the back dam	June – August 2018
	3. Applying recreational facilities (hiking trails, watchtower, harbor)	March – August 2018

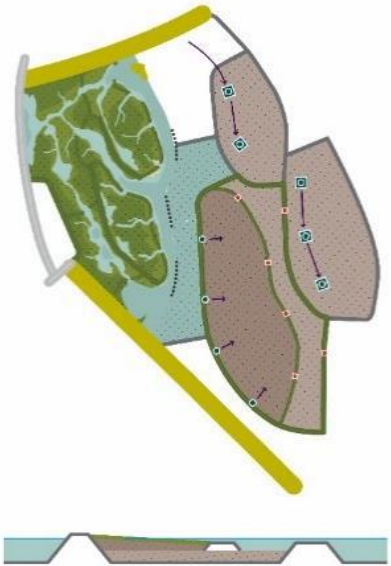
#### Relevant images: 13

The instalment of the polder reef on the back dam is clearly visible on CLF13. As with previous dams, basalt rocks that were used for reinforcement of the dam appear as 'road' on the class map. As an object the dam is not new, but merely reinforced, hence it is not visible as change on CM13. CMAC13 does correctly show the change when only a single change criterion is considered change.

The application of recreational facilities is such as hiking trails is visible in the same way. The hiking trails are consistently marked as 'road' due to the freshly applied gravel.

## 6.1.8 Phase 7

Table 6.9: Phase 7 Project Planning

Phase	Description	Period
	<ol style="list-style-type: none"> <li>1. Development of an initial bird paradise with land and water transitions and gradients by natural process of skew and flow</li> <li>2. 2nd filling module C</li> </ol>	December 2016 – March 2017

**Relevant images: 14, 15, 16, 17, and 18**

The island has neared completion and an initial bird paradise was developed with areas having transitions between water and land. Analysing CLF15 and onwards it can be concluded the new character of these areas is correctly detected by ACDet. The system now classified the areas as 'water inland'. A class that was trained on shallow water including vegetation, which is exactly what the areas where developed into. As the areas have changed in class completely without any object changes, the changes have gone by largely undetected on the change maps. However, the change maps all changes picked up the class changes correctly.

For an unknown reason the construction of module D is not included in the project planning. The module has visually started to appear as early as M8 and onwards. The detection of the module as a separate object has progressed correctly from OM8 up to OM18. Due to the unsynchronised detection of module D as new object and as new class it has gone largely undetected on the change maps. On the change maps all changes it appeared correctly and vividly, mainly on CMAC11 when the module was filled with the class 'sand' in full.

## 6.2 Time Series Change

As has become clear from discussing the results and validation, a large factor in change detection is the time period considered. To verify that all the gradual changes shown on the change maps add up to a more complete image of the island a cumulative change map is constructed based on the change maps given in Table 6.1 (CM2 to CM18).

The cumulative change map is shown in Figure 6.4. All the minor changes shown on the individual change maps which may have seemed insignificant add up to a clear outline of the island. The fillings of the modules have gone largely undetected. This can be attributed to the asynchronous detection of objects and classes. In general ACDet picked up modules as separate objects in stage earlier than the module being classified into a new class. Hence only a single change criterion was met. A second unique occurrence during the construction of the Marker Wadden was modules changing in class completely, e.g. from 'sea' to 'sand' when being filled, while the object remained exactly the same size and thus not considered as change.

Although the fillings are not shown in Figure 6.4, it gives a very clear view of the newly constructed island and is therefore of value when assessing an area on change with the use of images in a specific time series.

Another method of assessing the change during a specific time period is to compare the first and last image of the period. The downside is the lack of insight in the changes that might have happened during the period. But a possible advantage is the cleaner image, without insignificant temporary changes. Figure 6.3 shows the change map between M1 and M18. Note that this is the change map using the double criteria of change: object and class change. The colouring is used is consistent with the classes, legend is given in Figure 6.1. The island is now very distinguishable as a newly constructed form. The problem of asynchronous detection of objects and class does not play a role in this picture and thus the fillings of the modules are visible. The fact that some modules are still not filled is due to the fact that in the first picture, M1, these areas were 'sea', and in the last picture, M18, they were drowned and again classified as 'sea', hence unchanged.

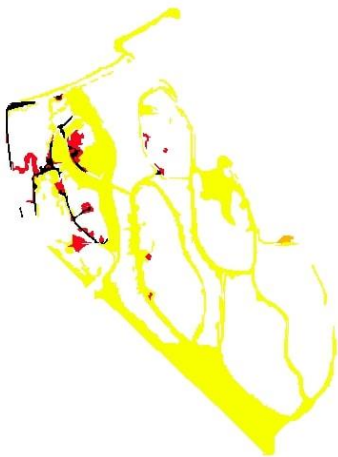


Figure 6.3: CM\_M1M18

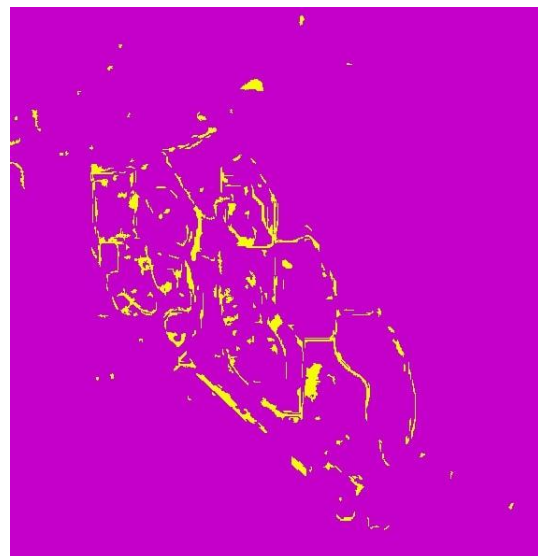


Figure 6.4: Cumulative Change Map of Time Series

### 6.3 Large Area Map

One of the goals of ACDet is to provide a scanning tool for large areas to flag considerable change, eliminating the need to zoom in and manually identify change between two consecutive images. This can be seen as the application of the research performed in this paper. Figure 6.5 to Figure 6.10 show the two consecutive images M10 and M11, their class maps, the change map and the change map including all changes.

The image considered is relatively small with an area of 36.45x30.91 km, due to a limit in processing power available. Already it is a challenging task to lift out the changes when only analysing the original images M10 and M11. The class maps give enhanced view on the situation, making it readily visible that a significant part is added to the Marker Wadden. On the land areas in the images it is much harder to identify change, even with the class maps at hand due to the sheer number of objects.

The change map visualized in Figure 6.9 shows a very limited number of changes. The task of identifying changes has been completely eliminated enabling an operator to focus on interpreting the changes marked by ACDet. Comparing the change map to the change map showing all changes (Figure 6.10) it seems as if major changes are missed. A quick look does not show you the considerable expansion of the Marker Wadden as it does in Figure 6.10. The value in the limited change map shown in Figure 6.9 lies in the information reduction. The number of changes flagged by ACDet in the all change map is too high to be considered a useful reduction in information load to the operator. A quick zoom-in and analysis of the individual points given in Figure 6.9 will increase the speed and quality of an operator assessing change in satellite images.

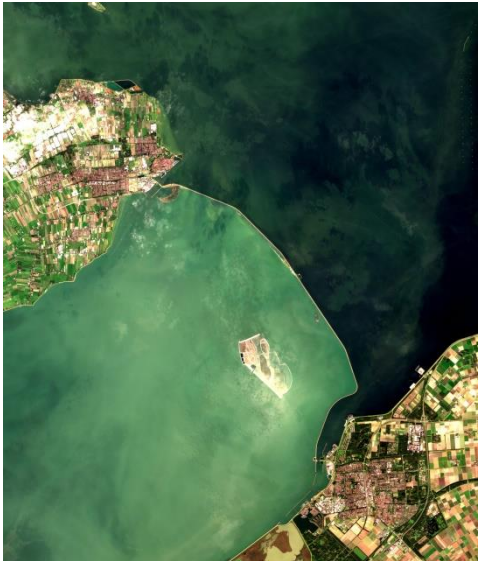


Figure 6.5: M10

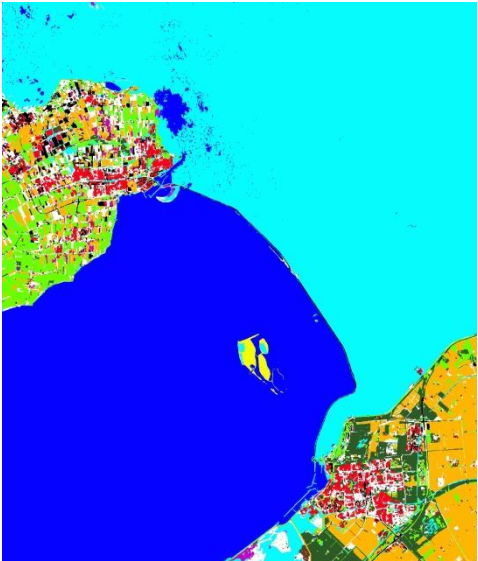


Figure 6.6: CLF10



Figure 6.7: M11

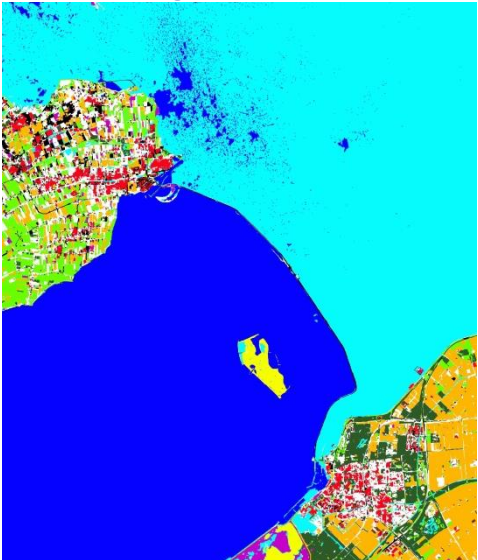


Figure 6.8: CLF11

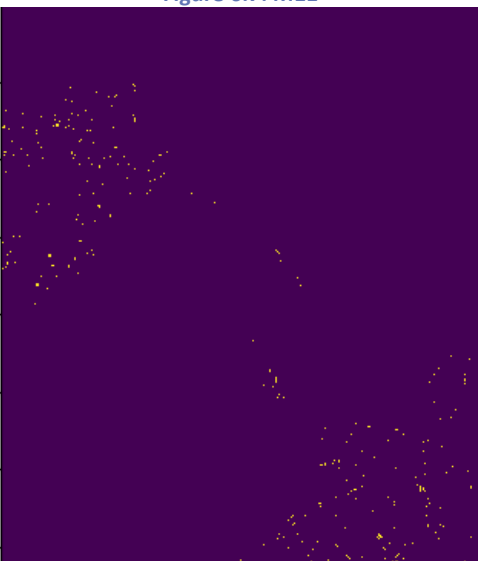


Figure 6.9: Change Map M10M11

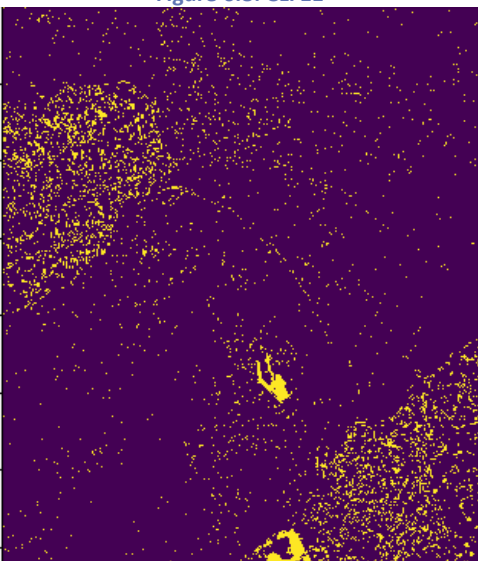


Figure 6.10: Change Map M10M11 All Change



## 6.4 Error Sources

In the automated change detection performed by ACDet there are two main sources of error. These errors lie in the criteria used to measure change and the method of segment matching. An area is marked as changed when both it has changed in class and the object has changed. An object is considered as changed when it is removed or added. Existing objects are marked as unchanged.



Figure 6.11: M4

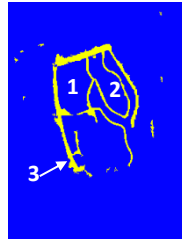


Figure 6.12: M4 Classification

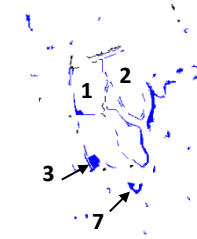


Figure 6.13: M4M5 Object Change Map. Blue depict new objects and black removed objects.



Figure 6.14: M5



Figure 6.15: M5 Classification

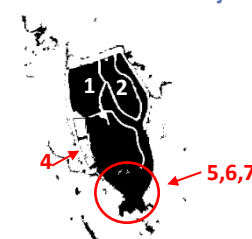


Figure 6.16: M4M5 class overlay map

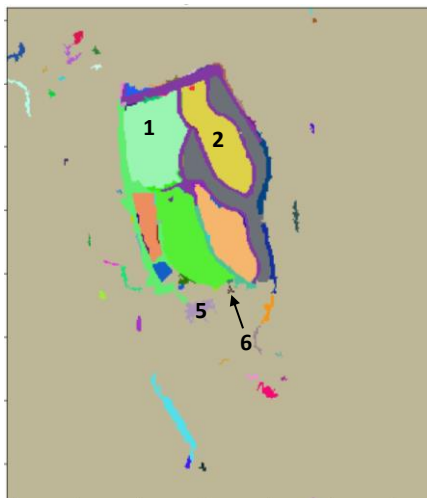


Figure 6.17: M4 segmentation

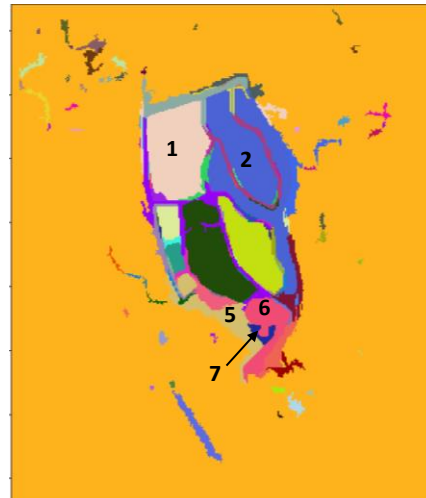


Figure 6.18: M5 segmentation

A source of error is in the manner of how segments are matched. As explained in section 3.6.2 segments in consecutive images are matched by finding the segment with the most overlap in terms of location. When a small segment in image 1 lies in its totality within a new larger segment in image 2, these segments will be matched. As existing segments are considered unchanged the new large segment will be left-out on the change map. An example of this situation is shown in Figure 6.16, Figure 6.17, and Figure 6.18 with areas marked 5, 6, and 7. The areas 5 and 6 appear as large new segments in the M5 segmentation. They are however not marked as new and thus invisible on the object change map. At the same time the much smaller, less significant segment 7 is visible on



the object change map. Looking at the M4 segmentation one can see two small segments, marked 5 and 6, present. The segment matching algorithm looks for the best overlap of these existing M4 segments in M5. As they lie within the newly formed, much larger areas 5 and 6 these are considered as matched and thus left out since they are existing. As segments 6 visible in M5 is already matched with the small segment 6 present from M4, there is no segment left to match area 7 to and thus it is considered as a new segment, hence it is visible on the object change map.

The same thing happens with the areas marked 1 and 2 in Figure 6.11 up to Figure 6.18. These are major areas that have gone from the class 'sea' to 'sand'. This change is neatly picked up in the class map and thus the class overlay map. As objects however the areas are considered as unchanged as the entire object has evolved, but the segment as such already existed. Since the criterium is to have both the object and the class changed these areas are left out on the change map.

A reverse situation is also possible. Considering in Figure 6.11 to Figure 6.16 the area marked as 3. In M5 this area has been dammed in and is thus considered a new object as seen in the object change map. Class wise it is however unchanged, remaining the class 'sea'. The area is therefore not considered as changed and thus not shown on the change map.

All changes missed by the system are well identifiable as well as the reason behind it. It is valuable to note that in none of the cases the change is completely ignored by ACDet, only a combination of choices made and criteria set make that certain changes are left out. Tweaking of these choices and criteria can increase the number of reported changes but at the cost of reduced accuracy and increased operator load.

## 7 Conclusions & Discussion

A reduction in system and launch cost and an increase in hardware capabilities has caused for an immense increase in the number of satellites, both government and commercial, observing the earth. With that the quantity of earth observation data has grown exponentially combined with spatial and temporal resolutions never seen before.

The sheer amount of data required to look outside the field of earth observation for methods to harness the full potential of it. Human analysis of satellite imagery is simply not possible anymore at this point. The rapid developments in computing power, digital storage, and machine learning algorithms have given an opportunity to set the next step in earth observation. With an interdisciplinary approach this research focused on the question how machine learning with the input of time series data can be used to come to an accurate, flexible and most of all automated change detection system for remote sensing.

An ever limiting factor when it comes to earth observation is the quality of the input data. This goes for both the satellite images as well as the labelling data. To limit the influence of the input data quality on the overall system performance a combination of techniques is incorporated in ACDet. On top of the multispectral 9 band imagery, 12 extra features are computed. The additional features use a combination of bands to incorporate contextual information and limit influence of a single pixel on the classification outcome. The effect of local irregularities is further reduced with the method called classification cleaning. Objects are homogenised in class and the classification resolution is brought in line with the resolution of the imagery.

The labelling used to generate training data has been extracted from multiple government verified databases. Next to that multiple classes from these databases have been aggregated to come to the 10 elementary classes that were used to train the MLP classifier. The use of elementary classes such as 'Sand' and 'Water Inland' has proven to be of great importance. It lets the algorithm focus on spectral distinctive classes and produce a high confidence output.

Increased value lies in the detection of objects. The challenges are first identifying objects similar to how a human would outline them in the image and second, having consistent object detection in a time series of images. With the use of graph-based segmentation ACDet is able to achieve these two requirements not only in a way that is much faster than alternative methods, but also consistent for consecutive images. The main advantage of the technique is the ability to use multi-dimensional data where similar segmentation methods can only deal with greyscale images. The consistency in a time series, which is of the utmost importance to limit irregularities in change detection, is increased by the use of a moving average on the satellite imagery.

This method of pre-processing images also shows its value when it comes to limiting the effects of temporary changes. In this research the aim was to present a satellite image operator with the minimal amount of information, showing only changes of importance. Based on this and on the specification of data available, the decision was made to focus on lasting changes of considerable scale, preferably infrastructure. Temporary changes due to either the temporary presence of objects or local weather conditions would cause for unwanted increase of low-quality information to the operator. The use of a moving average has proven in this research to be an effective way to reduce

the effects of temporary present objects, clouds, and lightning conditions. The process of classification cleaning further limited the effects of temporary changes and smoothed the output.

Existing change detection methods are focused on comparing two consecutive images. Gradual changes are therefore overlooked. In this research multiple elements have contributed to a time series approach in change detection. The use of a running average indirectly introduced a time series element into the system. Small and gradual changes of significance appear on the averaged image, while irrelevant effects are filtered out. Next to that the output of a cumulative change map gives the operator the possibility to observe an area within a specific time period. The map shows all changes in this time period. Large changes occurring in a gradual manner become clear as the cumulative change adds up.

When looking for change most systems focus on specific pre-defined objects and monitor them over time. By predefining the objects type one limits the changes detected by the system and might miss changes that would have been of great strategic value. The ACDet system has the unique ability to highlight significant change without predefining the object type. This is achieved by classifying pixel into elementary classes and grouping the pixels into objects using a graph based segmentation.

The output of ACDet was validated using the construction of the Marker Wadden island as test case. The construction of the island perfectly matched the time series used in this research and presence of an official project planning by the constructor provided an external source for validation, not often found in similar research.

Change is considered valid when both the class and object have changed. Where a change in object is considered the appearance or disappearance of an object. Validation showed change was picked up correctly by ACDet, but the strict criteria made that much of the change was filtered out. Looking at the all change map, where only one of the criteria met was considered as change, proved that none of the change was overseen by the system but some was purposely left out on the change map due to the criteria set. A fundamental difference compared to a situation when the system would have missed certain changes.

Multiple combinations of circumstances caused change to be filtered out. In the case of an existing object changing class in completeness at once it was not shown on the change map. This unique situation happened multiple times in the construction of the Marker Wadden with complete modules switching between sand and water. Another circumstance was the segmentation algorithm picking up objects before the classification of them changed. For example when a module was in construction, but still a few meters underwater it was seen as water correctly, but the object itself was already detected. The asynchronous change of the object and the class made that it was not seen as change, but it did appear on the change map all change.

A third situation of change considered insignificant was when a small segment grew considerable in size. As it was an existing object it was not considered as changed due to the criteria set. Further development of ACDet could have the system also take into account existing objects that have changed in size considerably and account them as change.

On the change map all change every single change was picked up correctly by ACDet. Zoomed in on the Marker Wadden test case it seemed the change map all change gave a better output than the

selective change map that required both criteria of change to be met. However the true value of the system lies in flagging change on a large area map. Zooming out showed that using a change map all change gives too much information and does not provide a real improvement compared to the current situation of manual change detection. The selective change map showed a very limited number of changed areas, considering the size of the area observed and the time period of the images (several months). An operator can use such a map and apply a viewing tool to quickly assess the changes flagged by the system. For the selected cases the change map all changes of that smaller area will then provide more information on the change at hand.

## 8 Recommendations

As every research this study has points of recommendation for improvement and future work. The recommendations are split up into low level (detailed) and high level (global) improvements. The detailed recommendations describe improvements that can be made to the system's methodology and parameter settings which will have a direct effect on the performance of ACDet. The global recommendations describe improvements that can be made to the input data and set-up of the system to increase performance. All recommendations are ordered on importance, starting with the highest impact improvement that can be done.

### 8.1 Detailed Improvements

As discussed in section 6.4 there are three error situations. The main error source is the decision to consider existing objects as unchanged, even when the size is changed considerably. As explained it can happen that a small spot of sand in image 1 is detected and in image 2 there is a large new sand module on the same spot. As the first sand segment completely lies within the new module it is matched and considered as existing. It is recommended to compute the change in area size for existing segments. Objects that change in size above a set threshold should also be considered as changed.

A second source of change not meeting both criteria is asynchronous detection. Meaning the change in object and change in class do not happen in the same image. In the observed cases the objects were detected earlier than the class changes, due to sand modules having a layer of water on top of them. A recommended action to counteract this is to tweak the parameters used in the graph based segmentation to a state where the detection of objects goes in sync with class change.

### 8.2 Global Improvements

A recommendation is the use satellite image with a higher spatial and temporal resolution. A significant error in the performance is due the existence of mixed (polluted) pixels. A higher spatial resolution, better fitting most object sizes, will result in a lower percentage of mixed pixels and thus an increased classification performance. An increase in temporal resolution provides the system with much more input data to detect change. Most likely resulting in a more accurate change map. The running average currently combined sometimes half a year of images into one average. This was due to the low temporal resolution of the images, and the large number of images that had to be left out because of clouding. Nevertheless it puts a constraint on the quantity and detail of change one can observe with such input imagery. Commercial grade satellite images would greatly improve the outcome of the change maps. Of course this will increase the required computational power.

A fourth recommendation is the use of accurate, time-bound labelling data. In this research the labelling data was used was updated once a year. If labelling data is available with a higher update frequency temporary effects such as seasons are better trainable and the performance of the classification would go up.

## Bibliography

- Alpaydin, E. (2004). *Introduction to Machine Learning*. Cambridge, Massachusetts, USA: MIT Press. doi:10.1017/CBO9781107415324.004
- Arcadis. (2016). *WERKPLAN MARKER WADDEN*. Assen: Arcadis Nederland. Retrieved 06 20, 2018, from [https://ro.lelystad.nl/plannen/NL.IMRO.0995.OV00020-/NL.IMRO.0995.OV00020-OW01/b\\_NL.IMRO.0995.OV00020-OW01\\_tb6.pdf](https://ro.lelystad.nl/plannen/NL.IMRO.0995.OV00020-/NL.IMRO.0995.OV00020-OW01/b_NL.IMRO.0995.OV00020-OW01_tb6.pdf)
- Azure Machine Learning Team. (2015). *Microsoft Azure Machine Learning: Algorithm Cheat Sheet*. Microsoft. Retrieved 08 13, 2018, from <http://aka.ms/MLCheatSheet>
- Baumann, P. R. (2009). HISTORY OF REMOTE SENSING SATELLITE IMAGERY PART II. Oneonta, New York, USA: Department of Geography, State University of New York. Retrieved 09 1, 2018, from <https://www.oneonta.edu/faculty/baumanpr/geosat2/RS%20History%20II/RS-History-Part-2.html>
- Beckers, B., Haneveer, M., Hekman, E., Ingen van, J., Jimenez Lluva, D., Kok, D., . . . Watts, T. (2015). HIREs: Holland's Intelligence, Reconnaissance, and Earth Surveillance. *10th IAA Symposium on Small Satellites for Earth Observation* (pp. B10-0606). Berlin: DLR.
- Coppin, P., Jonckheere, I., Nackaerts, K., Muys, B., & Lambin, E. (2004). Review Article Digital change Detection methods in ecosystem monitoring: a review. *International Journal of Remote Sensing*, 25(9), 1565-1596. doi:<http://dx.doi.org/10.1080/0143116031000101675>
- Cundill, S., der van Werff, H., & der van Meijde, M. (2015). Adjusting spectral indices for spectral response function differences of very high spatial resolution sensors simulated from field spectra. *Sensors*, 15, 6221-6240. doi:10.3390/s150306221
- DigitalGlobe. (2017, 05). *Satellite Information*. Retrieved 08 14, 2018, from DigitalGlobe-Resources: <https://www.digitalglobe.com/resources/satellite-information>
- ESA. (2009, 07 14). *Histry of Europe in Space*. Retrieved 09 22, 2018, from ESA: [https://www.esa.int/About\\_Us/Welcome\\_to\\_ESA/ESA\\_history/50\\_years\\_of\\_Earth\\_Observation](https://www.esa.int/About_Us/Welcome_to_ESA/ESA_history/50_years_of_Earth_Observation)
- ESA. (2018). *Data Products*. Retrieved 08 11, 2018, from Sentinel-2: [http://www.esa.int/Our\\_Activities/Observing\\_the\\_Earth/Copernicus/Sentinel-2/Data\\_products](http://www.esa.int/Our_Activities/Observing_the_Earth/Copernicus/Sentinel-2/Data_products)
- European Patent Office. (2016, 08 1). *Earth Observation*. Retrieved 9 22, 2018, from European Patent Office: <https://www.epo.org/news-issues/technology/space/earth-observation.html>
- Felzenszwalb, P. F., & Huttenlocher, D. P. (2004). Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2), 167-181. doi:10.1023/B:VISI.0000022288.19776.77
- Furieri, A. (2017, 03 03). SpatiaLite. Pontedera, Pisa, Italy: Laboratorio per la Qualita del Territorio. Retrieved from <https://www.gaia-gis.it/fossil/libspatialite/index>

- Gardner, M., & Dorling, S. (1998). ARTIFICIAL NEURAL NETWORKS (THE MULTILAYER PERCEPTRON) A REVIEW OF APPLICATIONS IN THE ATMOSPHERIC SCIENCES. *Atmospheric Environment*, 32, 2627-2636.
- GDAL /OGR Contributors. (n.d.). *gdal\_rasterize*. Retrieved 08 08, 2018, from Geospatial Data Abstraction software Library: [https://www.gdal.org/gdal\\_rasterize.html](https://www.gdal.org/gdal_rasterize.html)
- GDAL/OGR contributors. (2018). *Geospatial Data Abstraction software Library*. Retrieved 08 03, 2018, from Open Source Geospatial Foundation: <http://gdal.org>
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE International Conference on Computer Vision*. doi:10.1109/ICCV.2015.123
- Heaton, J. (2012). *Introduction to the Math of Neural Networks* (1st ed.). Chesterfield, MO, USA: Heaton Research. Inc.
- Het Kadaster. (n.d.). Basisregistratie Topografie. Apeldoorn, The Netherlands. Retrieved from <https://www.kadaster.nl/brt>
- Huete, A. (1988). A soil-adjusted vegetation index (SAVI). *Remote Sensing of Environment*, 25, 295-309. doi:10.1016/0034-4257(88)90106-X
- Kang, N. (2018, 06 27). *Multi-Layer Neural Networks with Sigmoid Function— Deep Learning for Rookies (2)*. Retrieved 08 14, 2018, from Towards Data Science: <https://towardsdatascience.com/multi-layer-neural-networks-with-sigmoid-function-deep-learning-for-rookies-2-bf464f09eb7f>
- Karpathy, A. (2018). *Course Notes Spring 2018 CS231n Convolutional Neural Networks for Visual Recognition*. Retrieved from Stanford: <http://cs231n.stanford.edu/>
- Kingma, D., & Jimmy, L. (2017). Adam: A Method for Stochastic Optimization. *3rd International Conference for Learning Representations 2015*. San Diego. Retrieved from <https://arxiv.org/pdf/1412.6980.pdf>
- Kramer, H. J. (2018). *Copernicus: Sentinel-2*. (ESA) Retrieved 08 11, 2018, from eoPortal Directory: <https://directory.eoportal.org/web/eoportal/satellite-missions/c-missions/copernicus-sentinel-2>
- Krizhevsky, A., Sutskever, I., & Hinton, G. (n.d.). ImageNet Classification with Deep Convolutional Neural Networks. In F. Pereira, C. Burges, L. Bottou, & K. Weinberger (Ed.), *Advances in Neural Information Processing Systems 25*. Neural Information Processing Systems 2012. Retrieved from <http://www.cs.toronto.edu/~fritz/absps/imagenet.pdf>
- Lema, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets. *Journal of Machine Learning Research*, 18(17), 1-5.
- Lemaitre, G., Nogueira, F., Oliveira, D., & Aridas, C. (n.d.). *Comparison of the different under-sampling algorithms*. Retrieved 08 09, 2018, from Imbalanced-Learn: <http://contrib.scikit->

[learn.org/imbalanced-learn/stable/auto\\_examples/under-sampling/plot\\_comparison\\_under\\_sampling.html](http://learn.org/imbalanced-learn/stable/auto_examples/under-sampling/plot_comparison_under_sampling.html)

Lemaitre, G., Nogueira, F., Oliveira, D., & Aridas, C. (n.d.). *Random over-sampling*. Retrieved 08 09, 2018, from Imbalanced-Learn 0.3.0.: [http://contrib.scikit-learn.org/imbalanced-learn/stable/auto\\_examples/over-sampling/plot\\_random\\_over\\_sampling.html?highlight=random%20over%20sampling](http://contrib.scikit-learn.org/imbalanced-learn/stable/auto_examples/over-sampling/plot_random_over_sampling.html?highlight=random%20over%20sampling)

Lemaitre, G., Nogueira, F., Oliveira, D., & Aridas, C. (n.d.). *Random under-sampling*. Retrieved 08 09, 2018, from Imbalanced-Learn 0.3.0.: [http://contrib.scikit-learn.org/imbalanced-learn/stable/auto\\_examples/under-sampling/plot\\_random\\_under\\_sampler.html](http://contrib.scikit-learn.org/imbalanced-learn/stable/auto_examples/under-sampling/plot_random_under_sampler.html)

Macias, A. (2018, 07 05). China is quietly conducting electronic warfare tests in the South China Sea. *CNBC Politics*. Retrieved 08 19, 2018, from <https://www.cnn.com/2018/07/05/us-intel-report-china-quietly-testing-electronic-warfare-assets-on-sp.html>

Natuurmonumenten. (2017). *Marker Wadden Facts*. Rijkswaterstaat. Retrieved 08 16, 2018, from [https://oudewebsite.natuurmonumenten.nl/sites/default/files/Marker-Wadden-factsheet\\_0.pdf](https://oudewebsite.natuurmonumenten.nl/sites/default/files/Marker-Wadden-factsheet_0.pdf)

NLR. (2018, 08 11). *Earth Observation*. Retrieved from NLR: [www.nlr.nl](http://www.nlr.nl)

Noorbergen, H. (2018, 06 25). gewaspercelen registratie. Marknesse, Zuid-Holland, The Netherlands.

OpenStreetMap. (2018). Open Street Maps. Retrieved 09 20, 2018, from <https://www.openstreetmap.org>

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

Planet. (2018). *Product*. Retrieved 09 22, 2018, from Planet: <https://www.planet.com/products/>

Prati, R. C., Batista, G. E., & Monard, M. C. (2009). Data mining with imbalanced class distributions: concepts and methods. *4th Indian International Conference on Artificial Intelligence (IICAI-09)*, (p. 359{376). Retrieved from <http://sites.labc.icmc.usp.br/pub/mcmonard/PratiIICAI09.pdf>

QGIS Development Team. (2018). QGIS Geographic Information System. Open Source Geospatial Foundation Project. Retrieved from <http://qgis.osgeo.org>

Raghav, R., Lemaitre, G., & Unterthiner, T. (2017). *Compare the effect of different scalers on data with outliers*. Retrieved 08 13, 2018, from SciKit-Learn: [http://scikit-learn.org/stable/auto\\_examples/preprocessing/plot\\_all\\_scaling.html#sphx-glr-auto-examples-preprocessing-plot-all-scaling-py](http://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html#sphx-glr-auto-examples-preprocessing-plot-all-scaling-py)

RDW. (2012). *Overzicht maten en gewichten in Nederland*. Retrieved 08 12, 2018, from <https://www.rdw.nl/-/media/rdw/rdw/pdf/sitecollectiondocuments/ontheffingen-tet/themasite-ontheffingen/handleidingen/2-b-1097b-overzicht-maten-en-gewichten.pdf>



- SciKit-Image. (2014). *Image Segmentation*, Docs for 0.15.dev0. Retrieved 08 01, 2018, from SciKit-Image: [http://scikit-image.org/docs/dev/user\\_guide/tutorial\\_segmentation.html](http://scikit-image.org/docs/dev/user_guide/tutorial_segmentation.html)
- SciKit-Learn. (2018). *scikit-learn user guide* (Vol. Release 0.19.2). Retrieved from [http://scikit-learn.org/stable/\\_downloads/scikit-learn-docs.pdf](http://scikit-learn.org/stable/_downloads/scikit-learn-docs.pdf)
- Shuxiang, X., & Ling, C. (2008). A Novel Approach for Determining the Optimzal Number of Hidden Layer Neurons for FNN's and Its Application in Data Mining. *5th International Conference on Information Technology and Applications* (pp. 683-686). Sydney: University of Tasmania. Retrieved from <https://eprints.utas.edu.au/6995/>
- SINERGISE. (2018). *Sentinel 2 EO products*. (Laboratory for geographical information systems, Ltd.) Retrieved 04 25, 2018, from Sentinel Hub: [https://www.sentinel-hub.com/develop/documentation/eo\\_products/Sentinel2EOproducts](https://www.sentinel-hub.com/develop/documentation/eo_products/Sentinel2EOproducts)
- Singh, A. (1989). Review Article Digital change detection techniques usign remotely-sensed data. *International Journal of Remote Sensing*, 10(6), 989-1003. doi:<https://doi.org/10.1080/01431168908903939>
- Tewkesbury, A., Comber, A., Tate, N., Lamb, A., & Fisher, P. (2015). A critical synthesis of remotely sensed optical image change detection techniques. *Remote Sensing of Environment*, 1-14. doi:<https://doi.org/10.1016/j.rse.2015.01.006>
- Tucker, P. (2018, 07 09). Russia Building Up Military Sites on Poland's Border Before Trump-Putin Meeting. *Defense One*. Retrieved 08 19, 2018, from Defense One: <https://www.defenseone.com/technology/2018/07/satellite-photos-show-new-activity-russias-military-exclave-kaliningrad/149531/?oref=d-river>
- Verrelst, J., Koetz, B., Kneubühler, M., & Schaepman, M. (2006). DIRECTIONAL SENSITIVITY ANALYSIS OF VEGETATION INDICES FROM MULTI-ANGULAR CHRIS/PROBA DATA. *Proceedings of the ISPRS Commission VII Symposium 'Remote Sensing: From Pixels to Processes'*. XXXVI. Enschede, The Netherlands: International Society for Photogrammetry and Remote Sensing. Retrieved from <http://www.isprs.org/proceedings/XXXVI/part7/PDF/227.pdf>
- Xavier, A., & Vettorazzi, C. (2004, 05). Mapping leaf area index through spectral vegetation indices in a subtropical watershed. *International Journal of Remote Sensing*, 25(9), 1661-1672. doi:<http://dx.doi.org/10.1080/01431160310001620803>
- Xu, H. (2006, 06 20). Modification of normalised difference. *International Journal of Remote*, 27(14), 3025-3033. doi:<http://dx.doi.org/10.1080/01431160600589179>
- Yuheng, S., & Hao, Y. (2017). Image Segmentation Algorithms Overview. *arXiv:1707.02051v1*. Retrieved from <https://arxiv.org/abs/1707.02051>
- Zhang, J. (2010). Multi-source remote sensing data fusion: status and trends. *International Journal of Image and Data Fusion*, 1(1), 5-24. doi:<https://doi.org/10.1080/19479830903561035>

Zhu, Z. (2017). Change detection using landsat time series: A review of frequencies, preprocessing, algorithms, and applications. *ISPRS Journal of Photogrammetry and Remote Sensing*, 370-384. doi:<https://doi.org/10.1016/j.isprsjprs.2017.06.013>

## List of Figures

Figure 3.1: Classification Procedure	14
Figure 3.2: Construction of Running Average	15
Figure 3.3: Image 1. Sentinel 2A 2015-11-13	16
Figure 3.4: Image 2. Sentinel 2A 2015-12-23	16
Figure 3.5: Image 3. Sentinel 2A 2016-03-12	16
Figure 3.6: Image 4. Sentinel 2A 2016-04-01	16
Figure 3.7: Image 5. Sentinel 2A 2016-04-11	16
Figure 3.8: Image 6. Sentinel 2A 2016-04-21	16
Figure 3.9: Averaged image	17
Figure 3.10: Mean image without cloud removal	18
Figure 3.11: Mean image with cloud removal using percentile	18
Figure 3.12: Labelling using QGIS	20
Figure 3.13: Polygon overlay	21
Figure 3.14: Random Over-Sampling	22
Figure 3.15: Random Under-Sampling	22
Figure 3.16: Layers and nodes of MLP network	24
Figure 3.17: Classification procedure of MLP network (Kang, 2018)	24
Figure 3.18: ReLu activation function (Karpathy, 2018)	25
Figure 3.19: Training convergence using stochastic gradient descent (Krizhevsky, Sutskever, & Hinton)	25
Figure 3.20: NN Layer design - effect of layer size (Karpathy, 2018)	25
Figure 3.21: 3x3 Contextual Kernel	27
Figure 3.22: 5x5 Contextual Kernel	27
Figure 3.23: Class distribution feature 4	29
Figure 3.24: NDWI (SINERGISE, 2018)	29
Figure 3.25: NDVI (SINERGISE, 2018)	30
Figure 3.26: MNDWI (SINERGISE, 2018)	30
Figure 3.27: SAVI (SINERGISE, 2018)	30
Figure 3.28: NDVI705 (SINERGISE, 2018)	31
Figure 3.29: LAI-SAVI (SINERGISE, 2018)	31
Figure 3.30: Robust Scaler (Raghav, Lemaitre, & Unterthiner, 2017)	32
Figure 3.31: Class distribution feature 4 Standardized	32
Figure 3.32: Perceptually important regions in an arbitrary grayscale image (Felzenszwalb & Huttenlocher, 2004)	33
Figure 3.33: Set of components consisting of a single vertex each	34
Figure 3.34: Segmentation steps	35
Figure 3.35: Original picture M1	37
Figure 3.36: Pixel based classification	37
Figure 3.37: Cleaned classification	37
Figure 3.38: Change Detection Procedure	38
Figure 3.39: Overlay	38
Figure 3.40: Segment Matching Procedure	40
Figure 3.41: Change Map Unfiltered	41
Figure 3.42: Face Connectivity (SciKit-Learn, 2018)	42

Figure 4.1: Precision of uncleaned classification of mean images	45
Figure 4.2: Recall of uncleaned classification of mean images	45
Figure 4.3: F1 Score of uncleaned classification of mean images	46
Figure 4.4: Percentage of unclassified pixels per mean image for uncleaned classification	46
Figure 4.5: Precision difference between uncleaned and cleaned classification	47
Figure 4.6: Recall difference between uncleaned and cleaned classification	48
Figure 4.7: F1 Score difference between uncleaned and cleaned classification	48
Figure 4.8: Percentage of unclassified pixels difference between uncleaned and cleaned classification	49
Figure 5.1: Marker Wadden Project Location (Natuurmonumenten, 2017)	50
Figure 5.2: Map of cut-out area used for testing (OpenStreetMap, 2018)	51
Figure 6.1: Legend for Change Maps	54
Figure 6.2: Marker Wadden Modules (Arcadis, 2016)	55
Figure 6.3: CM_M1M18	61
Figure 6.4: Cumulative Change Map of Time Series	61
Figure 6.5: M10	63
Figure 6.6: CLF10	63
Figure 6.7: M11	63
Figure 6.8: CLF11	63
Figure 6.9: Change Map M10M11	63
Figure 6.10: Change Map M10M11 All Change	63
Figure 6.11: M4	64
Figure 6.12: M4 Classification	64
Figure 6.13: M4M5 Object Change Map. Blue depict new objects and black removed objects.	64
Figure 6.14: M5	64
Figure 6.15: M5 Classification	64
Figure 6.16: M4M5 class overlay map	64
Figure 6.17: M4 segmentation	64
Figure 6.18: M5 segmentation	64
Figure 0.1: Class distribution feature 0	80
Figure 0.2: Class distribution feature 1	80
Figure 0.3: Class distribution feature 2	81
Figure 0.4: Class distribution feature 3	81
Figure 0.5: Class distribution feature 4	82
Figure 0.6: Class distribution feature 5	82
Figure 0.7: Class distribution feature 6	83
Figure 0.8: Class distribution feature 7	83
Figure 0.9: Class distribution feature 8	84
Figure 0.10: Class distribution feature 9	84
Figure 0.11: Class distribution feature 10	85
Figure 0.12: Class distribution feature 11	85
Figure 0.13: Class distribution feature 12	86
Figure 0.14: Class distribution feature 13	86
Figure 0.15: Class distribution feature 14	87
Figure 0.16: Class distribution feature 15	87

Figure 0.17: Class distribution feature 16	88
Figure 0.18: Class distribution feature 17	88
Figure 0.19: Class distribution feature 18	89
Figure 0.20: Class distribution feature 19	89
Figure 0.21: Class distribution feature 20	90

## List of Tables

Table 3.1: Labelling Classes	19
Table 3.2: Features	28
Table 3.3: Class Transition Filtering	42
Table 4.1: Confusion Matrix M6-Uncleaned	44
Table 5.1: Cloud Levels	51
Table 6.1: Time Series Results	52
Table 6.2: Phase 0 Project Planning (Arcadis, 2016)	55
Table 6.3: Phase 1 Project Planning (Arcadis, 2016)	55
Table 6.4: Phase 2 Project Planning (Arcadis, 2016)	56
Table 6.5: Phase 3 Project Planning (Arcadis, 2016)	57
Table 6.6: Phase 4 Project Planning (Arcadis, 2016)	57
Table 6.7: Phase 5 Project Planning (Arcadis, 2016)	58
Table 6.8: Phase 6 Project Planning (Arcadis, 2016)	59
Table 6.9: Phase 7 Project Planning	60
Table 0.1: Images used for running average	91
Table 0.1: Specifications Laptop	92
Table 0.2: Specification External Hard Drive	92

## List of Abbreviations

ACDet	Automated Change Detection
AI	Artificial Intelligence
API	Application Programming Interface
ASIS	Department of Aerospace Systems, ISR and Space Utilisation
BRP	Basisregistratie gewaspercelen (database)
CNN	Convolutional Neural Network
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
ESA	European Space Agency
GDAL	Geospatial Data Abstraction Library
GeoJSON	a database file format
ISR	Intelligence Surveillance and Reconnaissance
LAI	Leaf Area Index
MLP	Multi-Layer Perceptron Neural Network
MNDWI	Modified Normalized Difference Water Index
MST	Minimum Spanning Tree
NDVI	Normalized Difference Vegetation Index
NDVI705	Red-edge Normalized Difference Vegetation Index
NDWI	Normalizes Difference Water Index
NIR	Near Infrared
NLR	Netherlands Aerospace Centre
NN	Neural Network
PDOK	Publieke Dienstverlening op de Kaart
PYTHON	programming language used for ACDet
QGIS	open source geospatial software
ReLU	The rectified linear unit function
SAVI	Soil Adjusted Vegetation Index
SciKit	A Machine Learning Toolbox for PYTHON
RF	Random Forest
RGB	Red, Green, Blue colour coding
Shapefile	A database file format
SpatialLite	The database file format used in ACDet
Tanh	A hyperbolic tan function
Top10NL	GIS database of the Netherlands
TU	Technical University

## A - Feature Distributions

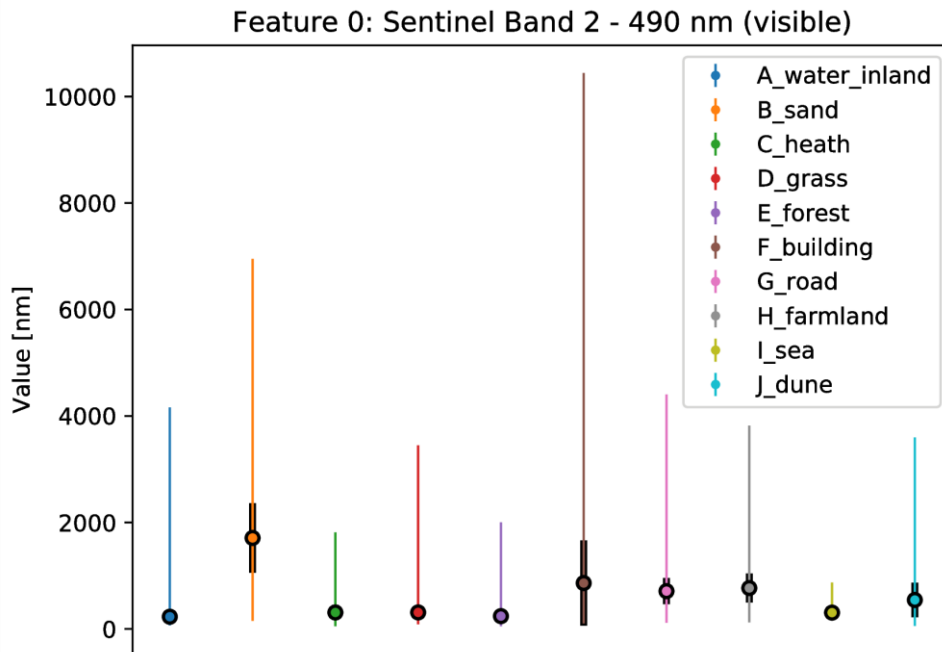


Figure 0.1: Class distribution feature 0

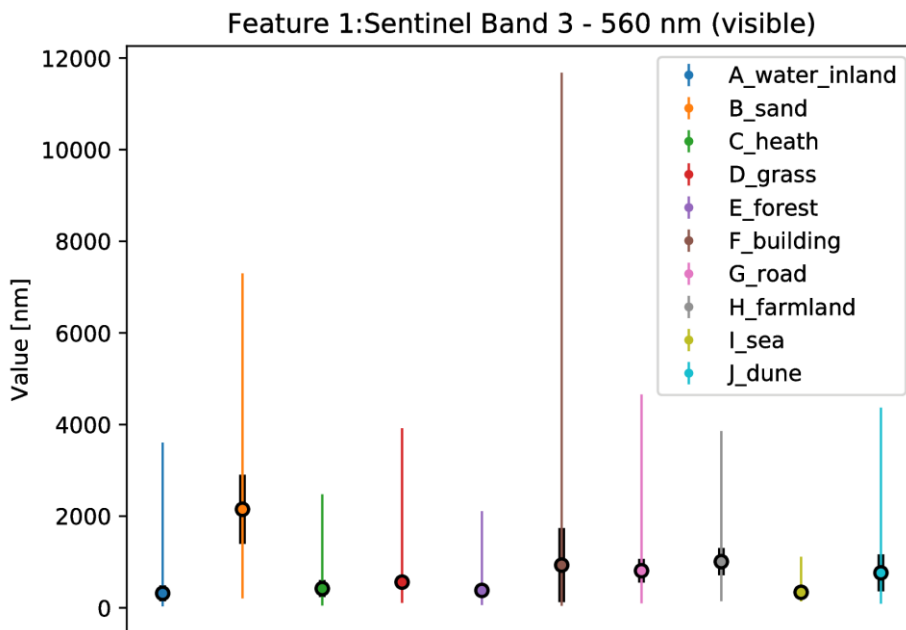


Figure 0.2: Class distribution feature 1



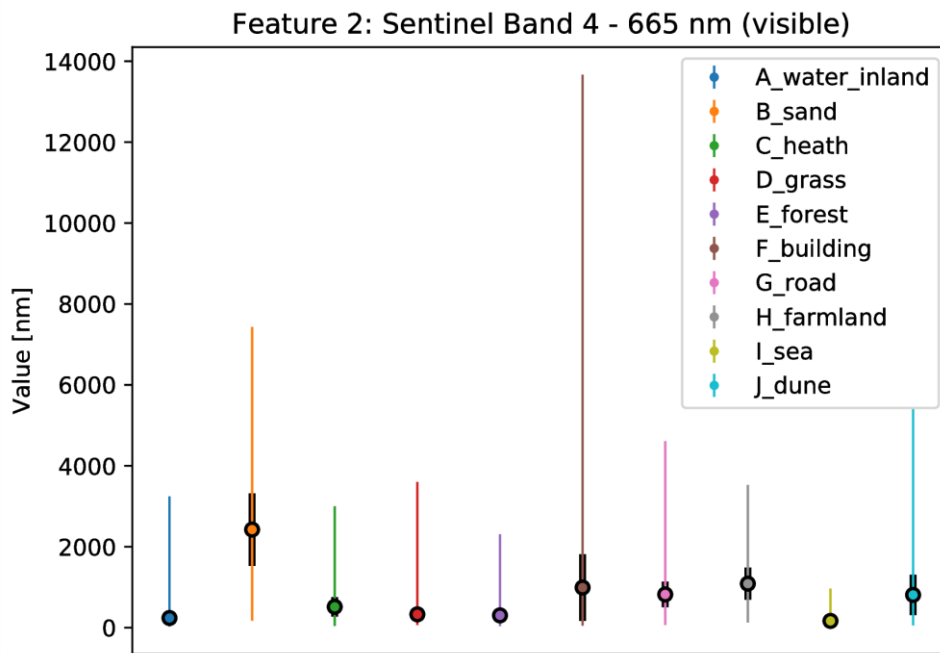


Figure 0.3: Class distribution feature 2

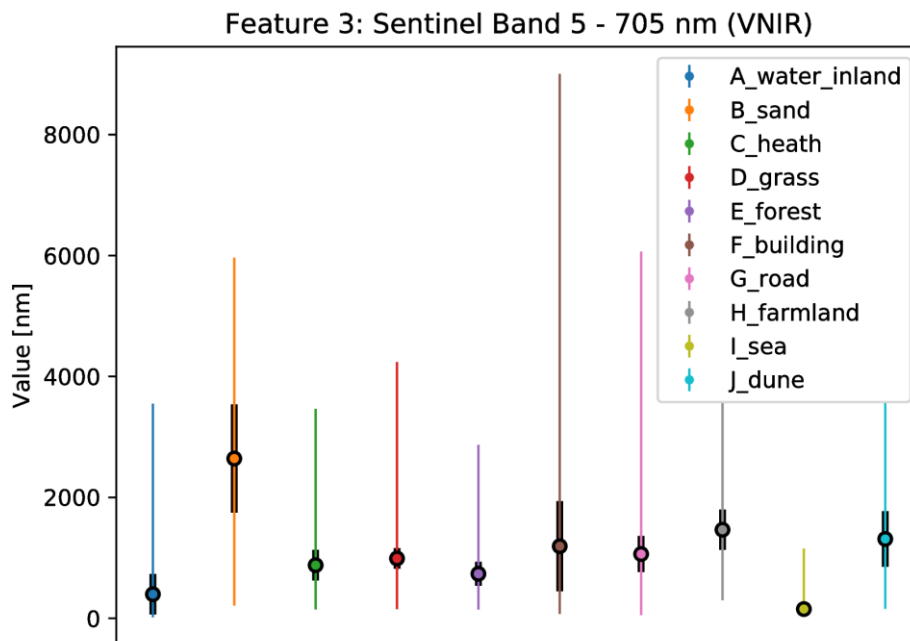


Figure 0.4: Class distribution feature 3

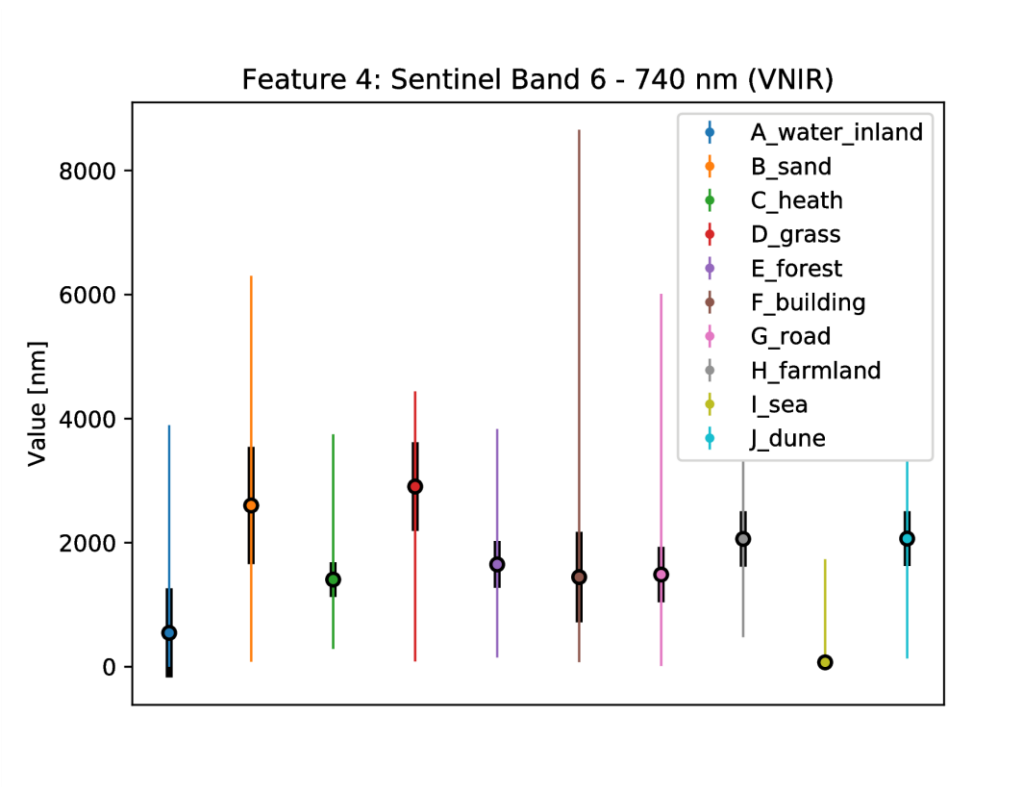


Figure 0.5: Class distribution feature 4

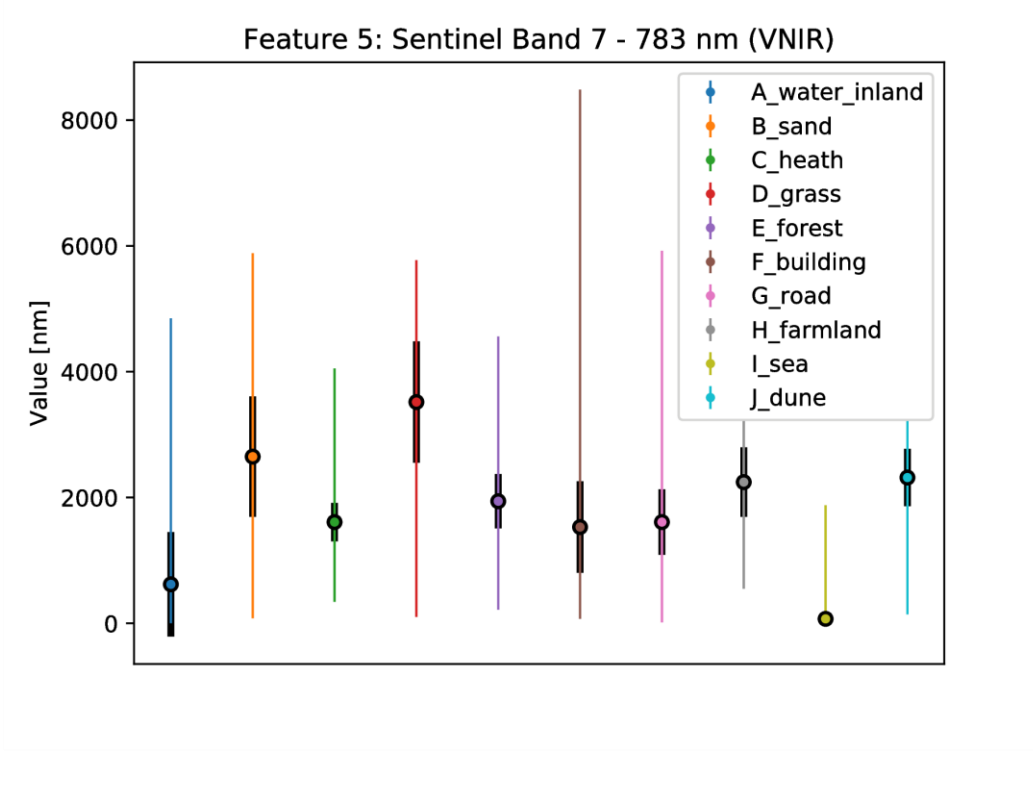


Figure 0.6: Class distribution feature 5

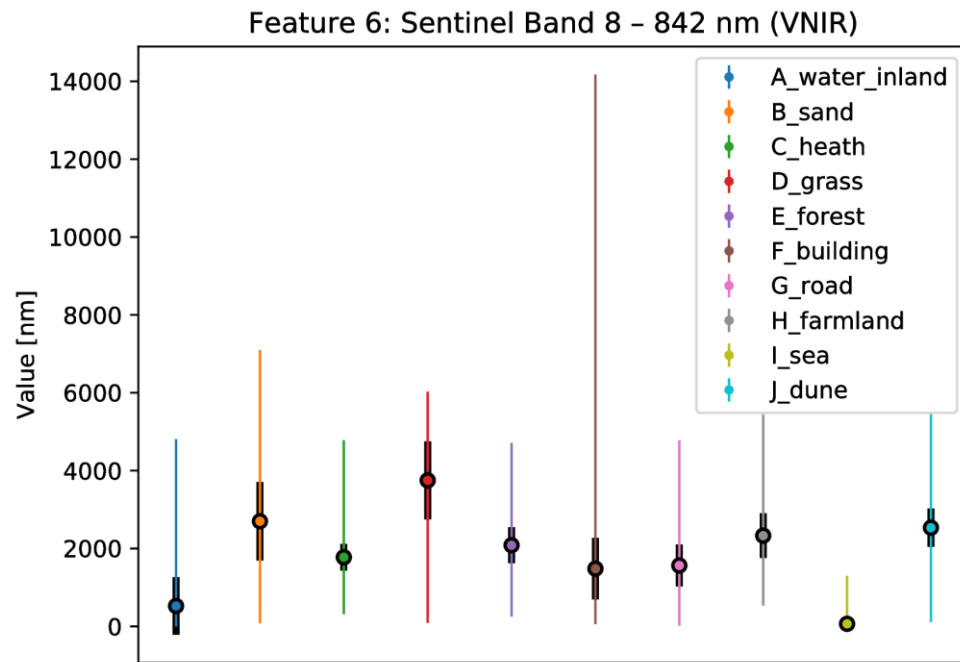


Figure 0.7: Class distribution feature 6

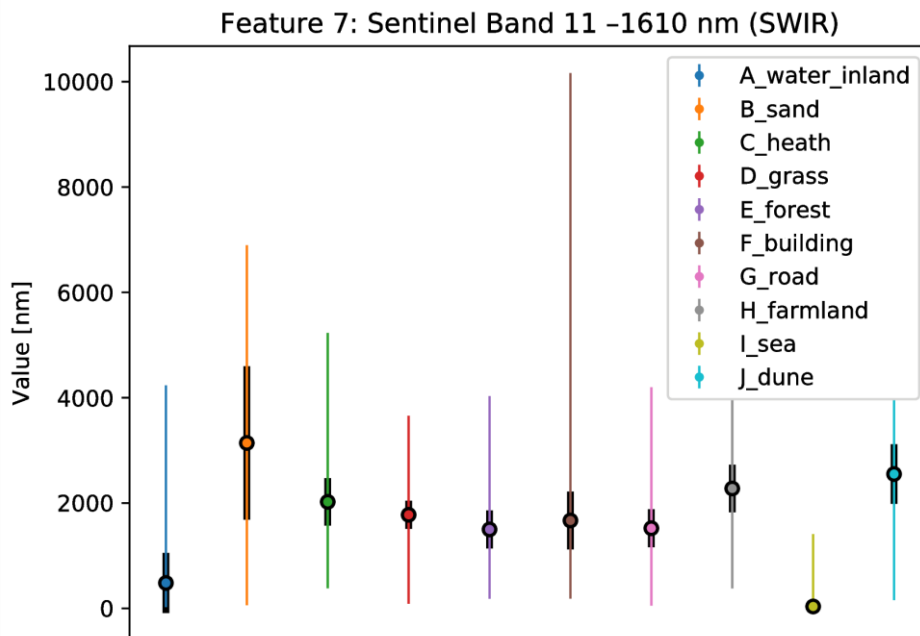


Figure 0.8: Class distribution feature 7

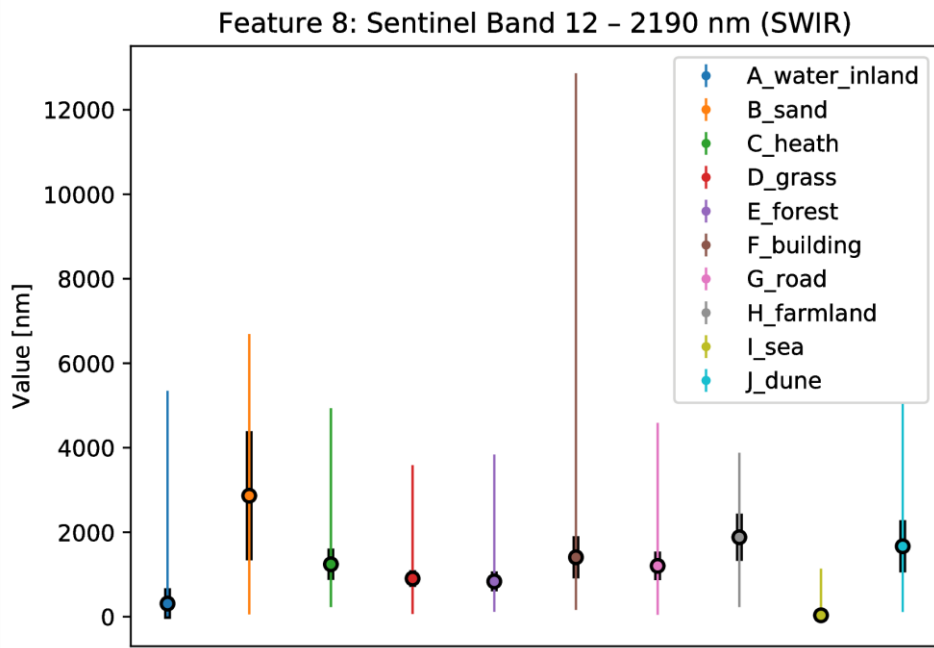


Figure 0.9: Class distribution feature 8

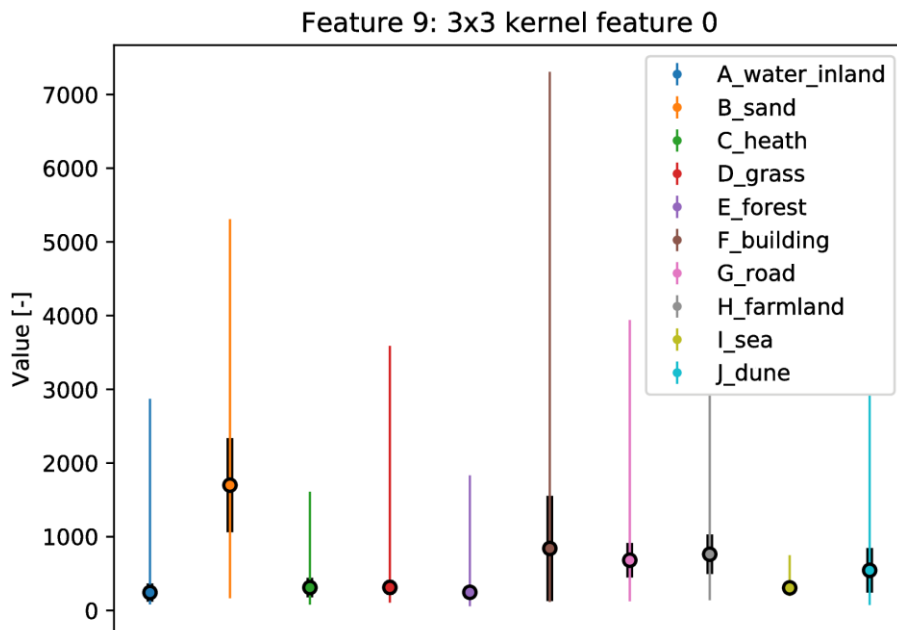


Figure 0.10: Class distribution feature 9

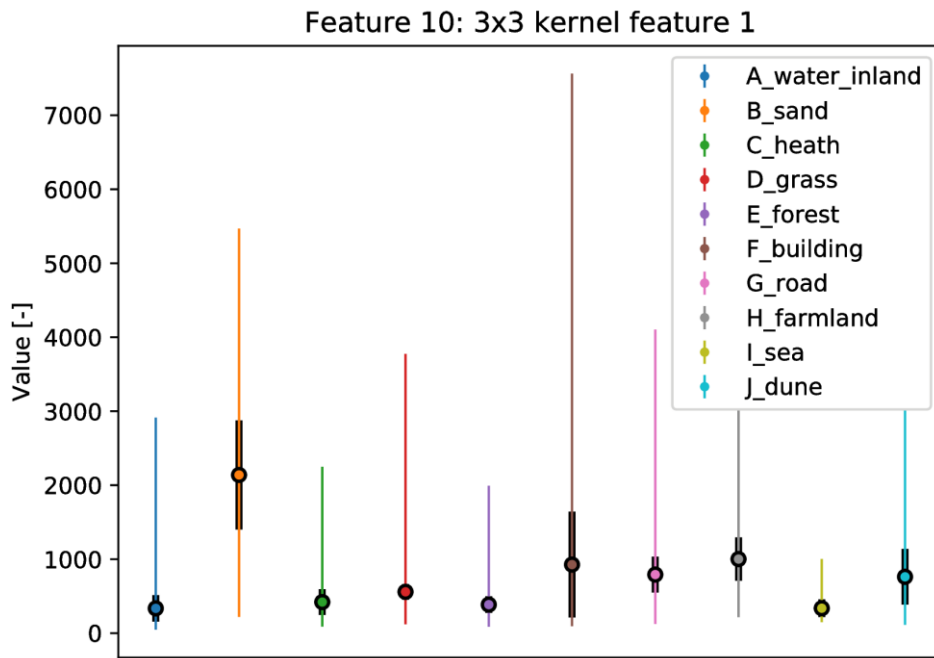


Figure 0.11: Class distribution feature 10

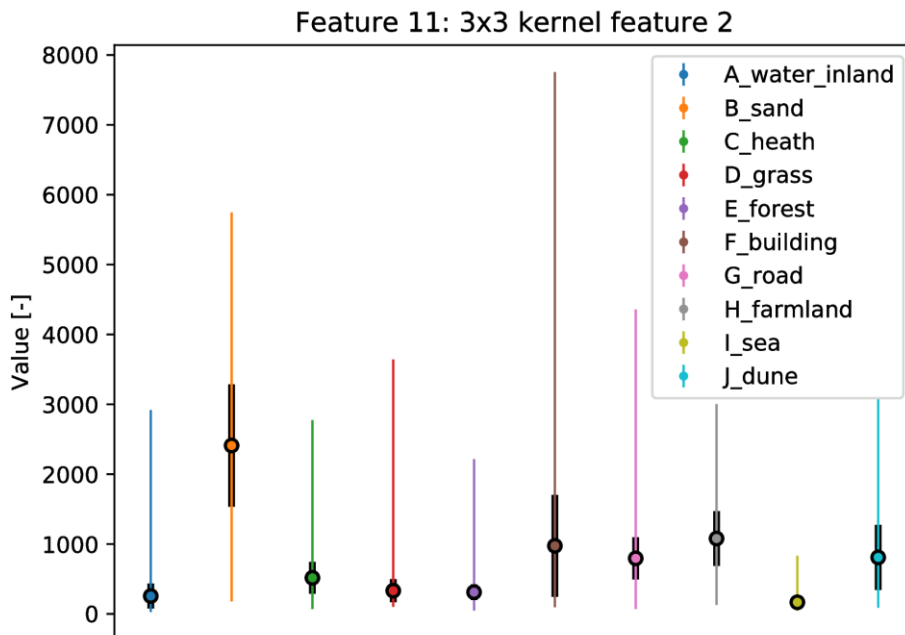


Figure 0.12: Class distribution feature 11

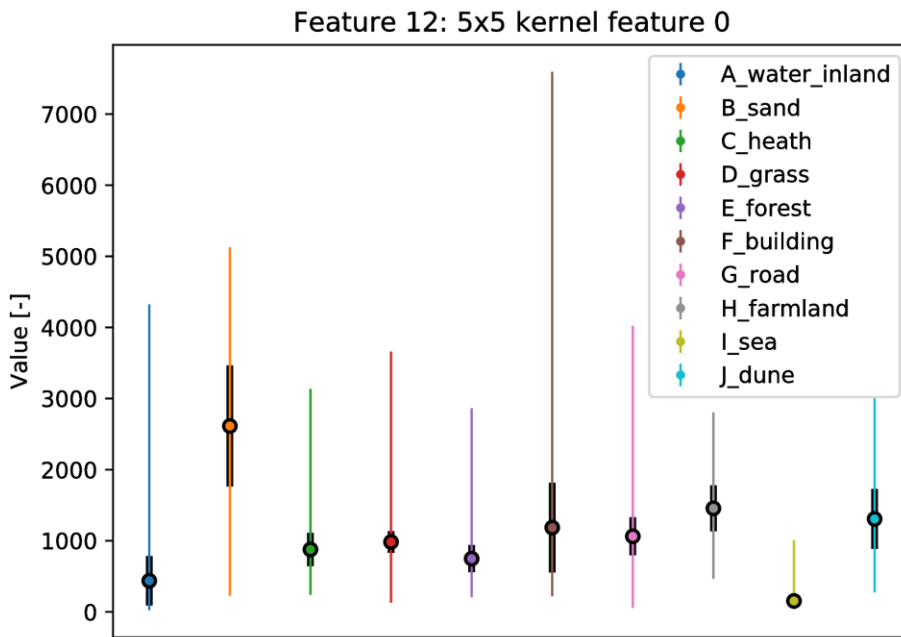


Figure 0.13: Class distribution feature 12

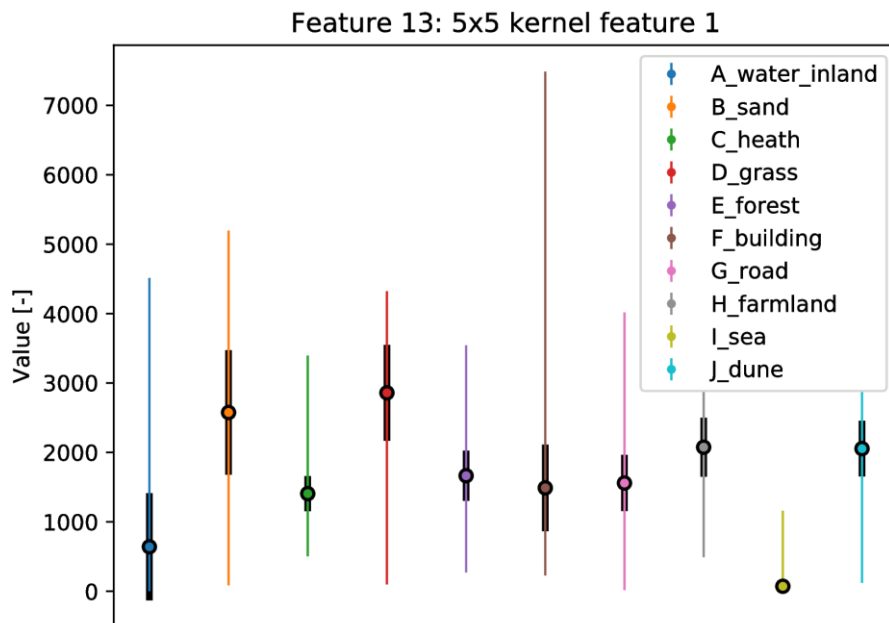


Figure 0.14: Class distribution feature 13

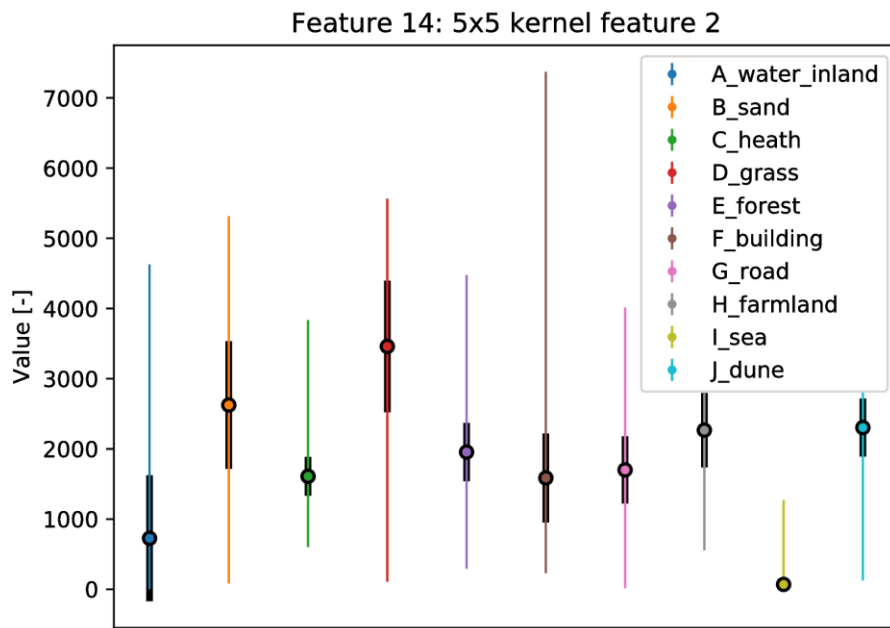


Figure 0.15: Class distribution feature 14

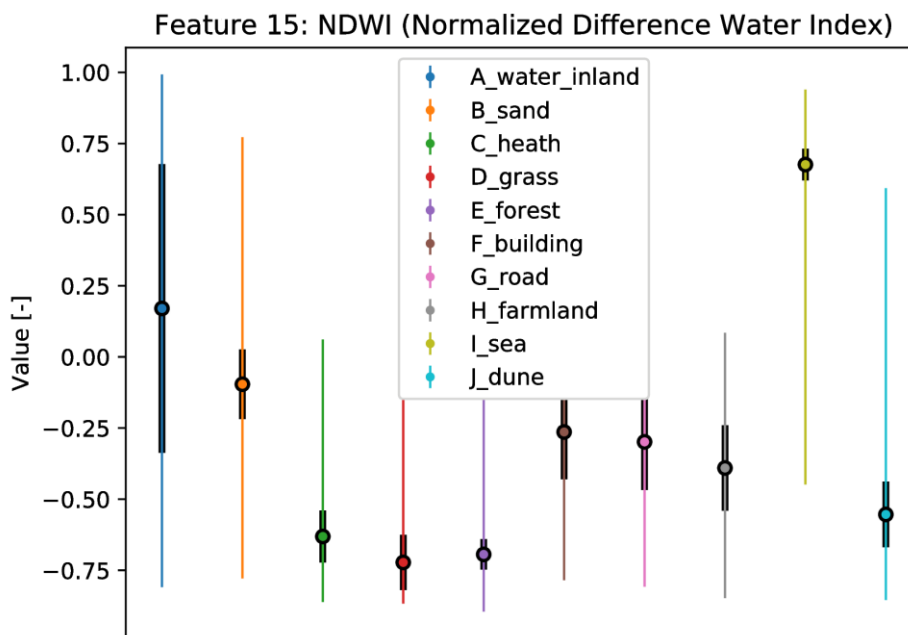


Figure 0.16: Class distribution feature 15

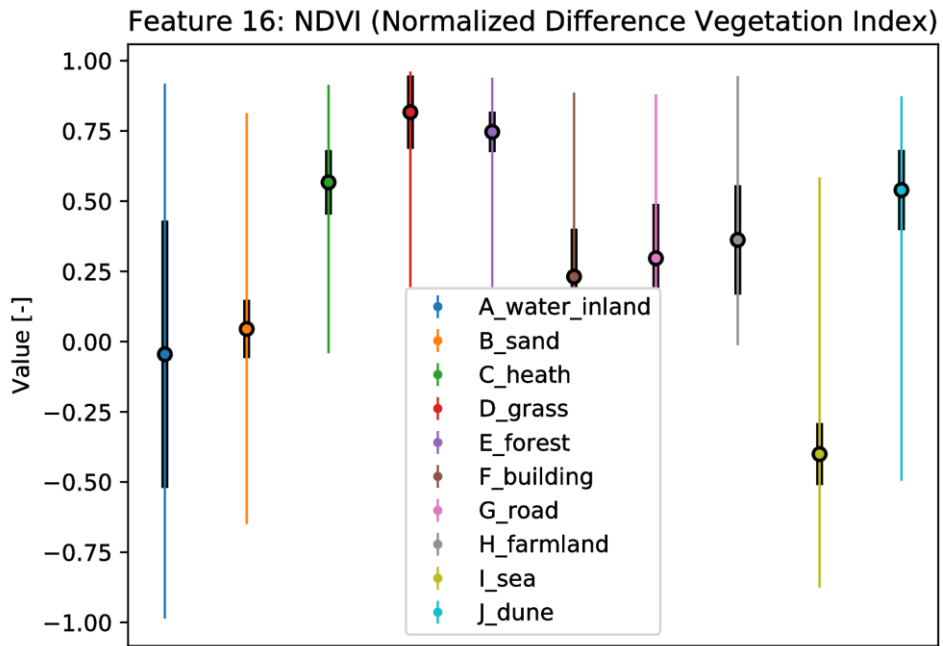


Figure 0.17: Class distribution feature 16

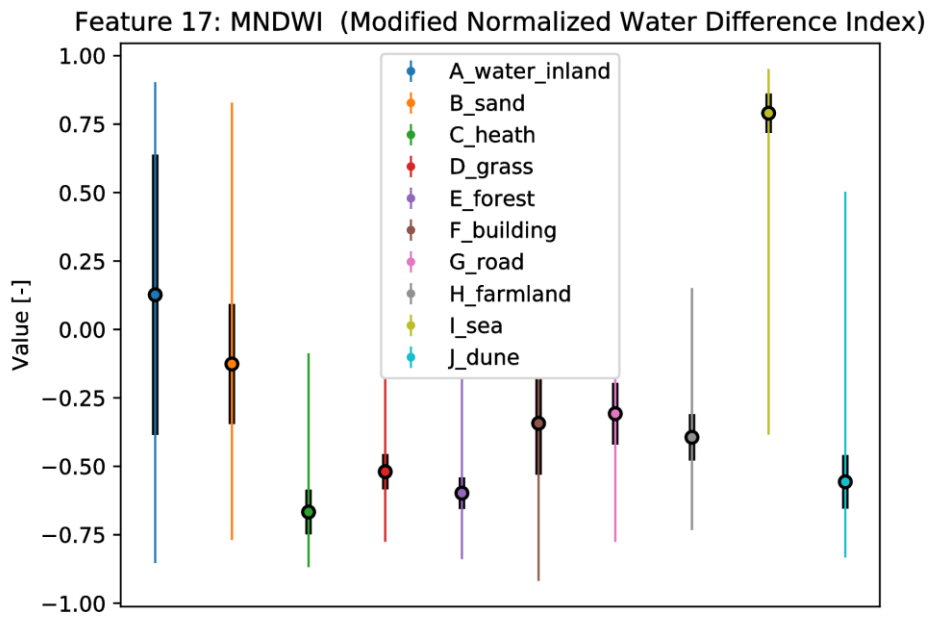


Figure 0.18: Class distribution feature 17



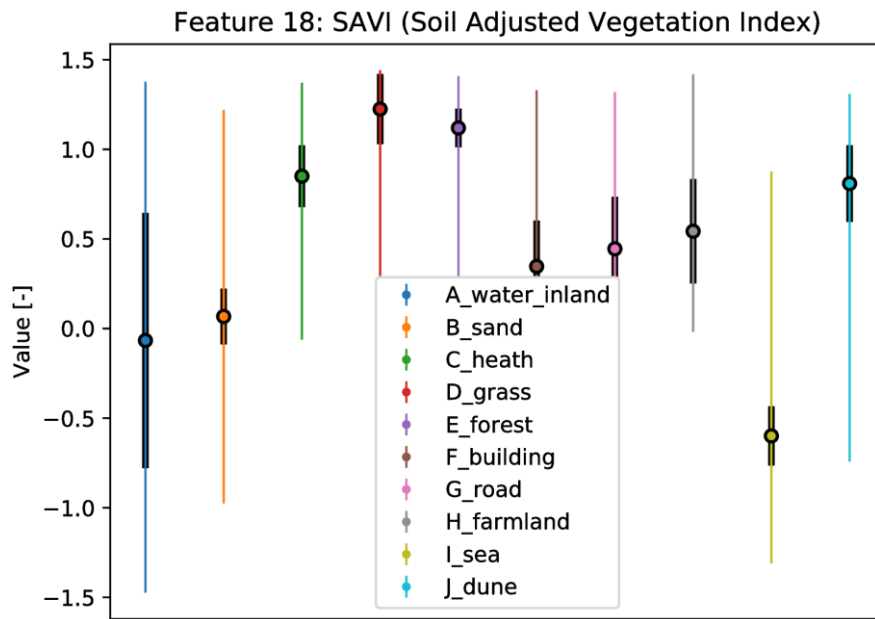


Figure 0.19: Class distribution feature 18

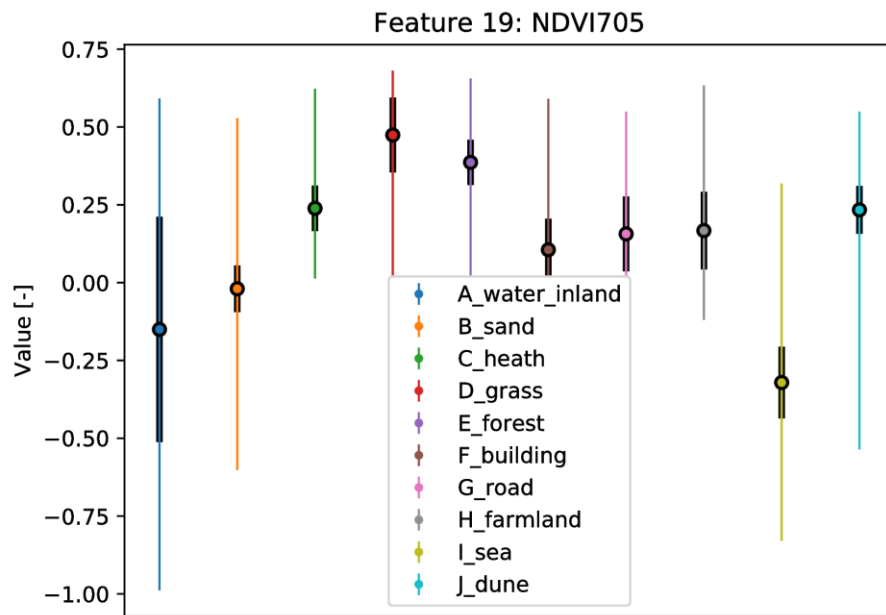


Figure 0.20: Class distribution feature 19

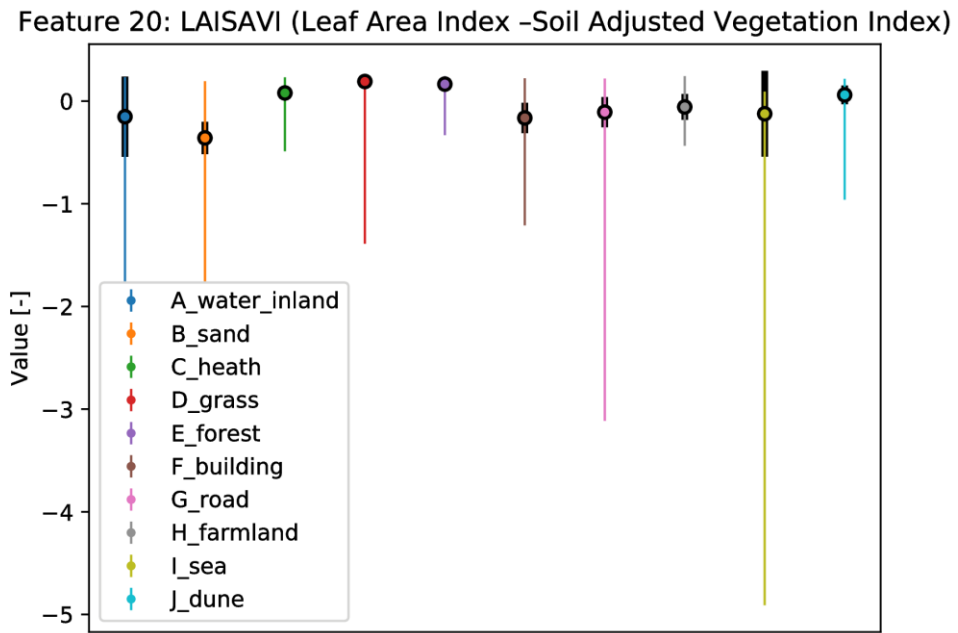


Figure 0.21: Class distribution feature 20

## B - Images

Table 0.1: Images used for running average

Original Image	Cloud		Averaged Image																	
	level		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
S2A_L2A_20151113T105301_T31UFU_R051	1	x																		
S2A_L2A_20151223T105843_T31UFU_R051	2	x	x																	
S2A_L2A_20160312T105037_T31UFU_R051	0	x	x	x																
S2A_L2A_20160401T105024_T31UFU_R051	0	x	x	x	x															
S2A_L2A_20160411T105025_T31UFU_R051	0	x	x	x	x	x														
S2A_L2A_20160421T105029_T31UFU_R051	0	x	x	x	x	x	x													
S2A_L2A_20160501T105310_T31UFU_R051	1		x	x	x	x	x	x												
S2A_L2A_20160511T105343_T31UFU_R051	1		x	x	x	x	x	x	x											
S2A_L2A_20160908T105022_T31UFU_R051	0		x	x	x	x	x	x	x											
S2A_L2A_20170327T105021_T31UFU_R051	0		x	x	x	x	x	x	x											
S2A_L2A_20170526T105031_T31UFU_R051	0		x	x	x	x	x	x	x	x										
S2B_L2A_20170829T105019_T31UFU_R051	2								x	x	x	x	x							
S2A_L2A_20170923T105021_T31UFU_R051	2								x	x	x	x	x	x						
S2B_L2A_20171107T105229_T31UFU_R051	0								x	x	x	x	x	x						
S2B_L2A_20171117T105319_T31UFU_R051	2								x	x	x	x	x	x	x					
S2B_L2A_20171217T105439_T31UFU_R051	1									x	x	x	x	x	x					
S2B_L2A_20180225T105019_T31UFU_R051	1									x	x	x	x	x	x	x				
S2A_L2A_20180302T105021_T31UFU_R051	0									x	x	x	x	x	x	x				
S2A_L2A_20180421T105031_T31UFU_R051	0									x	x	x	x	x	x	x				
S2B_L2A_20180506T105029_T31UFU_R051	0									x	x	x	x	x	x					
S2A_L2A_20180511T105031_T31UFU_R051	2														x	x	x	x		
S2A_L2A_20180521T105031_T31UFU_R051	1																	x	x	
S2B_L2A_20180526T105029_T31UFU_R051	0																			x

## C - System Specification

Table 0.1: Specifications Laptop

Specification	Value
Make / Model	Dell / Latitude E7250
System type	X64 based PC
Processor	Intel Core i5-5300U CPU @ 2.30 GHz, 2295 Mhz, 2 Cores, 4 Logical Processors
Physical Memory (RAM)	16 GB
Virtual Memory	21.1 GB

Table 0.2: Specification External Hard Drive

Specification	Value
Make / Model	LACIE / P9227
Interface	USB 3.1
Storage Capacity	4TB