Monitoring Compliance with the Common Agricultural Policy

Casper Samsø Fibæk

Master Thesis
M.Sc. Eng. Surveying, Planning and Land Management
Aalborg University Copenhagen
June 2017

Zealand 24. February 2017
Image courtesy of the U.S. Geological Survey

Intentionally left blank.

AALBORG UNIVERSITY
STUDENT REPORT

Aalborg University – Copenhagen
A.C. Meyers Vænge 15
2450 Copenhagen SW
www.plan.aau.dk/

Title
Monitoring Compliance with the Common Agricultural Policy

Semester
4th. Semester

Project period
Spring 2017

Supervisor
Jamal Jokar Arsanjani

Page count
71

Preface

This thesis was written in the spring of 2017 as part of the Surveying, Planning, and Land Management programme at Aalborg University in Copenhagen.

In the thesis, I develop an application for gathering satellite data and propose a workflow for monitoring compliance with a subset of agriculture subsidy regulation. The subset comes from an assessment of relevant agricultural laws with spatial requirements described in the European Union's Common Agricultural Policy.

The application is available at https://monitor.trig.dk

Jamal Jokar Arsanjani has supervised the work, and Niras supplied office space and creative input for the duration of the work. I would like to thank Jamal and Niras for all their help.

- *Casper Samsø Fibæk, Maj 2017*

Intentionally left blank.

Abstract

This thesis proposes a system for monitoring farmer compliance with spatial regulatory requirements specified in the European Union's Common Agricultural Policy (CAP). An application for combining satellite data from different sources is created to identify areas of potential non-compliance. Selected imagery from this application is then analysed using remote sensing and machine learning methods.

First, the spatial requirements contained within the CAP are reviewed and then followed by an examination of relevant methods for monitoring these requirements. The thesis investigates publicly available data portals for managing satellite imagery and metadata from the Landsat, Copernicus and ASTER programmes. An application to combine data from these sources and monitor agricultural sites is developed and discussed. The application is hosted online using cloud services and is available at https://monitor.trig.dk

The thesis then considers methods for extending the application to monitor a subset of the regulations specified in the CAP. Two approaches are tested; (1) a method based on vegetation indices and k-means clustering (2) a machine learning approach based on random forest machine learning algorithms. For all tests sentinel, 2B data is used along with field data supplied by the Danish AgriFish Agency.

The result is an online application and alert system for monitoring farmland and a two-tier workflow to determine areas of possible non-compliance by looking at statistical outliers, crop classification and heterogeneity. Conclusively, the thesis suggests the establishment of an open European-wide dataset for multi-season crop ground truth samples and minor changes to the current monitoring workflow.

The project area consists of two Danish islands: Lolland and Falster, located just south of Zealand.

Keywords: Copernicus, API, NodeJS, Agriculture, Regulation, Remote Sensing, Machine Learning, Red Edge, ESA, NASA

Table of Contents

Appendix

Table of Images and Figures

Table of Tables

Monitoring Compliance with the Common Agricultural Policy

M.Sc. Eng. Surveying, Planning, and Land Management.

Casper S. Fibæk, Aalborg University Copenhagen

1. Introduction

On the 7th of March 2017, the European Space Agency (ESA) launched their latest remote sensing satellite Sentinel 2B, as part of the Copernicus Programme. Sentinel 2B joins its twin satellite Sentinel 2A and together they will supply 10m resolution imagery with a revisit time of 2-3 days at latitudes between 30 and 60 degrees (European Spacy Agency, 2015; Mariam-Webster, 2017).

Partially, in anticipation of the launch of Sentinel 2B, the Danish Ministry of the Environment and Food prepared public a tender to capitalise on the new possibilities offered by the decreased revisit time. The tender looks specifically at the possibility of monitoring compliance with the CAP using Sentinel 1, 2 and Landsat 8 (Ministry of Environment and Food - Danish Agrifish Agency, 2017c).

This study is inspired by the topicality of trying to apply these datasets. This thesis does not aim to answer all the questions of the tender nor to solve its listed problems directly. The focus is on creating a prototype system for monitoring compliance with relevant regulation using the new data supplied while combining it with similar data. The end results is a prototype workflow for analysing the data programmatically.

One of the challenging aspects of working with remote sensing is handling the large amount of data (Liu, 2015). This project creates a way of processing that data, without the need for large-scale systems. Furthermore, for usability sake, tools were created for the end users to interact with, both programmatically through Application Programming Interfaces (API's) and a Graphical User Interface (GUI).

Before setting up a workflow, the thesis reviews the relevant regulation in the CAP and makes an assessment on how well the various parts lend themselves to monitoring through remote sensing. Afterwards, a workflow is tested on a subset of the regulations. The workflow uses Machine Learning and remote sensing tools from the Orfeo Toolbox, which is developed by the French Space Agency - CNES (J. Inglada & Christophe, 2009) and the Shark Machine Learning Library developed by Copenhagen University (Igel, Heidrich-Meisner, & Glasmachers, 2008).

First chapter two reviews the spatial regulations of the CAP, and the following chapter describes the controls thereof. The general methods used for creating the monitoring are described in Chapter four. After this, Chapter five describes and tests a workflow for monitoring compliance with the CAP. Finally, chapter six and seven wraps up with a conclusion and a discussion of the findings.

1.1. Problem Statement and Research Questions

"Can medium resolution1 satellite data be used to monitor agricultural areas to assess compliance with EU agricultural regulation in a semi-automatic fashion?"

1) What parts of the Common Agricultural Policy is suitable for compliance monitoring through remote sensing?
2) How can the substantial amount of data supplied by The Copernicus Programme, the Landsat Programme and ASTER, be combined t0 monitor agricultural sites?
3) What kind of workflow would be suitable for supplementing compliance control on a large scale?

[1] Medium resolution defined here as 2 - 30m spatial resolution (SEOS, 2005).

1.2. Project Area

A project area in Denmark was chosen to ensure a manageable scope for the research project and to reduce the size of the data needed for testing. The project area consists of the two islands; Lolland and Falster as well as the minor northern islands in Smålandshavet - "the small island's sea".

The project area is primarily agricultural and surrounded by sea. The area has a long history of growing sugar beets (Sørensen, 2016) which could make the area unsuitable for crop recognition testing. However, since the growing of sugar beets require a two-year in-field gap between the growth of sugar beets (Sukkerroer.nu, 2017); it was deemed suitable for the purpose of this project.

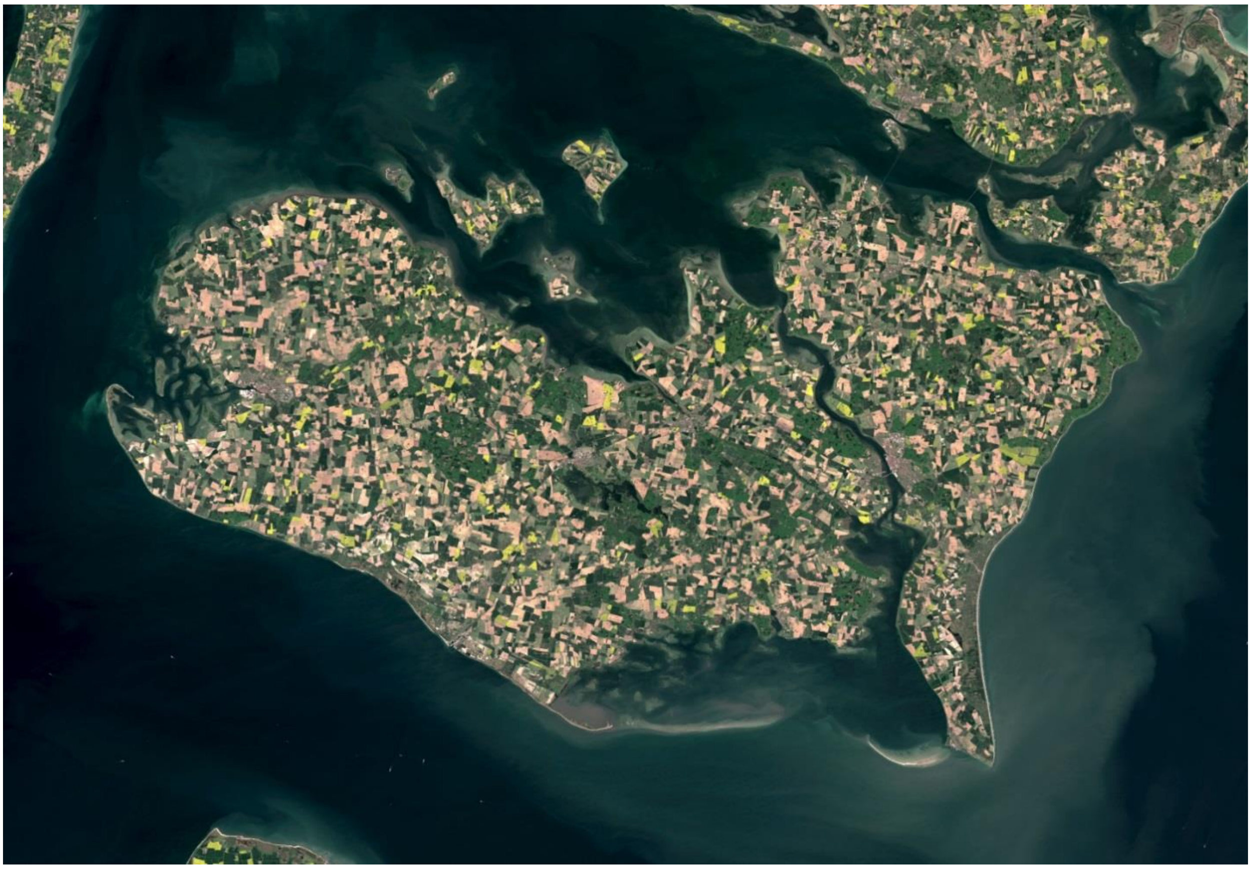


Figure 1.1 The project area, source: Landsat 8 Imagery (LC81950222016133LGN00)

2. The Common Agricultural Policy (CAP)

This chapter provides a brief overview of the CAP and then explores the spatial regulations that farmers must comply with, to receive direct payments from the European Union. The focus on spatial requirements is to assess suitability for remote sensing compliance control. The suitability of spatial conditions will be assessed in chapter three.

2.1. Overview

The European Economic Community[2] first proposed the CAP in 1958 and introduced it into law in 1962. It is a major EU policy with the stated purpose of supporting agriculture and rural communities. It takes up 38% of the entire EU Budget in 2017 and of that percentage; 72% is allocated to direct payments to farmers (Directorate General - Agriculture and Rural Development, 2017; Ludlow, 2005).

Over time, the CAP has gone through many reforms (Ludlow, 2005). Today the CAP consists of two pillars:

I. Support of agricultural production through direct payments, coupled subsidies and market regulation (Lovec, 2016). Here, the CAP specifies a basic payment scheme[3] (BPS) where direct subsidies are paid out to individual farm holdings. To receive these payments, the farmer has to comply with compulsory regulation specified in the Cross Compliance Mechanism (European Commission, 2017b). The basic payment scheme is topped-up by other schemes such as; greening and small farmers payment (European Commission, 2015).

II. The rural development policy. Through the rural development policy, there are schemes defined for agricultural, environmental and climate measures and organic farming. These are voluntary schemes that farmers can undertake over a set period to receive additional subsidies for restoring, preserving or enhancing ecological agricultural environments. In the second pillar, there are also schemes for payments to

[2] ”An institution of the European Union, an economic association of western European countries set up by the Treaty of Rome (1957). The original members were France, West Germany, Italy, Belgium, the Netherlands, and Luxembourg.” –Oxford Dictionary (EEC)

[3] In 2013 the Basic Payment Scheme superseded the Single Payment Scheme

farmers in areas that are hard to farm, such as mountainous regions (Augére-Granier, 2015). Pillar two and the voluntary requirements are only briefly processed in this paper, as they are voluntary and mainly implemented through national rural development programmes (Saraceno, 2003).

2.2. Cross Compliance

Cross compliance is a compulsory scheme consisting of two elements: The Statutory Management Requirements (SMR) and maintaining land in "Good Agricultural and Environmental Condition" (GAEC). Regulation (EU) No 1306/2013 Annex II specifies the overall content of these two schemes. Agencies in the EU member states can define more stringent rules on top of the cross compliance reference or baseline laws (Meyer, Matzdorf, Müller, & Schleyer, 2014). In Denmark, the Minister of Food and Environment is currently in charge of defining these rules (Miljø- og Fødevareministeriet, 2017a).

2.2.1. Statutory Management Requirements (SMR)

There are 18 conditions specified in the SMR's to which the farmers have to adhere. Most of these are not relevant in a compliance monitoring via remote sensing scenario, as they focus on animal welfare, livestock identification and traceability (Meyer et al., 2014).

The below table lists the SMR's that have spatial requirements that are, as such, potentially suitable for satellite monitoring.

	SMR		Spatial Requirements (Banned)
1.	Nitrate Vulnerable Zones (Council Directive: 91/676/ECC)	I.	Distributing organic manure on fields during off-limit periods.
		II.	Applying nitrogen fertilisers when land is: waterlogged, flooded or covered by snow.
		III.	Applying nitrogen fertilisers on soil that was recently frozen.
		IV.	Applying Organic Manure within 10m of surface water or 50m within wells that provide drinking water.
		V.	Storing manure in temporary field-heaps for more than 12 months.
2.	Wild Birds (Directive:	I.	Modifying or damaging protected area.

	2009/147/EC)	
3.	Conservation of Fauna and Flora (Council Directive: 92/43/EEC)	I. Modifying or damaging protected area.

Table 2.1 Statutory Management Requirements (SMR)

As baseline laws have to be implemented similarly across EU member states (Meyer et al., 2014), only the baseline regulations mentioned above will be processed and not the more stringent rules that could be imposed by individual member states.

2.2.2. Good Agricultural and Environmental Conditions (GAEC)

The CAP regulation only specifies the overall guidelines of the GAEC. Below is an abbreviated extract of the guidelines as defined in Regulation (EU) No 1306/2013 Annex II.

	GAEC Standards
1.	Establish buffer strips along water courses.
2.	Water used for irrigation must follow authorisation procedures.
3.	Protect groundwater against pollution.
4.	Regulate minimum soil cover.
5.	Regulate land management to limit erosion.
6.	Maintain soil organic matter; ban on burning arable stubble except for plant health reasons.
7.	Retain landscape features; ban on cutting hedges and trees during the breeding and rearing season for birds.

Table 2.2 Good Agricultural and Environmental Conditions (GAEC)

The actual implementation of these guidelines is up to the individual member states, and their relevancy for remote sensing monitoring would, therefore, have to be assessed at a national level (Meyer et al., 2014).

In 2015, the Danish consultancy COWI created a report on the Danish Cross Compliance implementation. They compared the Danish implementation with the implementations in a series of other European countries; Sweden, Germany (Lower Saxony and

Slesvig-Holstein), the Netherlands, Poland, and France. The purpose of the report was to give the Danish Government knowledge of the implementation of Cross Compliance in other European countries. The interest in this was due to the nature of the baseline laws specified in the CAP, that meant that the national implementations of the CAP and the cross compliance mechanism vary widely, especially in regards to the GAEC (COWI, 2015).

In the report, COWI specifies that:

> *"Generally speaking, Denmark has CC[4] requirements in the same areas as the other EU countries and Länder studied. A number of requirements in Denmark are more specifically formulated than in the other countries which use terms that are more general, for example requirements for water abstraction, conservation of landscape features and the design of facilities for the storage of livestock manure."*
>
> - (COWI, 2015, p. 24, paragraph 2)

For the purpose of this study, the Danish implementation of the GAEC standards and their relevance for remote sensing will be used (Miljø- og Fødevareministeriet, 2017b).

The Below table lists the implemented GAEC in Denmark that has explicitly stated spatial requirements for agricultural fields (Miljø- og Fødevareministeriet, 2017b). It is an abbreviated table that only serves to highlight relevant spatial requirements.

#	GAEC Spatial Requirements
1.	It is illegal to spread fertiliser in two-meter buffer strips along open nature stream and lakes above 100m^2.
2.	GAEC 2 has no spatial requirements.
3.	Storing fertiliser in-field is illegal without authorization. If authorised, it is not allowed to risk contaminating ground water, and you are not allowed to drain water to lakes over 100m^2 and coastal waters. If storing organic manure in field, the following spatial requirements apply:

[4] Abbreviation of Cross Compliance.

I. 50m distance to common water supply services.
II. 25m distance to non-common water supply services.
III. 15m distance to streams and lakes over 100m^2

You can store compost of at least 30% dry matter in the field following the manure distance requirements. Compost cannot be kept in the same place more than 12 months, and the same spot cannot be used to store compost for another five years.

---

4. For farms earning more than 50.000,- DKK and having arable land above 10 hectares: The farmer must establish catch crops in the Autumn season. There are two brackets for the required percentage of catch crops on land following crops that do not absorb nitrogen during autumn: 10% or 14% depending on the amount of organic fertiliser used for the field (Ministry of Environment and Food, 2016).

Catch crops must be established no later than the 1st of August, however, for some types no later than the 20th of August.

Catch crops, except crops following maize, cannot be ploughed, withered or destroyed before the 20th of October.

Various schemes for alternatives to catch crops are made possible in the Danish regulation.

---

5. Making changes to areas located in Natura2000 areas is illegal.

Areas that apply for fallow status must be held covered with plants.

---

6. It is banned to burn arable stubble except for plant health reasons.

From harvest until the 15th of February it is illegal to plough your field if all the below requirements are met:

I. It is a part of a connected area of more than 5000 m^2.
II. It is slanted more than 12 degrees.
III. It has a high risk of erosion by run-off.
IV. It is a part of a field block that applied for direct payments.

---

7. Retain protected landscape features including lakes and ancient monuments.

It is illegal to cut hedges and trees during the breeding and rearing season for birds; 15th of March to 31. July.

Table 2.3 The implemented spatial GAEC in Denmark.

Temporal requirements for fertilisation periods is excluded from the above table; this is because examples of satellite-based detection of the spread of fertiliser on fields were not found during the literary review.

2.3. Greening

To incentivise farmers to use farmland more sustainably, a 2013 reform of the CAP introduced the regulatory concept of greening. It is required by member states to allocate at least 30% of the budget for direct payments for greening measures (European Commission, 2017a). It is compulsory for farmers, and to be eligible to receive greening funds, the farmers also must adhere to cross compliance. Chapter 3, Article 43-46 of EU Regulation 1307/2013 specify the rules surrounding greening. The following subchapter summarises that regulation.

2.3.1. Crop Diversification

Article 44 specifies requirements for crop diversification. The purpose of crop diversification is to ensure environmental benefits and making the soil and ecosystem more resilient (Bio Intelligence Service, 2010).

The table below summarises the crop diversity requirements:

Total arable land	Requirement
Less than 10 ha	I. There are no requirements for crop diversity.
Above 10 ha and below 30 ha	I. At least two crops from distinct categories. II. The main crop cannot cover more than 75% of arable land.
Above or equal to 30 ha	I. At least three from distinct categories. II. The main crop cannot more than 75% of arable land. III. Two main crops combined cannot cover more than 95% of arable land.

Table 2.4 Crop diversity requirements (European Commission, 2017a)

The member states set their period of control, but the EU guidelines say that a period of three months is appropriate. In Denmark, the control period for crop diversity is 15th of May until the 25th of July. The crops found in the farmer's field during this time serve as the basis for testing adherence (European Commission, 2016).

2.3.2. Permanent Grasslands

The national governments designate areas for permanent grasslands in environmentally sensitive areas as stated in Article 45. Permanent grassland is agricultural land that has been left as grassland for at least five years. Farmers may not plough or convert these areas.

The amount of permanent grassland to the total claimed agricultural area cannot fall below five percent (European Commission, 2017a). The baseline law states that this ratio is calculated on a national level, but the member states can choose to do the calculation at holding level. If the national level is met, the government can decide to not implement mandatory requirements for its farmers (Paragraph 2, Article 45).

2.3.3. Ecological Focus Areas

If a farm has more than 15 hectares of arable land, the farmer must dedicate at least 5% of arable land to ecologically beneficial elements. Below is a list of some of the approved types:

- Fallow land
- Field Margins
- Hedges and trees
- Voluntary buffer strips

It can also be the establishment of catch crops or nitrogen-fixing crops. Catch crops are crops that a planted in between two main crops to prevent leaching of nitrate in the soil (Lockhart & Wiseman, 1983). Catch crops are especially interesting from a spatial viewpoint as they spring up during a period where the field would otherwise be bare.

It is up to the member states to select what constitutes ecological focus areas by selecting from the list specified in article 46, paragraph 2.

2.4. Other schemes

Besides the Basic Payment Scheme and Greening, there are other systems for direct payments to the farmers. Some of these do not have direct spatial components but are included here to provide a more comprehensive overview of the CAP.

The redistributive payment allows the governments to distribute more funds to the first hectares of a field i.e. the first 30 hectares receive an added payment, effectively a voluntary support system to small farmers (European Commission, 2015, p.7).

Although the CAP was decoupled from the growth of certain crops in 2003, it is still possible for member states to implement voluntary coupled support for certain crops or animal products. This allows EU member states to aid regions that are having specific difficulties. In countries where this rule is implemented, it would be beneficial to have a remote monitoring system to control whether the applied for coupled crops are the ones being grown (European Commission, 2015, p. 9-10).

Lastly, there is a payment for areas with natural constraints such as sloping areas. This scheme is only implemented in Denmark and Slovenia (European Commission, 2015, p. 10).

2.5. Regulation with Spatial Requirements

The table below summarised the above regulations and the spatial requirements. The following chapter then reviews the control efforts and how these requirements might be monitored using medium resolution satellite imagery.

Scheme	Rule	Control Type	Requirement
CC	SMR 1	Detect field heaps Change detection	Storing manure in temporary field-heaps for more than 12 months.
CC	SMR 2	Change detection	Modify or damage protected area.
CC	SMR 3	Change detection	Modify or damage protected area.
CC	GAEC 1	Distance calculation	Buffer strips along streams and lakes.
CC	GAEC 3	Detect fertiliser storage in field	Fertiliser stored in field is illegal.
		Distance requirements Detect organic manure	Keep safe distance from organic manure stored in-field to water.
		Detect field heaps Change detection	Stored compost cannot stay in the same place for more than 12 months, and the spot cannot be used for next five years.
CC	GAEC 4	Detect ploughing Crop detection	Must establish catch crops before a set date.
		Detect ploughing Crop detection	Cannot plough or destroy catch crops before a set date.
CC	GAEC 5	Change detection	Modify or damage protected area.
		Crop detection	Fallow land must have cover crop.
CC	GAEC 6	Detect burned stubble	Ban on burning arable stubble.
		Detect ploughing Slope detection	Ploughing is illegal before set date is illegal if land is erosion prone.
CC	GAEC 7	Change detection	Retain landscape features and must not cut hedges and trees during rearing season.
Greening	1	Crop identification	Crop diversity requirements.
Greening	2	Crop identification Change detection	Must retain permanent grass.
Greening	3	Crop identification Distance calculation	Must retain ecological focus area.
Voluntary Coupling		Crop identification	Funding for growing specific crop.
Natural Constraints		Slope detection	Funding for sloping land.

Table 2.5 CAP regulation with spatial requirements potentially suitable for remote sensing.

3. Compliance Monitoring

This chapter delves into the rules surrounding the handling of the compliance control in the European Union. First, the legal framework is described, and then the distinct types of spatial requirements are investigated regarding their suitability for remote sensing control. The result of the chapter is a selection of a subset of the regulation for further testing and implementation into a monitoring workflow.

3.1. Legal Framework for Control

Title V and VI of (EC) No 1306/2013 states the central rules for the control of the CAP and the intentions. The purpose of the rules is to protect the financial interests of the European Union and compel the member states to monitor compliance systematically. The implementation rules are based on (EC) No 809/2014 and a supplementary ruleset (EC) 640/2014.

A minimum of one percent of all applicants must be controlled by either in-field inspections or remote sensing (1606/2013 article 59, 2). The selection of applicants for control is based, in part on a risk-based assessment and a random control sample. This is to ensure a representative error rate. The random sample should comprise between 20% to 25% percent of the total inspections. If inspections reveal significant non-compliance, the number of inspections should increase for the following period. In Denmark, this means that the control encompasses approximately 2.000 farms and 35.000 parcels each year (Ministry of Environment and Food - Danish AgriFish Agency, 2017).

The member states choose which method to use for the inspections: Either in-field inspections or remote sensing control. The conditions for receiving subsidies that must be controlled are listed in (Article 37 in 809/2014) and includes:

- Area measurement
- Declared land use vs. actual land use
- Cross compliance checks
- Minimum activity
- Greening

If the control method is remote sensing and it was not possible for the agency in charge of compliancy monitoring to verify all parts of the regulation – the agency carries out a supplementary field-visit called a 'Rapid Field Visit'.

3.2. Control by Remote Sensing

In Denmark, the Danish AgriFish Agency has experience using remote sensing for control by using Very High Resolution[5] (VHR) satellite imagery (Chellasamy, Ferré, & Greve, 2016). They are currently investigating the use of mid-resolution imagery to supplement the control (Ministry of Environment and Food - Danish Agrifish Agency, 2017c).

> *"At Danish Agrifish Agency, we are very excited and optimistic about the increased opportunities that come with the Sentinel satellites. The possibilities for using EO data within the agricultural control are significantly increased and combined with the use of state-of-the-art methods and complex processing the applications are promising."*
>
> **- Sanne Eskesen, Project Manager at Danish Agrifish Agency in (Nyborg, 2017, p. 1)**

The Danish VHR methodology appears to correspond with the methods described by the European Commission's Joint Reseach Center – Monitoring Agricultural Ressources (EU Science Hub, 2017).

The CAP regulation does not contain extensive requirements for the execution of remote sensing control. Article 70, 809/2014 briefly states: "Where appropriate, the on-the-spot checks may be carried out by applying remote-sensing techniques". Article 40 elaborates on how remote sensing might be applied:

A. Perform photo interpretation [...] to recognise land cover types, crop type, and measuring area.
B. Carry out physical inspections (rapid field visits) when the results are not conclusive.
C. Make all checks to verify compliance with the obligations of the parcel.
D. Take alternative action to measure area when imagery does not cover the area.

[5] Imagery with a Ground Sampling Distance of less than 0.75m.

The European Commission can provide the control authority of member states with VHR imagery (Article 21 of 1606/2013). The commission has several framework contracts with a range of imagery provides. The Commission also uses the data to create European-wide crop yield estimates (Article 22b of 1606/2013). The member states can use the G4CAP application to order and access these images (Matteo, 2017). G4CAP also provides access to Sentinel 2 imagery and recently implemented an alert system for sentinel 2 imagery, when a new image for a control zone is available (Matteo, 2016).

3.3. Control Type Assessment

As stated in article 70 (EC) 809/2014, it is not possible to replace the in-field inspections with automated remote sensing applications, unless it is possible to achieve the same accuracy. However, it is possible to supplement the current checks, especially regarding efforts to find at-risk areas for control zones and areas were rapid field visits could reveal non-compliance.

The following subchapters review the different control types found in Table 2.5 CAP regulation with spatial requirements potentially suitable for remote sensing..

3.3.1. Distance calculations

Article 38 in 809/2014 specifies the tolerance levels for area measurements; the geometric accuracy must be at least two meters. This requirement alone excludes using mid-resolution imagery for area measurements. VHR imagery would be better suited for area measurements. For other distance calculations, such as 50-meter distance to wells providing drinking water, mid-resolution imagery might be useful. However, since the relative geodetic accuracy of Landsat 8 is ~ 20m (Storey, Choate, & Dekota, 2014) and ~ 5m for Sentinel two (Vajsova & Åstrand, 2015) these images are unsuitable for most distance calculation required by the regulation.

3.3.2. Organic manure, field heaps, and fertiliser storage

Automatic detection of the various kinds of heaps and storages could be difficult because of their diverse forms and compositions. The size of the heaps might also be too small for mid-resolution satellites to detect correctly. If during an on-the-spot visit an inspector detects a heap or storage, they could flag it for change detection monitoring to be reviewed during the same period the following year. If no change is detected the farmer could be non-compliant with the 12-month period specified in SMR 1 and GAEC 3.

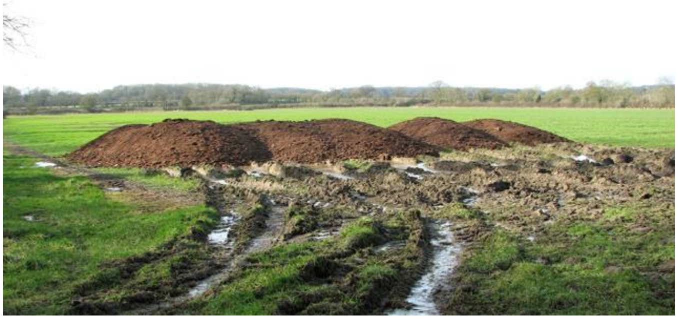


Figure 3.1 Field heaps of manure. Source: Evelyn Simak

3.3.3. Burned arable stubble

Detecting stubble-burning with remote sensing is well described in the literature, both using sentinel 1 and 2 (Verhegghen et al., 2016) and using lower spatial resolution satellites such as MODIS (Smith et al., 2007). The red edge bands of the sentinel 2 are especially useful in determining burn severity (Fernández-Manso, Fernández-Manso, & Quintano, 2016). In the case of applying the research to monitoring the GAEC (6) requirement, it would be beneficial to acquire a ground truth data set of burned stubble to make the detection automatically using a trained classifier (Yadav et al., 2014). MODIS and VIIRS[6] supply an active fires dataset at 1km and 365m resolution (NASA, 2017). However, 365m resolution is most likely low for most field in Europe. Without a ground truth dataset, investigating sudden drops in the values of vegetation indices could be indicative of stubble burning.

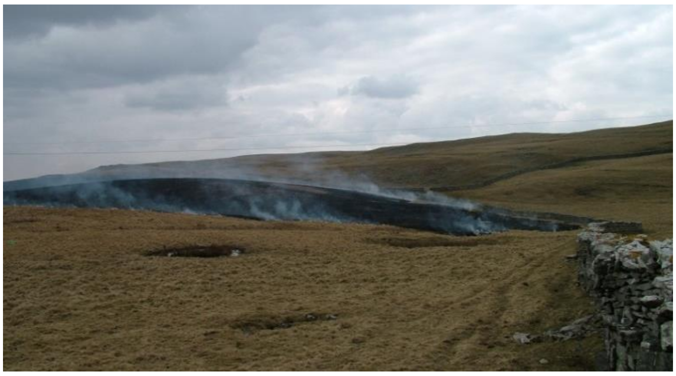


Figure 3.2 Stubble burning in England. Source: David Brown

[6] Visible Infrared Imaging Radiometer Suite – A weather satellite.

3.3.4. Change detection, ploughing and grass cutting

Change detection using optical satellite is currently being used to monitor various sites such ecologically protected areas (Willis, 2015), protection for illegal logging (Ya'acob et al., 2014), and urban land use change (Hegazy & Kaloop, 2015). Much research into land cover change detections uses the MODIS satellite due to its high revisit time and wide swath (Toller & Isaacman, 2009). With the launch of Sentinel 2B, the constellation could prove an attractive tool for detecting land use changes due to its high revisit time. The smaller swath of 290km versus 2330km would increase the data required (European Spacy Agency, 2015). However, the detection method would benefit from the increased spatial resolution.

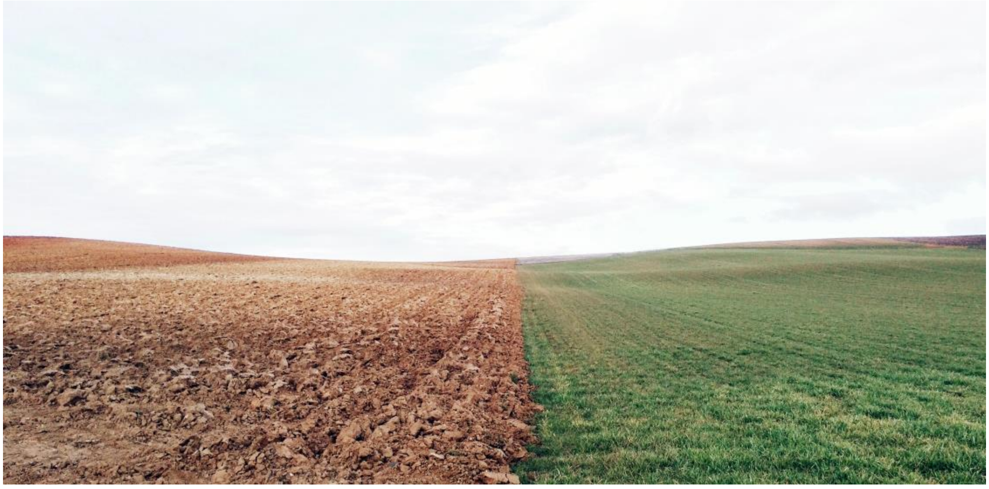


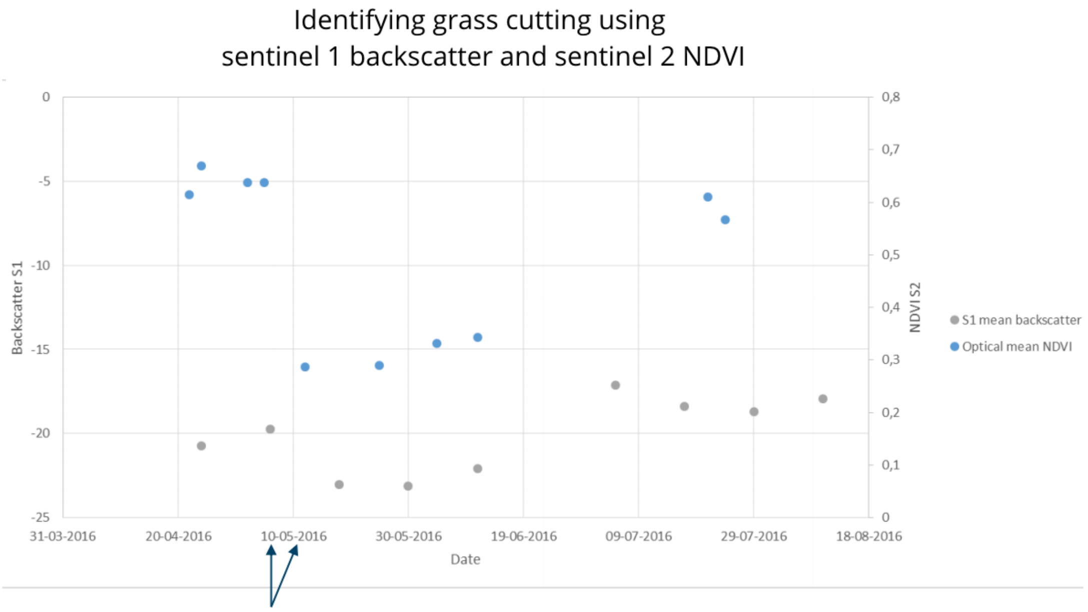
Figure 3.3 Ploughed field in Zamora, Spain. Source: User Unsplash from Pixabay

The Sentinel one constellation is an "[...] imaging radar mission providing continuous all-weather, day-and-night imagery at C-band" (European Spacy Agency, 2013, p. 36). The Sentinel one constellation has been used for change detection of wetlands (Muro et al., 2016). On the 7th of April 2017 DHI Gras, a Danish earth observation focused company, successfully demonstrated the application of sentinel 1 and 2 data to detect ploughing and grass cutting in Denmark to monitor CAP compliance (Nyborg, 2017). DHI did their work as part of the tender mentioned in Introduction.

The method described by DHI was presented at the Copernicus Training and Information Session in Aarhus the 9th of March 2017. It looks at two primary ways of detecting ploughing (Nyborg & Eskesen, 2017).

1) Rapid changes in Sentinel 1 radar backscatter values
2) Rapid drop in Sentinel 2 NDVI values

The presentation notably did not include NDVI values from Landsat 8 or ASTER data, which could be due to the theme of the conference.



Grass potentially cut during this period

Figure 3.4 Identifying Grass Cutting using Sentinel 1 and 2.
Source: Slide from joint DHI and Danish AgriFish Agency presentation (translated).

Ecologically protected areas, as well as fields during periods of special restrictions, could be continually monitored for changes using a combination of satellites with an attached alert system. Such a system would pertain to many CAP regulations in 2.5. Regulation with Spatial Requirements. The Sentinel Application Platform (SNAP) includes tools for change detection using Sentinel 1.

3.3.5. Crop detection

GAEC four and five sets requirements for the establishment of catch crops, cover crops, and minimum soil cover. This could be monitored by continually monitoring vegetation indices and comparing their values during prohibition periods (Bannari, Morin, Bonn, & Huete, 1995).

3.3.6. Declared crop type and identification

The farmer declares the main crop and catch-crops on his fields for each season. Many of the CAP requirements, especially greening and coupled payments, are highly dependent on these being correct. Much research has been done to establish crop identification platforms, and it is the subject of continuous work on improving crop type classification integration of Landsat and Sentinel 2 data (Hansen, 2017). Sentinel 1 data can be used in combination with optical satellites to improve the crop identification (Jordi Inglada, Vincent, Arias, & Marais-Sicre, 2016).

The declared field-parcel might not match in crop type, and it is also possible that the parcel is drawn incorrectly or multiple types of crops are grown on the same parcel. Finding these incorrect fields could be done by looking at the internal variance or heterogeneity in vegetation indices values of fields or by segmentation analysis on declared crops. The Orfeo Toolbox contains multiple segmentation algorithms along with unsupervised classification methods such as ISODATA or K-Means, which could also be utilised to segment the fields (J. Inglada & Christophe, 2009).

In the creation of a monitoring workflow in chapter 5. A Case Study Workflow, the problems of determining the declared crop type and crop heterogeneity will be the main focus. Providing a workflow to more effeciantly monitor these two issues would be beneficial to the controlling agencies, as they are time consuming to control with the current VHR imagery methodology (Ministry of Environment and Food - Danish Agrifish Agency, 2017c).

3.3.7. Slope detection

The CAP specifies special restrictions for erosion prone areas that are sloping. For the project area, the average field-parcel slopes 1.12 degrees. This was calculated as follows

$$\text{Weighted average slope: } \sum_{i=1}^{n} s_i * \frac{a_i}{\sum_{i=1}^{n} a_i}$$

Where s_i is the slope of each field and a_i is the area of said field

using the 2015 Danish DEM. Calculations on the DEM reveal that only one plot lives up to the requirements specified in the CAP. This requirement does not seem to be a problem for the project area, but other areas in Europe would be more affected.

Controlling for this issue could be a combination of a DEM with a control of vegetation indices for the minimum soil cover required to ensure no exposed soil.

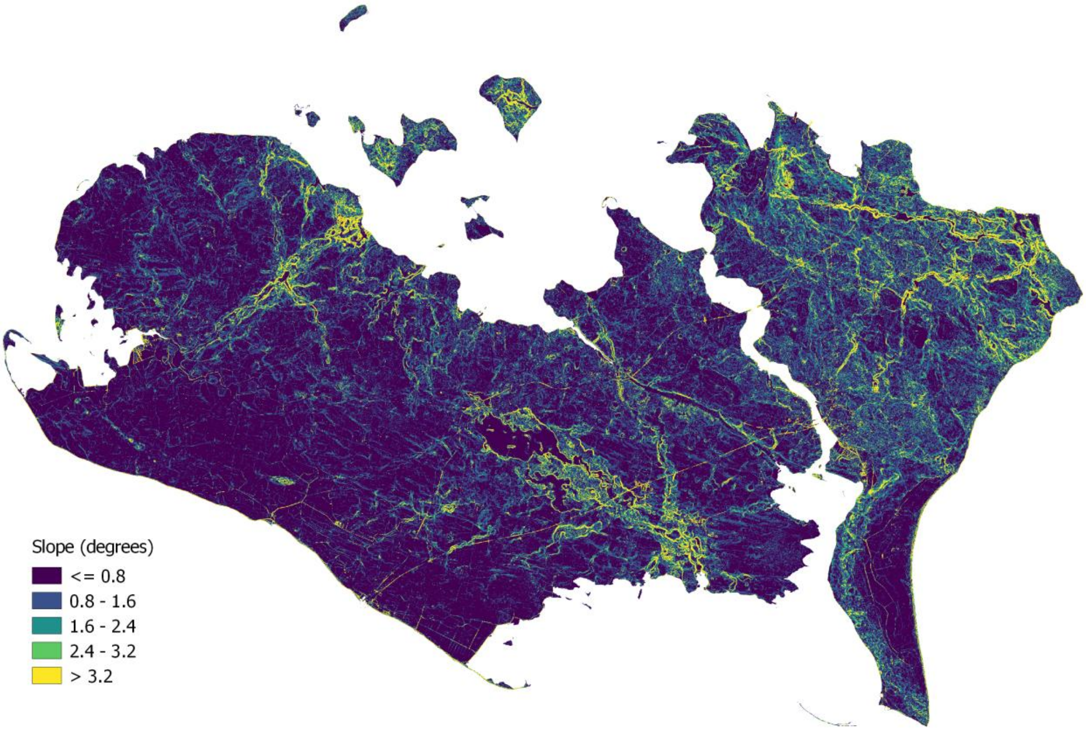


Figure 3.5 Slope map of the project area

4. Monitoring and Data Management

The first part of this chapter is a review of tools available to access Landsat, Sentinel and ASTER data. Next, the chapter considers the creation of a management tool that can combine multiple sources and help automate the monitoring process.

There is a substantial amount of data supplied by remote sensing satellites. In 2016, there were 374 active satellites in orbit with the stated purpose of remote sensing. Of these; 165 have optical imaging as the primary objective, and 34 have radar imaging (Lavender, 2016). Projections show that at the end of 2016 the Copernicus programme alone generated six terabytes of data a day (Bargellini & Laur, 2016) with a recent tweet indicating that this number is now doubled (ESA EarthObservation, 2017). Simulations of Sentinel 2A and B show that a site can expect at least one cloud-free image per month (Hagolle et al., 2015). Combining Sentinel two with other sources will provide a better chance for more cloud-free images. NASA is currently working on harmonising Sentinel 2 and Landsat 8 data, which will ease monitoring and machine learning efforts (Claverie, Masek, & Ju, 2016).

To handle the raw amount of data generated requires large-scale systems. The primary access point to Copernicus data is through the Copernicus Open Access Hub[7]. The Copernicus Hub also supplies programmatic access, but does not host non-ESA imagery. Therefore, should you wish to combine multiple sources of data, you are required to check multiple portals or using a different portal that provides the access required.

The European Commission's Joint Research Centre offers information and tools regarding remote sensing for the controlling agencies. One tool is the G^{4}CAP application mentioned in 3.2. Control by Remote Sensing. Besides VHR imagery, the tool is also able to download sentinel 2 imagery and includes an alert system for when new Sentinel two images are available for a control area (Matteo, 2016).

[7] https://scihub.copernicus.eu/

4.1. Review of Monitoring Tools

Before developing a new monitoring tool, this chapter examines tools already available. The below list features a cross-segment of portals and satellite data managers.

	API Access to raw imagery	Search by geometry	GUI	Notifications	Monitor sites	Inspect and modify	Sentinel 1	Sentinel 2	Landsat 8	ASTER	Free	Open Source
COAH[8]	✓	✓	✓				✓	✓			✓	✓
G^{4}CAP		✓	✓	✓	✓	?		✓			✓	
SentinelSat	✓	✓					✓	✓			✓	✓
Sat-API	✓	?						✓	✓		✓	✓
USGS Earth Explorer		✓	✓					✓	✓	✓	✓	✓
USGS LandsatLook						✓			✓		✓	
USGS GloVis			✓					✓	✓		✓	✓
Libra			✓						✓		✓	✓
api.nasa.gov	*	✓							✓		✓	
earthdata.nasa.gov			✓								✓	
Daac2Disk		✓	✓							✓	✓	
Landsat Explorer		✓	✓			✓			✓		✓	
Astro Digital	✓	✓	✓	✓	✓	✓		✓	✓		?	
EOS Land Viewer		✓	✓			✓		✓	✓		✓	
EOS Engine	✓	✓		?	?	?	✓	✓	✓			
Sentinel-hub		✓	✓	✓	✓	✓		✓	✓			*
QGIS - SACP[9]		✓	✓			✓		✓	✓	✓	✓	✓

Table 4.1 Portals and Satellite Data Managers
? denote unverifiable features, * denotes partial verification.
Functionality checked 20-21 May 2017.

[8] Abbreviation of: Copernicus Open Access Hub

[9] Abbreviation of: Semi-Automatic Classification Plugin (Congedo, 2016)

4.2. Monitoring System

Looking at Table 4.1 Portals and Satellite Data Managers we see, that no single solution enables programmatic access to Landsat 8, Sentinel and ASTER data and an alert/monitoring module. This chapter describes the process of developing a prototype management system that would enable this while storing metadata instead of raw files to keep the scale of the system small.

The name of the prototype system is Vågen[10]. Vågen is a geospatial mashup combining data from multiple sources in a thin WebGIS client following best practices as described by Pinde Fu and Jiulin Sun in (Fu & Sun, 2011). All the code for the monitoring system is available at https://github.com/casperfibaek/sentinelMonitor.

4.2.1. Conceptual overview

The application is divided into two parts; (1) the front end and the graphical user interface and (2) the backend. A cloud-based Windows 2012 R2 server host a NodeJS – ExpressJS application that controls both parts. A PostgreSQL database holds image metadata and user information. Communication between the front end and backend is through a RESTful API (Fielding, 2000). The NodeJS Request library handles communication with external API's such as the Copernicus SciHub. The certification authority "Let's Encrypt" ensures HTTPS encryption.

[10] Vågen is a Danish pun. (noun: the Hawk, adj: to be awake, verb: to oversee)

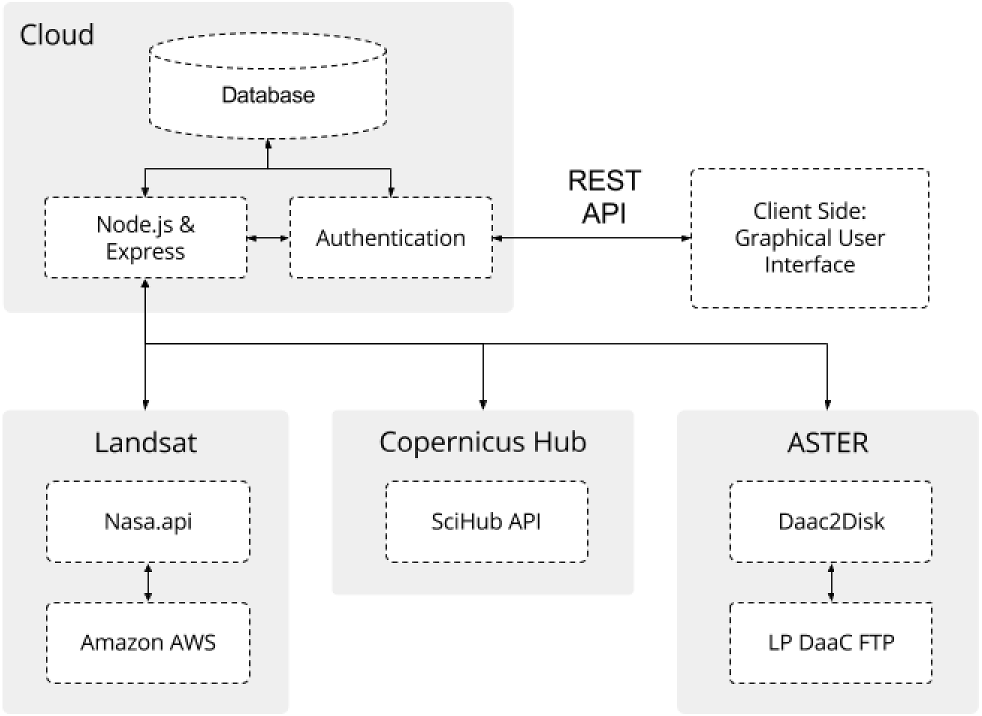


Figure 4.1 Basic overview diagram of Vågen

4.3. The Backend

The backend is controlled by Express routes and revolves around a session system and a PostgreSQL database. NodeJS is a programme that allows the user to use JavaScript on the backend instead of the browser (Haverbeke, 2015). On top of NodeJS, the framework Express is used (Mike Cantelon, Marc Harter, Holowaychuk, 2014). Express is used to create a web server and the various routes for communicating with the user. This subchapter will go through the different aspects of the application.

4.3.1. Database Setup

The database is a PostgreSQL 9.6 database. Combining NodeJS with a NoSQL database, such as MongoDB or CouchDB, is a popular choice as it is possible to write the entire backend in JavaScript (Mike Cantelon, Marc Harter, Holowaychuk, 2014). However, for this application PostgreSQL and the NodeJS Module 'node-postgres' was used instead, this was to ensure that the platform could utilise the PostgreSQL database extension 'PostGIS'. PostGIS allows for spatial queries in a PostgreSQL database.

The database consists of three tables:

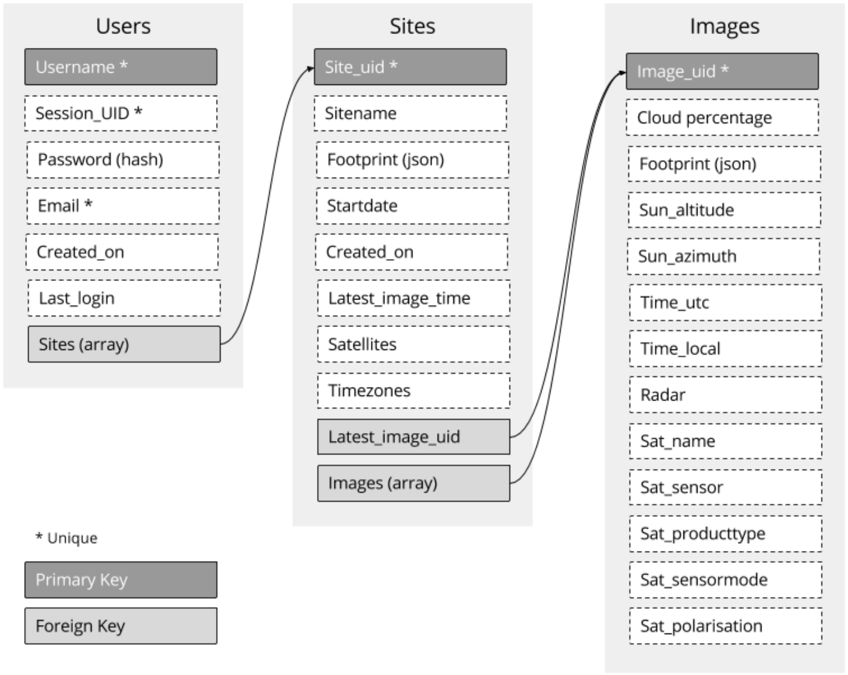


Figure 4.2 Simple Database Relations Diagram

Below is an example of a request to the database with node-Postgres. ExpressJS sets up a route at: “HOST/API/getSites” which inputs an object containing the username and the session key via the HTTP Protocol POST and returns a list of the user's sites.

```
// Open PostgreSQL Client with node-Postgres
var client = new pg.Client(connectionString)
client.connect(function (err) {
  if (err) { errMsg.serverError(err, res) }

  var request = `
    SELECT site_uid, latest_image_time,
    latest_image_uuid, timezone, footprint
    FROM (
      SELECT
        UNNEST(users.sites) AS arr_site_uid,
        sites.latest_image_time AS latest_image_time,
        sites.latest_image_uuid AS latest_image_uuid,
        sites.site_uid AS site_uid,
        sites.footprint AS footprint,
        sites.timezone AS timezone,
        users.username AS username,
        users.session_id AS session_id,
        images.image_uid as image_uid
      FROM sites, users, images
    ) AS b
    WHERE arr_site_uid = site_uid
    AND latest_image_uid = image_uid
    AND username = '${user.username}'
    AND session_id = '${user.session}';`

  client.query(request, function (err, result) {
    if (err) { errorMessage.queryError(err, res) }

    errorMessage.endConnection(client, err, res)

    return res.status(200).json({
      status: 'success',
      message: result.rows
    })
  })
})
```

The request uses template literals to make the SQL Statements more readable as NodeJS supports ECMA Script 2015 (ECMA International, 2015).

The images use an array data type to save the collection of images attached to each site. An array structure ensures that each site can contain multiple images that are iterateable (PostgreSQL Documentation, 2017) in a single field. The images can also now be unique as illustrated in Figure 4.2 Simple Database Relations Diagram.

4.3.2. Handling Users and Security

Because the application should allow the user to create a profile to manage individual monitored sites, it was necessary to implement user control. The database holds the user's passwords in a hashed and salted format.

Hashing a password is using a function that takes a password and returns a scrambled cypher to store in the database. Instead of encrypting and decrypting a password, the original password is never stored in the database. When a user types in their password the function is run, and Vågen compares the scrambled return to the stored value. If they match, an approved session key is added to the user (Provos & Mazieres, 1999).

Salting a password is to protect it against dictionary style attacks called rainbow tables. Salting works by inserting random values into the stored string. The hashing function can also be run multiple times to improve security (Morris & Thompson, 1979).

In Vågen, when a user visits the site a unique session is created. When the user logs in, the password is hashed and compared to the stored string. If successful, the session key validated. This session key now corresponds to a user in the database. The Bcrypt[11] library handles the hashing and salting, and the PostgreSQL database does not store the passwords as plain text.

Password hashing exclusively is not enough to provide proper security, especially when handling non-public data such as control zones for compliance testing. An attacker could still eavesdrop on passwords and other information when it's transferred to the express web server (Callegati, Cerroni, & Ramilli, 2009). Vågen uses HTTPS with a certificate signed by the Let's Encrypt[12] certificate authority to combat eavesdropping.

[11] https://www.npmjs.com/package/bcrypt

[12] https://letsencrypt.org/

We can now get the sites associated with the user as shown below.

```
var request = `
  SELECT site_uid, latest_image
  FROM (
    SELECT
      UNNEST(trig_users.sites) AS arr_site_uid,
      trig_sites.latest_image AS latest_image,
      trig_sites.site_uid AS site_uid,
      trig_users.username AS username,
      trig_users.session_id AS session_id
    FROM trig_sites, trig_users
  ) AS b
  WHERE arr_site_uid = site_uid AND username =
'${user.username}' AND session_id = '${user.session}';`
```

4.3.3. Gathering Metadata

As shown in Figure 4.1 Basic overview diagram of Vågen, Vågen requests information for various API's. The following pages show how vågen collects the data from ESA, NASA and USGS.

ESA – SciHub API

The European Space Agency provides an API with access to Sentinel data. It is possible to specify various parameters in the request such as footprint, cloud cover and sensor type. It returns a path to the images and their metadata.

```
https://scihub.copernicus.eu/dhus/search?
start=0&
rows=3&
platformname:Sentinel-2&
q=footprint:"Intersects(55, 10.5)"
```

The above request will return overall metadata for the three latest Sentinel 2 images over the Danish Island Fionia. A second request is needed to download imagery and to view more detailed metadata. When a user clicks a Sentinel 2 site, Vågen makes this second request and returns either the imagery or the thumbnail.

Below is an overview of how Vågen fetches and stores Sentinel data.

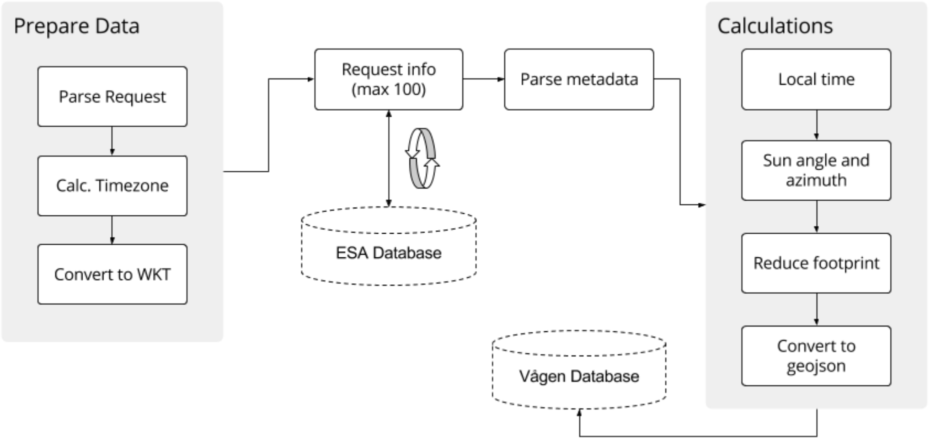


Figure 4.3 Overview diagram of Sentinel data collection

During testing, it became evident that it was quicker to calculate the local time and sun position in the Vågen backend, than making additional requests to the SciHub API for each image. For calculating the position of the sun, Vågen uses the SunCalc[13] library by Vladimir Agafonkin.

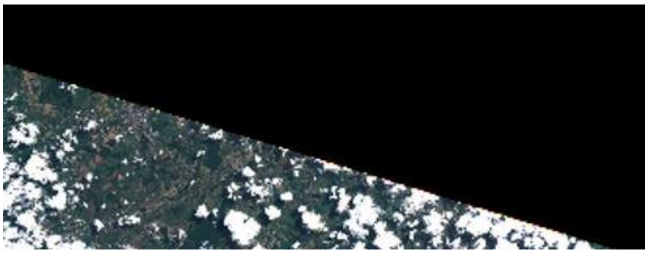


Figure 4.4 Sentinel 2 - Example edge tile

Vågen also reduces the footprints of the images, as they take up a lot of space in the database. The ESA footprints for Sentinel 2 products have redundant vertices. The above tile downloaded from SciHub is defined with 11 vertices at 14 decimals accuracy to the degree. This corresponds to nanometer scale accuracy at the equator[14]. Vågen uses the TurfJS[15] library to reduce the footprints and their impact on the database. The TurfJS functions used to reduce the footprints are shown below. The five signifies the precision while the two means that functions should ignore z-coordinates.

[13] https://github.com/mourner/suncalc

[14] $\frac{40.075km}{360} * 10^{-14} \approx 1nm$

[15] http://turfjs.org/

```
var footprint = turf.truncate(turf.convex( polygon ), 5, 2)
```

The function will reduce vertices count and the accuracy to meter level[16], by first creating a convex hull around the polygon and then reducing its coordinate precision. The core of the script that handles data retrieval from SciHub script can be found in Fetch Sentinel 1 & 2.

NASA

In the data portal review summarised in Table 4.1 Portals and Satellite Data Managers, only the Sat-API library was found to offer programmatic access to the raw Landsat 8 imagery. However, it was not possible to find documentation for tying into said library. Therefore, the solution for Vågen became a two-tier approach. The NASA Planetary API[17] is used to identify the images taken over a project area and the image ID's returned from this service, is used to request the raw imagery and metadata from an Amazon AWS bucket[18].

Below is an example of a request for the image ID's of all Landsat 8 images taken over Fionia in 2017. The return will be a simple array containing the ID's. Connecting to the planetary API requires an API key. Vågen uses a single API key for all users.

```
https://api.nasa.gov/planetary/earth/assets?
lon=10.5&
lat=55&
begin=2017-01-01&
api_key=personalAPIKey
```

[16] $\frac{40.075km}{360} * 10^{-5} \approx 1m$

[17] https://api.nasa.gov/

[18] https://aws.amazon.com/public-datasets/landsat/

This process of acquiring Landsat 8 metadata by Vågen is illustrated in the figure below.

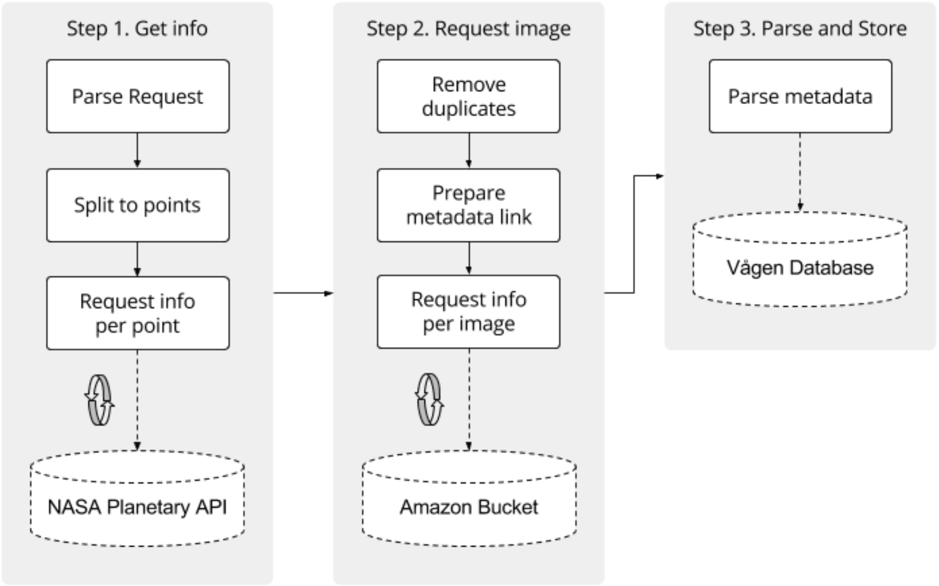


Figure 4.5 Requesting Landsat imagery

As the planetary API does not support polygons, the project area is split into individual vertices, and the images for each point is requested, not storing duplicate images. The core of the script to gather Landsat data is found in appendix V. Fetch Landsat.

ASTER

The United States Geological Survey (USGS) publishes a command line programme called Daac2Disk with has an accompanying web application. Inspecting this application shows an endpoint that accepts a JSON object as well as the hardcoded token ID. This enables setting up a RESTful API using NodeJS. Below is an overview of the process as well as a sample request.

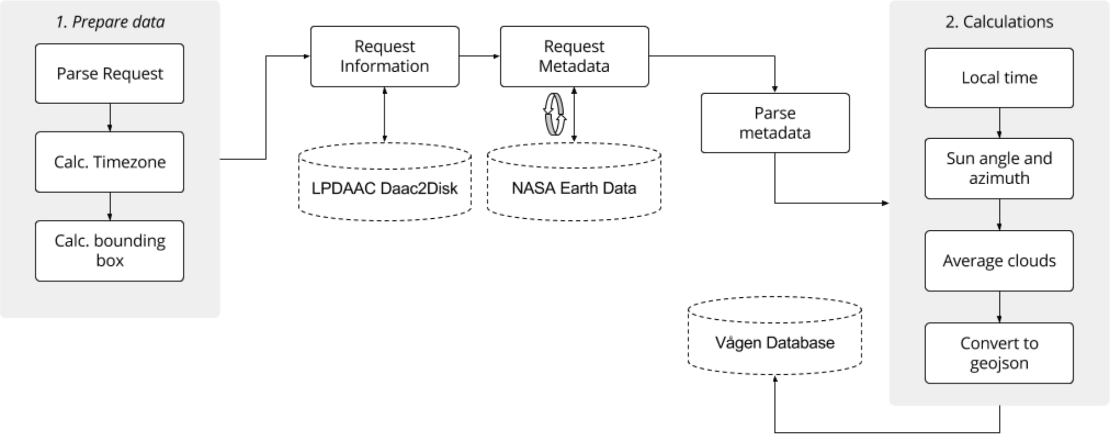


Figure 4.6 Requesting ASTER Imagery

```
var url =
’https://dartool.cr.usgs.gov/cgi-bin/Daac2Disk.cgi’

var postObject= {
    token: '23FFE8BF-CDC1-486B-B5E3-F2B7F55CE151',
    center: 'LPDAAC_ECS',
    shortname: 'AST_L1T',
    version: '003',
    begin: '2017-03-01', end: '2017-03-20',
    mode: 'coordinates',
    urlat: 60, urlon: 16,
    lllat: 51, lllon: 6,
    minhoriz: 0, maxhoriz: 35,
    minvert: 0, maxvert: 17,
    metadata: 'on'
 }
```

The Daac2Disk end-point does not accept a polygon feature, but a bounding box defined by its upper right and lower left corner. Therefore, the bounding box for the monitoring site is calculated using TurfJS before the imagery is requested.

ASTER imagery is a special case compared to Landsat 8 and Sentinel because imagery is ordered (Jet Propulsion Laboratory, 2017), which means that imagery for a site is not always available.

If a controlling agency were to incorporate ASTER data into their monitoring workflow, they would need to order imagery for certain periods beforehand. The script Vågen uses to collect ASTER data can be found in Fetch ASTER.

4.3.4. Communication with Frontend and Exposed API

Communication between the frontend and the backend is done through a RESTful API handled by Express (Mike Cantelon, Marc Harter, Holowaychuk, 2014). The API end-point is located at the monitor.trig.dk/api/. Communication is done through a series of HTTP Protocol GET and POST requests. Below is an example of a POST to '/api/createSite' to create a new site:

```
var createSite = function (request, callback) {
  $.ajax({
    type: 'POST',
    url: '/api/createSite',
    dataType: 'json',
    data: request
  })
    .done(function (response) {
      callback(response)
    })
    .fail(function (xhr, status, error) {
      callback({
        status: status,
        message: error,
        total: xhr
      })
    })
}

createSite(
  {
    projectname: 'Po River Valley',
    geometry: 'GeoJSON (Stringified)',
    options: {
      fromDate: '2017-01-01',
      sentinel1: false,
      sentinel2: true,
      ASTER: false,
      landsat8: true
    },
    user: {
      session: 'Nyr8kM24Q-f00qgWi2YAXH3-PAFtNGz3',
      username: 'testUser'
    }
  },
  /*
  function () { CALLBACK:
      if status is error - post error message
      else redirect to site page and load new sites
  }
  */
)
```

Vågen exposes all available requests to the backend under the global variable app.database.

4.3.5. Alerts and Continuous Monitoring System

The application can be set to look for new images at a fixed interval and email the user via libraries such as nodemailer[19], if the images reach a criterion such as overlap and cloud cover percentage. The user interface part of the alert system is not yet implemented, but the core of it is available in the backend.

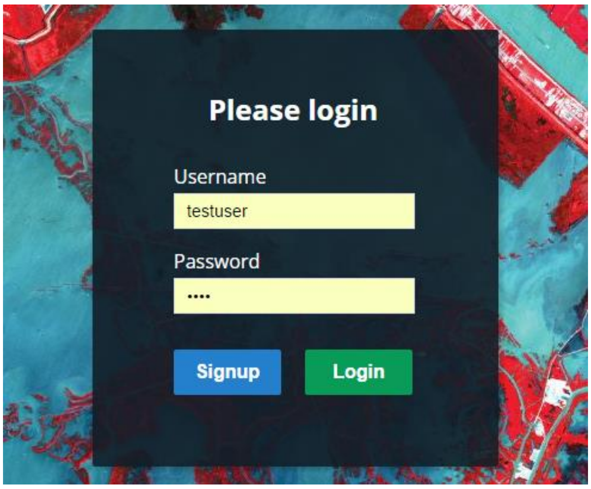
By running the Vågen system locally, it is possible to automatically download new suitable images as they become available by using the request module. It is then possible to trigger GDAL (GDAL, 2012)[20] and python scripts to process the data in accordance to the solutions found in chapter 5. A Case Study Workflow.

[19] https://nodemailer.com/about/

[20] There is also a NodeJS GDAL bindings library called node-gdal

4.4. Frontend

The Vågen frontend is a Single Page Application, meaning that the web page is never fully reloaded (Holmes, 2013). The background and the top and bottom bars remain static throughout navigation. It consists of four primary pages. (1) Login and signup, (2) Site overview, (3) Create site and (4) Inspect site. Each of the site states are controlled by the global variable app.render and callback functions to the backend REST API.

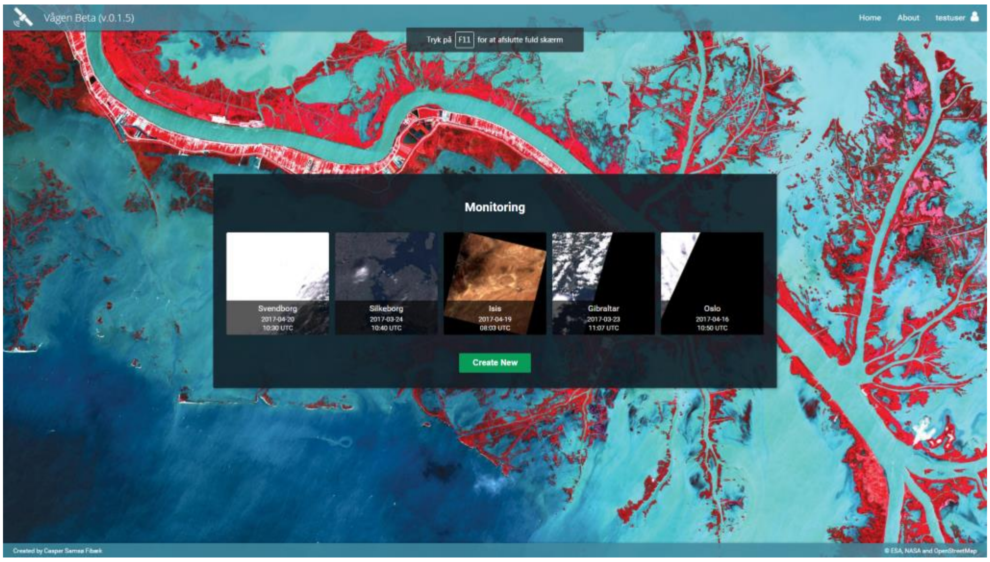


Please login

Username: testuser

Password: ••••

Figure 4.7 The GUI of Vågen (Login page)



Vågen Beta (v.0.1.5)

Tryk på F11 for at afslutte fuld skærm

Monitoring

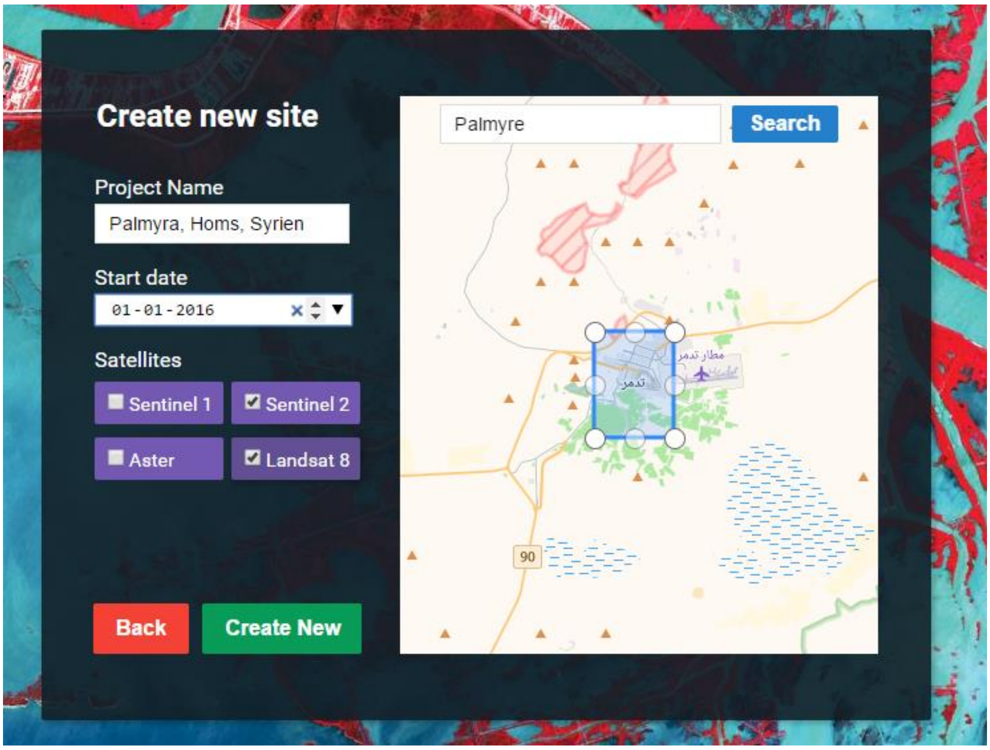
Svendborg	Silkeborg	Isis	Gibraltar	Oslo
2017-04-20 10:30 UTC	2017-03-24 10:40 UTC	2017-04-19 08:03 UTC	2017-03-23 11:07 UTC	2017-04-16 10:50 UTC

Created by Casper Samsø Fibæk

© ESA, NASA and OpenStreetMap

Figure 4.8 The GUI of Vågen (Site Overview)

The front end relies on the Leaflet library by Vladimir Agafonkin[21] for map generation and OpenStreetMap for base map tiles. OpenStreetMap's Nominatim [22] geocoding service enables the search functionality when creating a new site.



Create new site

Project Name: Palmyra, Homs, Syrien

Start date: 01-01-2016

Satellites

□ Sentinel 1 ☑ Sentinel 2

□ Aster ☑ Landsat 8

Palmyre

Figure 4.9 The GUI of Vågen (Create Site Screen)

When a project site is created, it is exported as stringified GeoJSON object (Butler et al., 2008) before being stored in the Vågen database. Then, when the sites are loaded on the inspect site screen (4) – the geometry of the particular site is compared with the footprint of its images and the overlap is calculated with TurfJS on the client side. Doing this allows the user to sort available images by overlap percentages.

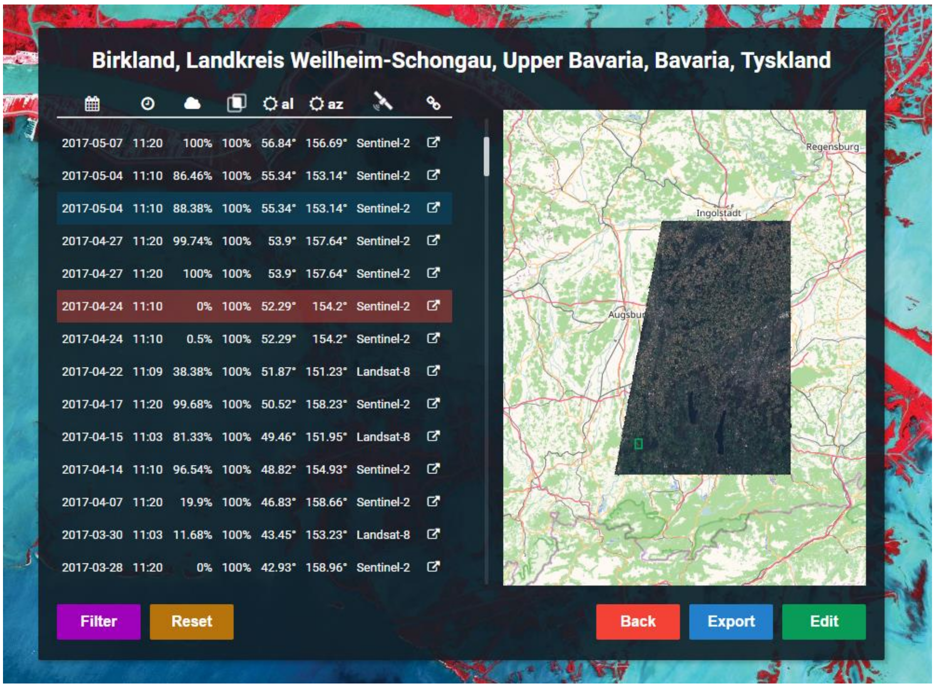
```
var projectGeom = JSON.parse(footprintOfSite)
var projectGeomArea = turf.area(projectGeom)

if (typeof image.footprint === 'string') {
  image.footprint = JSON.parse(image.footprint)
}

image.intersection = turf.intersect(
```

[21] http://leafletjs.com/

[22] http://nominatim.openstreetmap.org/

```
  image.footprint, projectGeom)
image.overlap = (turf.area(
  image.intersection) / projectGeomArea) * 100
```

The black edges around the sentinel images are removed, by converting the image to an HTML5 canvas and removing near black pixels. The resolution of the Sentinel two thumbnails is low compared with Landsat 8 and ASTER, which can make rough screenings of imagery difficult[23]. The Vågen application is limited in the sense that it does not store images itself, only references and metadata. This means that it is not possible to calculate new higher resolution thumbnails for inspection.



Birkland, Landkreis Weilheim-Schongau, Upper Bavaria, Bavaria, Tyskland

				al	az		
2017-05-07	11:20	100%	100%	56.84°	156.69°	Sentinel-2	↗
2017-05-04	11:10	86.46%	100%	55.34°	153.14°	Sentinel-2	↗
2017-05-04	11:10	88.38%	100%	55.34°	153.14°	Sentinel-2	↗
2017-04-27	11:20	99.74%	100%	53.9°	157.64°	Sentinel-2	↗
2017-04-27	11:20	100%	100%	53.9°	157.64°	Sentinel-2	↗
2017-04-24	11:10	0%	100%	52.29°	154.2°	Sentinel-2	↗
2017-04-24	11:10	0.5%	100%	52.29°	154.2°	Sentinel-2	↗
2017-04-22	11:09	38.38%	100%	51.87°	151.23°	Landsat-8	↗
2017-04-17	11:20	99.68%	100%	50.52°	158.23°	Sentinel-2	↗
2017-04-15	11:03	81.33%	100%	49.46°	151.95°	Landsat-8	↗
2017-04-14	11:10	96.54%	100%	48.82°	154.93°	Sentinel-2	↗
2017-04-07	11:20	19.9%	100%	46.83°	158.66°	Sentinel-2	↗
2017-03-30	11:03	11.68%	100%	43.45°	153.23°	Landsat-8	↗
2017-03-28	11:20	0%	100%	42.93°	158.96°	Sentinel-2	↗

Figure 4.10 The GUI of Vågen (Inspect Site Screen)

When a user clicks an image on the inspect site page, a ID unique to the image is send to the backend, and a thumbnail is requested from the appropriate external API. While loading this thumbnail, the footprint of the clicked image is used as a loading icon.

[23] Sentinel 2 Thumbnail GSD: 100km/343px ~ 0.292km GSD, Landsat 8 thumbnail GSD: 185km/965px ~ 0.192km GSD. ASTER Thumbnail GSD: 60km/690px ~ 0.087km

Clicking a sentinel image, the user will be prompted for their username and password for the SciHub. This can be circumvented in two ways – either by proxying through the Express web server with the credentials used by the backend, or going to a different service, such as an Amazon AWS bucket as was done with Landsat. The issue with the first options is that the administrators of the SciHub could close the account due to high traffic from a single account. If the other option were to be implemented a more complex backend would be needed, as sentinel 1 data is not available on Amazon [24] and therefore an additional connection with an external API would need to be maintained.

The incorporation of ASTER data into the interface is not completed, it is only available in the backend.

[24] https://aws.amazon.com/public-datasets/sentinel-2/

5. A Case Study Workflow

This chapter explores ways of applying data gathered through the Vågen application and the information collected surrounding the spatial requirements of the CAP to assist controlling agencies. The chapter shows ways in which mid-resolution satellite imagery can supplement the CAP control. The project area will be tested for the seasons 2015 and 2016. For this study, sentinel 2 data download via the Vågen platform is used.

The data for the study consists of two sentinel 2A satellite images. The images are late in the season and are one year and 16 days apart. Ideally, this would have been closer to one year and earlier in the season where more of the spring crops would still be visible in the fields and the official period for crop diversity testing would be in effect (European Commission, 2016). However, data for 2017 subsidy applications are not yet available, and the earliest cloudless sentinel 2A image of the project area is in August 2015. Sentinel 2A was launched on the 23rd of June 2015 (European Spacy Agency, 2015).

The analysis is carried out in three steps. First the pre-processing of the data used in both subsequent steps. Then a vegetation indices approach and finally a machine learning method using tools from the Orfeo Toolbox (J. Inglada & Christophe, 2009) and the SHARK Library is tested (Igel et al., 2008).

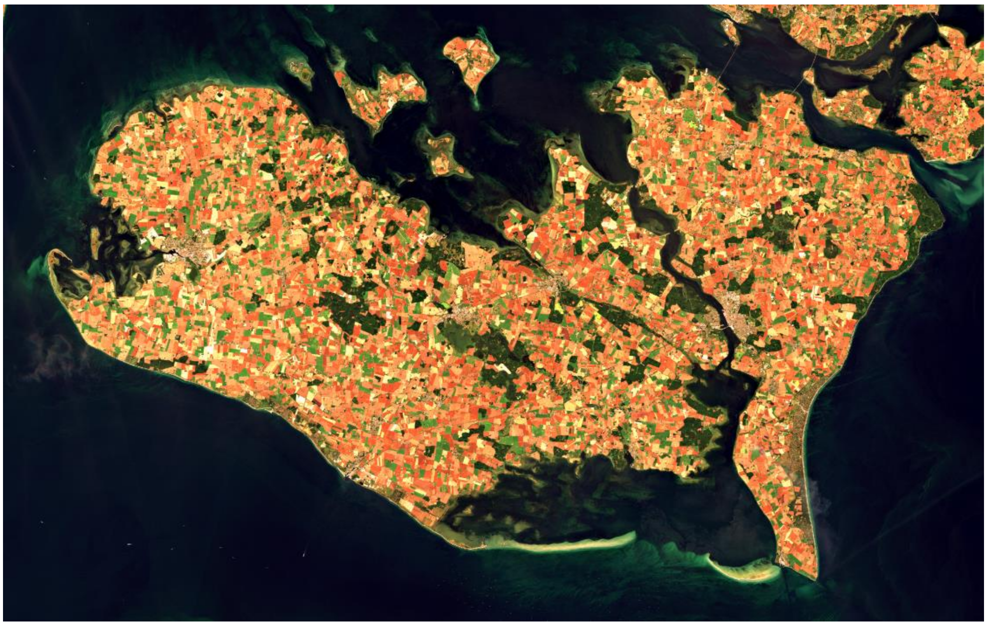


Figure 5.1 Project area on the 9th of August 2015

The chosen satellite images that both have a few cirrus clouds and the 2016 image has a few clouds in the centre of the image. These clouds will be handled during the pre-processing step.

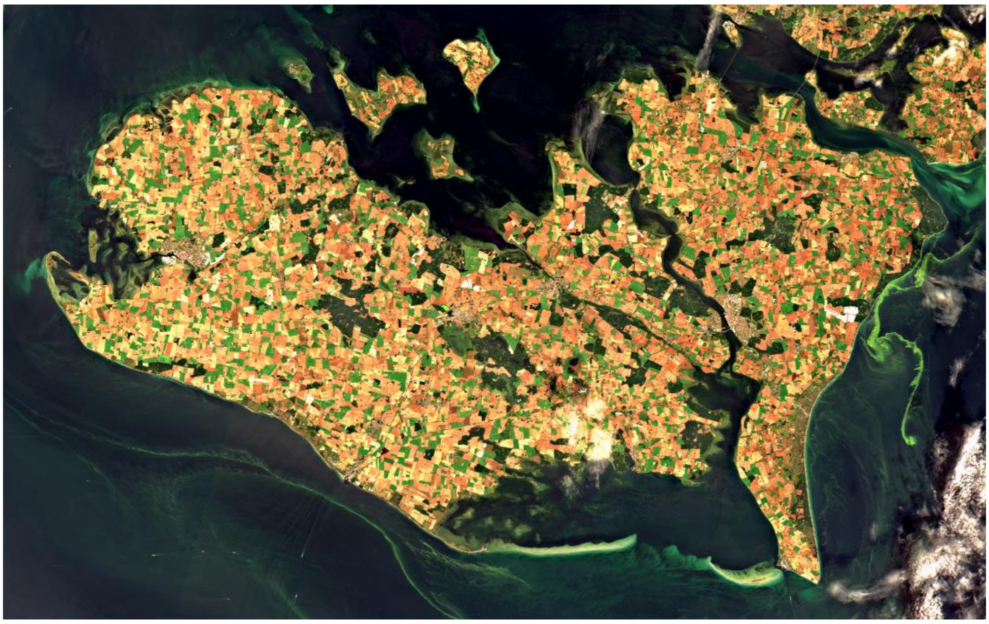


Figure 5.2 Project area on the 24th of July 2016

Besides the satellite imagery, data containing the fields, their area and their declared crop type is available for 2015 and 2016. This data is supplied by the Danish AgriFish Agency and is freely available on their website (Ministry of Environment and Food - Danish Agrifish Agency, 2017a).

5.1. Pre-processing

Before beginning any analysis, the vector files containing the fields for 2015 and 2016 need to be cleaned. The files, as they were available on the AgriFish Agency website on the 3rd of April 2017, all had multiple topological errors, mainly self-intersecting polygons (Bourke, 1993). This type of error breaks many scripts included in GDAL and the Orfeo Toolbox. The ST_MakeValid function from PostGIS was used to fix these errors. The ST_MakeValid function can be used in QGIS by including the LWGEOM plugin. Ensuring valid topology before the data is stored in the database of the controlling agency would eliminate the need for this step.

5.1.1. Creating Masks

For visualisation purposes and speeding up various classification algorithms, masks were created for the agricultural fields of both years. This was done using GDAL and the following function:

```
gdal_rasterize -burn 1 -ot Byte -tr 10 10
  fields2015.shp mask2015.tif
```

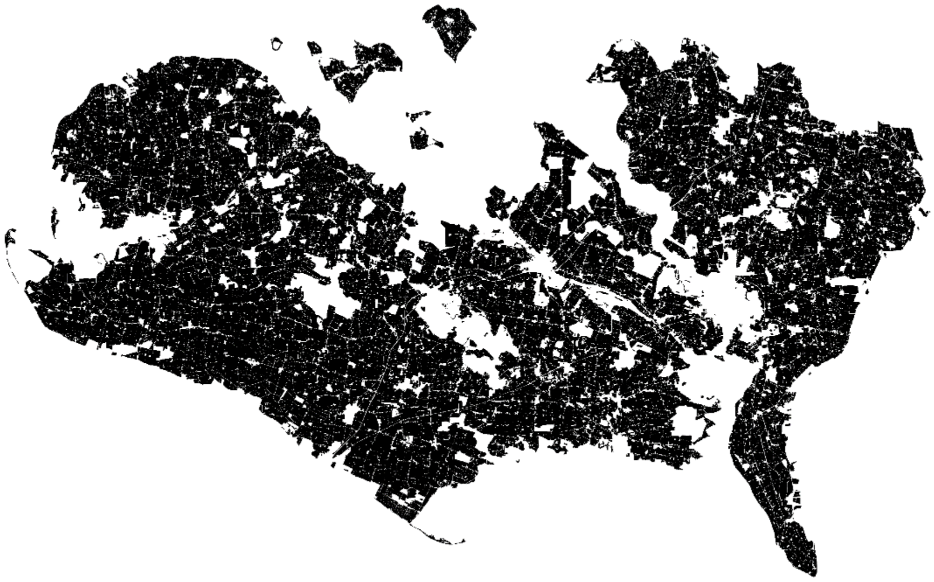


Figure 5.3 Mask of the fields in the project area 2015

The function burns the value one to all pixels intersected by the shape files and outputs a one band mask in 10 by 10 meters resolutions. Exporting as byte data format ensures that nodata values are stored as zeros.

5.1.2. Atmospheric Corrections

Optical Remote Sensing is impacted by a series of atmospheric effects, that can have a large impact if not corrected (Dech, Leutner, & Wegmann, 2016). Especially when comparing indices, atmospheric correction becomes a crucial factor. (Hadjimitsis et al., 2010) found that there was an 18% difference between NDVI values in the same images before and after correction.

The Copernicus SciHub is currently working on providing atmospherically corrected data through the SciHub (Copernicus Open Access Hub, 2017). However, at the time of writing level 2A[25] data is not yet publicly available. Instead, the Sen2Cor application is used to correct the images (Mueller-Wilm, 2016).

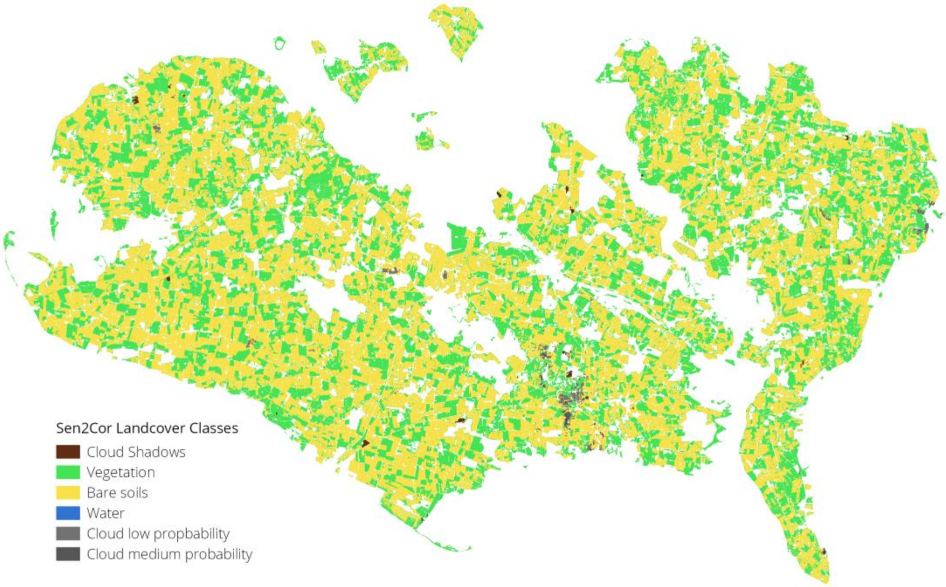


Figure 5.4 Sen2Cor Land Cover Classes (2016)

Using Sen2Cor has the added benefit of the generation of sen2cor landcover classes and cloud masks. These are useful for filtering out agricultural fields, that should not be a part of the analysis, as clouds and cloud shadows contaminate the underlying data.

The Danish national DEM can be used as part of the atmospheric correction steps (Styrelsen for Dataforsyning og Effectivisering, 2017). For this study, a tile from the global DEM SRTM was used to simulate a European-wide case, where national DEM's might not be freely available (Jarvis, Reuter, Nelson, & Guevara, 2008).

Sen2Cor functions can be executed using the command line, the Sentinel 2 Toolbox or through a python script. Here is how it was run through the command line:

```
L2A_Process path/to/granule/manifest
```

[25] Generally speaking, level 2 optical remote sensing data is corrected for geophysical parameters such as atmospheric scattering (Dech et al., 2016).

The configuration parameters for the correction can be set in the L2A_GIPP.xml file. These are the parameters used in this study:

Option	**Value**
Aerosol-type	Rural
Mid-latitude	Summer
Ozone content	Best Guess from metadata
Water vapour correction	True
Water vapour mask	True
Cirrus correction	True
Water vapour cirrus threshold	0.25
BRDF Correction	False (flat terrain)
BRDF Lower Bound	0.22

Table 5.1 Atmospheric correction parameters

The BRDF correction stands for Bidirectional Reflectance Distribution Function and helps reduce the effects of steep terrain on the imagery (Telespazio, 2015a, 2015b). For this study BRDF correction is turned off, as the islands are mostly flat as shown in 3.3.7. Slope detection.

5.2. Indices and K-Means Classification

Calculating vegetation indices has long been used for checking plant growth and health (Bannari et al., 1995). Before comparing fields and finding outliers vegetation indices and heterogeneity, two different vegetation indices are calculated for the images.

5.2.1. Normalised Differential Vegetation Index (NDVI)

NDVI is the most commonly used vegetation index (Xue & Su, 2017) and was first proposed in 1973 in (Rouse, Haas, Schell, & Deering, 1973). In this study, the index is used to estimate where crops are in their growth cycle as well as determining field homogeneity. Negative NDVI values correspond to water and snow. Bare soil has a very low value, and higher values correspond to dense vegetation. NDVI is calculated using GDAL and bands from the atmospherically corrected sentinel 2 data. This was done as shown below:

```
Python gdal_calc.py
-A /path/to/granule_band4.jp2
-B /path/to/granule_band8.jp2
--calc="
 (B.astype(float) - A.astype(float)) /
 (B.astype(float) + A.astype(float))"
--type="Float32"
--outfile="2015_NDVI.tif"
```

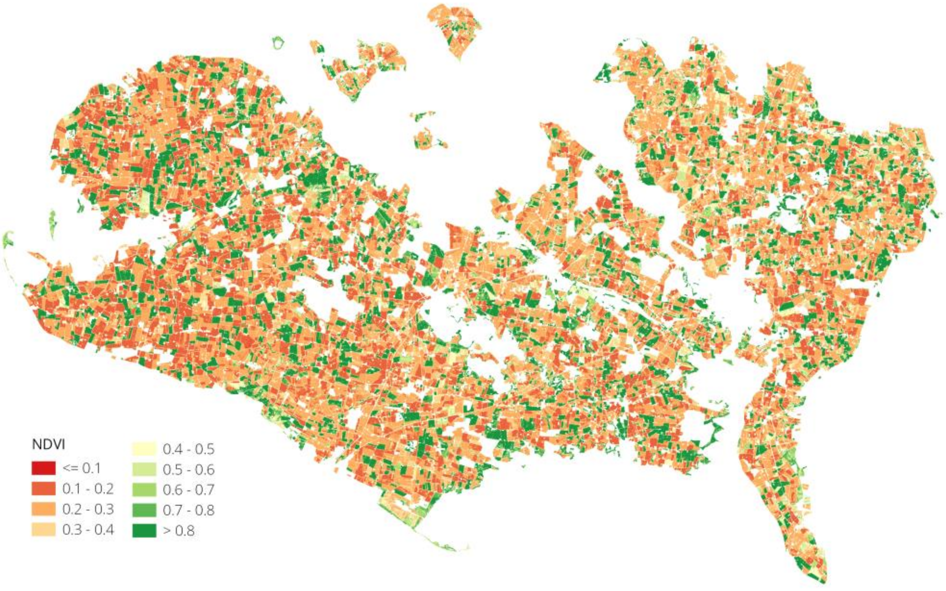


Figure 5.5 NDVI over project area 2015

5.2.2. Red Edge Normalised Vegetation Index (RENDVI)

There exist many different indices each having different purposes. In 1995 (Bannari et al., 1995) described more than fifty indices. Today the remote sensing indices database indexdatabase.de lists over 500 (Henrich, Krauss, Götze, & Sandow, 2017).

For the purpose of this study, a sentinel 2 approximation of the Red Edge NDVI was used (Gitelson & Merzlyak, 1994). This is due to its potential use in burn severity as described in (Fernández-Manso et al., 2016) and due to the red edge bands being key in providing information on the state of vegetation (Pereira, Ramos, Nunes, Azevedo, & Soares, 2016). The RENDVI can be calculated using GDAL as shown below.

```
python gdal_calc.py
-A /path/to/granule_band5.jp2
-B /path/to/granule_band6.jp2
--calc="
 (B.astype(float) - A.astype(float)) /
 (B.astype(float) + A.astype(float))"
--type="Float32"
--outfile="2015_RENDVI.tif"
```

The resulting map is very like the NDVI map when displayed from: -0.2 to 0.6. By using bands different from the NDVI index, it can contribute to finding outlier crops in other spectra.

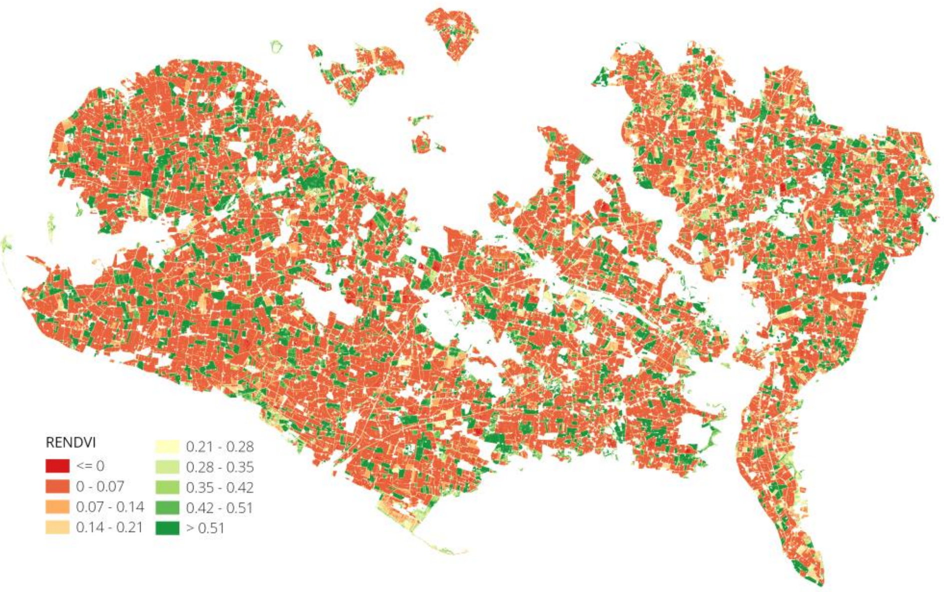


Figure 5.6 RENDVI over project area 2015

5.2.1. Indices outliers

To assist in pointing out areas for rapid field visit or potential risk zones as described in3.2. Control by Remote Sensing, this thesis proposes a comparison of indices across similar crop types to find significant outliers.

Before any comparison is carried out, the average index value and internal variation are calculated using either the Zonal Statistics Plugin for QGIS or the v.rast.stats python script from GRASS[26]. The average index score, or local score, for each field, is saved to its attributes along with the internal variance in indices values. The weighted mean index values per crop type is calculated as follows:

$$w_m = \sum_{i=1}^{n} \left(N_i * \frac{a_i}{\sum_{i=1}^{n} a_i} \right)$$

Where N_i is the average index score of each field per crop type and a_i is the area of the specific agricultural field

The standard deviation per crop type is now:

$$\sigma = \sqrt{\sum_{i=1}^{n} (N_i - w_m)^2}$$

Where N_i is the average index score the particular field

Which is needed for the z-score for each crop:

$$z_i = \frac{N_i - W_m}{\sigma}$$

The z-score, or standard score, is useful for comparing fields, is it shows per field how different in standard deviations a field is from the mean assuming normal distribution (Weisstein, 2017).

[26] https://grass.osgeo.org/

A script was created to calculate these indices scores. It is included in Indices and Variation. Below is the initial part of the script to calculate the weighted mean.

```
// CALCULATE WEIGHTED MEAN
for (var j = 0; j < features.length; j += 1) {
  var cropProp = features[j].properties
  var cropName = cropProp[options.cropName]

  var ndviMean = cropProp[options.cropNDVI]
  var ndviVariance =
    Math.pow(cropProp[options.cropNDVI_std], 2)

  var rendviMean = cropProp[options.cropRENDVI]
  var rendviVariance =
    Math.pow(cropProp[options.cropRENDVI_std], 2)

  if (cropProp[ndviMean] === '' ||
  cropProp[rendviMean] === '') {
    cropProp[ndviMean] = null
    cropProp[rendviMean] = null
  } else {
    var typeArea = uniques[cropName].totalArea
    var weight = cropProp[options.cropArea] / typeArea
    var ndvi_weight = ndviMean * weight
    var rendvi_weight = rendviMean * weight
    var internalWeightNDVI = ndviVariance * weight
    var internalWeightRENDVI = rendviVariance * weight

    uniques[cropName].internal.NDVI.weightedMeanVariance
      += internalWeightNDVI

  uniques[cropName].internal.NDVI.numbers.push(ndviVariance)
  uniques[cropName].NDVI.weightedMean += ndvi_weight
  uniques[cropName].NDVI.numbers.push(ndviMean)

  uniques[cropName].internal.RENDVI.weightedMeanVariance
    += internalWeightRENDVI
  uniques[cropName].internal.RENDVI.numbers.push(
    rendviVariance)
    uniques[cropName].RENDVI.weightedMean += rendvi_weight
    uniques[cropName].RENDVI.numbers.push(rendviMean)
    }
  }
```

Once the script is executed both training data for the machine learning algorithm is outputted along with an overview of the different crops and their averages. Each field in the outputted 'disp'[27] file now has a Z-Score for its average index score compared to the weighted average of all other fields of the same type. The script provides the user with a way to set cut-off points for selecting fields for control. The default values for selecting a field for control is:

1. At least 25 fields of the crop type
2. At least 10 hectares to the field
3. Z-score (either NDVI and RENDVI) above or below two.
4. No internal variance/heterogeneity above or below two Z-score. (Covered in the following chapter)

Using these parameters, it is possible to quickly find fields that significantly differ from other fields of the same type. There could be several reasons for a large z-score. Possible reasons could be:

- Early or late harvest
- Crop disease
- Establishing catch crops
- Declared crop is different than in-field crop.

The above list is not exhaustive. A significant outlier could be interesting for rapid field visits, and the cut-off points could be adjusted to correspond with the number of available field visits. If used to establish control zones, areas with many outliers could be selected. The following images and the accompanying table show a cross section of a few fields and their scores.

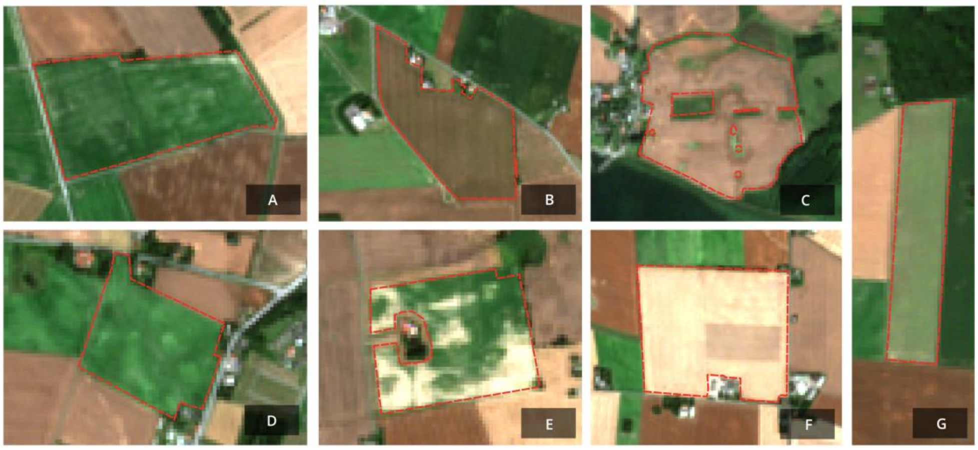


Figure 5.7 Example of fields and their z-scores for indices values

[27] Abbreviation of: dispersion

Image ID	A	B	C	D	E	F	G
Area (ha)	23.23	19.88	21.52	10.68	13.67	32.83	10.05
Crop Code	22	11	11	160	160	160	21
Local NDVI	0.836	0.490	0.291	0.834	0.574	0.165	0.583
Local RENDVI	0.554	0.209	0.098	0.530	0.345	0.078	0.249
Crop type NDVI	0.257	0.224	0.224	0.834	0.834	0.834	0.556
Crop type RENDVI	0.060	0.047	0.047	0.533	0.533	0.533	0.221
Z-score NDVI	8.757	4.258	1.071	-0.004	-2.802	-7.211	0.102
Z-score RENDVI	11.466	3.661	1.162	-0.034	-2.290	-6.192	0.271

Table 5.2 Table of z-scores (Vegetation indices)

In the above table and image the fields A, B, E and F would have been taken out for control with the default parameters of a two-standard deviation threshold. Field A is winter canola which is typically sown in mid-August (SEGES, 2017b) which would indicate that the field is very late for its preceding harvest or non-complaint. Field B has crop code 11, which is winter wheat that is typically sown in mid-September (SEGES, 2015). If this is the crop in-field, it is also very late with its preceding harvest. Fields E and F are both sugar beet fields that are negative outliers in their index values. Inspecting the image of field E, it would appear the field is either struck by disease or non-compliant. Field F is very likely non-compliant as the harvest for sugar beets is two months away, and the beets are not visible in the image.

5.2.2. Internal Variance and Heterogeneity

Besides looking at outliers in vegetation scores, it is also possible to find outliers regarding internal variance in the field. That is, how different are the index values within the field from each other. Some crops might have higher variance than others due to the nature of the crop such as the distance between crops in potatoes versus grass. Therefore, it makes sense to look at variance against the weighted average variance of other fields with the same crop type, much like in the previous sub-chapter. If a field has uncharacteristically high variance, it might be an incorrectly drawn field, crop disease or multiple crops being grown on the same field. During the current workflow implemented by the Danish AgriFish

Agency, determining heterogeneity is a time-consuming process (Ministry of Environment and Food - Danish Agrifish Agency, 2017c) that could be reduced with following steps.

Using an unsupervised classification such as K-means or ISODATA classification on the masked fields, it is possible to determine how fragmented a field is and thereby finding possible incorrectly drawn fields (Dech et al., 2016). Furthermore, such a classification can be incorporated into a rough control of whether a farmer lives up to the greening regulation regarding crop diversity discussed in 2.3.1. Crop Diversification. This chapter looks into both methods, starting with internal variance.

When calculating the zonal statistics for the fields, it is possible to output the internal variance of the index values of the field. By calculating the weighted average variance per crop type and looking at z-scores – we can again find outliers that have uncharacteristically high or low variance. The script created to calculate these variances are found in Indices and Variation and follows similar procedures as the preceding chapters. Below are a few examples of fields and their variance and accompanying z-scores.

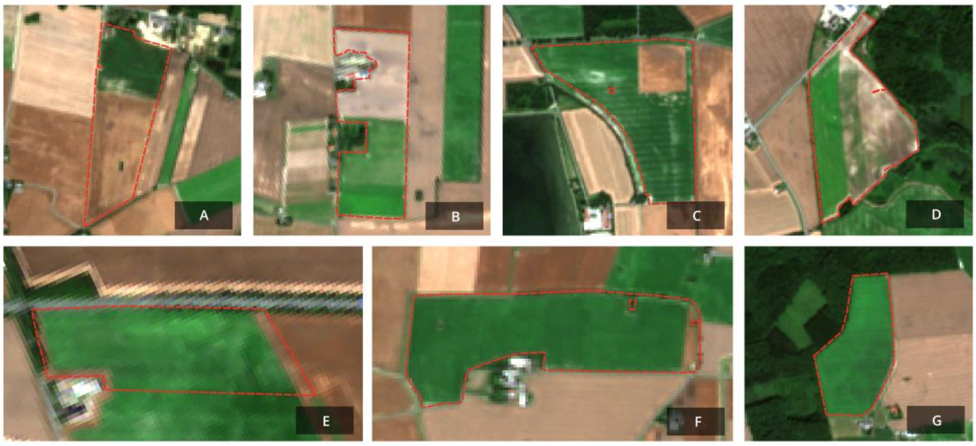


Figure 5.8 Examples of fields and their z-score for variance

Image ID	A	B	C	D	E	F	G
Area (ha)	20.46	18.54	40.55	34.41	11.04	19.37	28.62
Crop Code	1	150	160	150	160	152	160
Local NDVI σ^2	0.076	0.089	0.052	0.062	0.025	0.001	0.012
Local RENDVI σ^2	0.049	0.036	0.038	0.031	0.013	0.001	0.007
Crop type NDVI σ^2	0.010	0.016	0.010	0.016	0.010	0.013	0.010
Crop type RENDVI σ^2	0.006	0.007	0.006	0.007	0.006	0.005	0.006
Z-score NDVI	6.966	4.505	4.394	2.844	1.750	-1.101	0.453
Z-score RENDVI	7.357	4.082	5.152	3.341	1.327	-0.967	0.678

Table 5.3 Table of z-scores (variance)

Using the default parameters of the script, fields A, B, C and D would be taken out for control as they are above the two standard deviation threshold in either NDVI or RENDVI internal variance.

Besides looking at variance to test for heterogeneity, we can also look into using unsupervised classification methods to divide the fields into clusters using our masks created in 5.1.1. Creating Masks. An unsupervised classification is useful both as a rough test of crop diversity, but also to complement the above variance test. The classification method used in this study is the K-means (Bishop, 2006) unsupervised classification method as implemented by the Orfeo Toolbox (J. Inglada & Christophe, 2009).

For the purpose of this study 12 classes are used. To implement this in a workflow for control, would require an analysis of the appropriate number of classes corresponding to the time of image capture.

This thesis proposes either a majority ratio or the following index developed for this study to supplement internal variation testing:

$$\text{NHI: } \frac{\sum_{i=1}^{n}\left(\frac{x_i}{2a - x_i}\right) - 0.5}{0.5}$$

Where a is $\sum_{i=1}^{n} x_i$ and x_i is the area of the specific in-field crop

Equation 1. Normalised Heterogeneity Index (NHI)

The reason for using the sum of the area of the particular crops instead of the area of the field polygon is because we are working with raster data that has been clipped to polygons – the two areas will not correspond precisely.

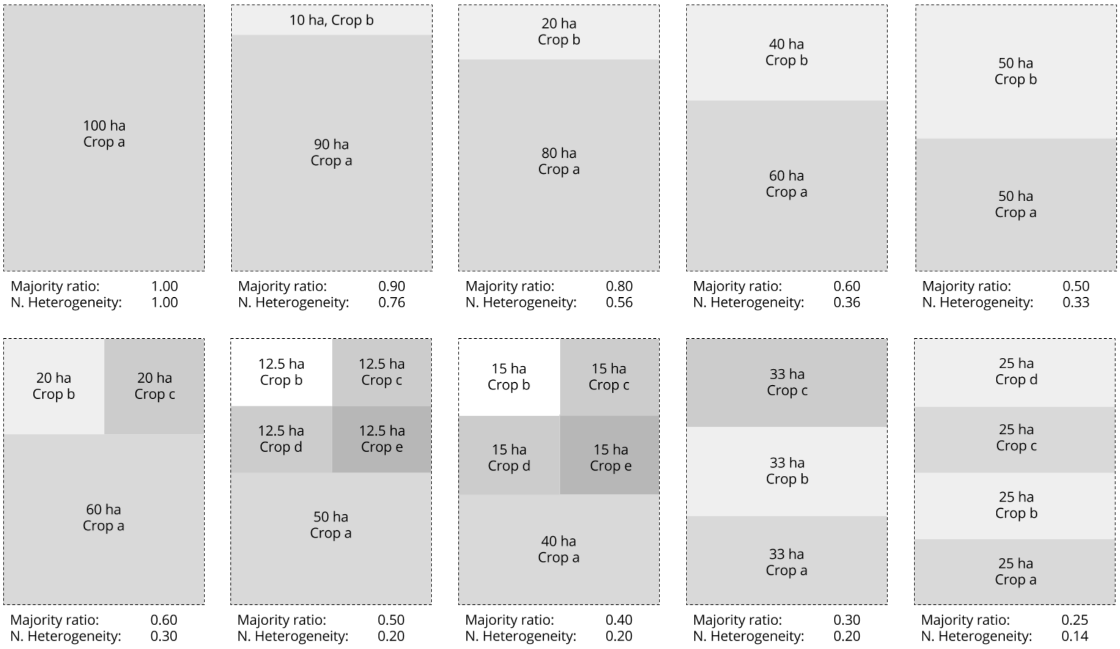


Figure 5.9 Comparison of NHI and Majority Ratio

Fields score lower on the NHI than the majority ratio, which is by design, as its purpose is to consider subsequent divisions. Running the script to calculate NHI also returns the primary crop in the field by using zonal statistics and enables us to test compliance with the greening regulation.

The script created can be found in appendix VIII. Calculating Normalised Homogeneity Index.

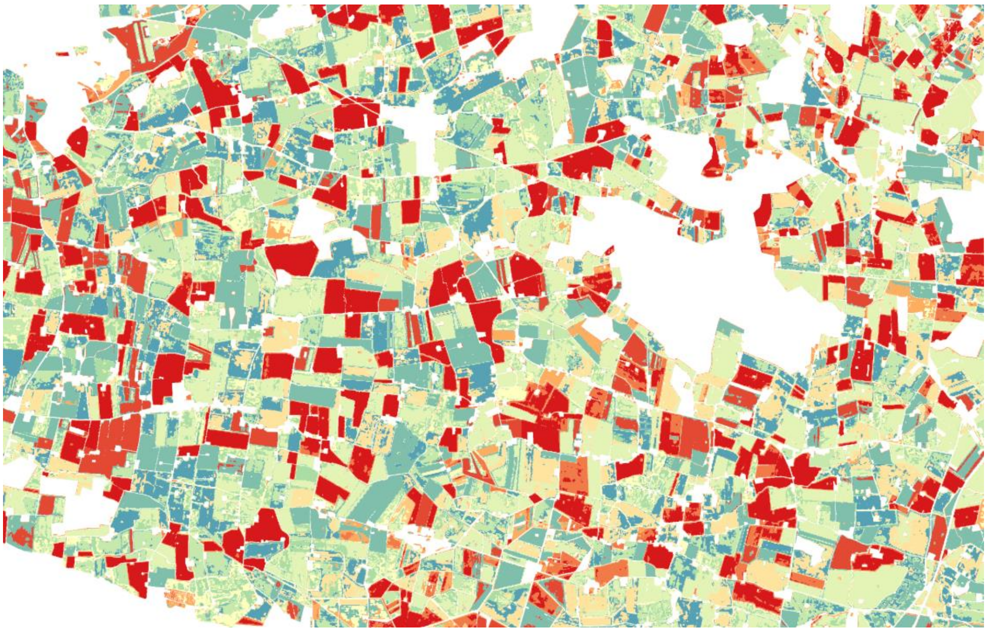


Figure 5.10 1:70.000 scale of K-Means unsupervised classification.

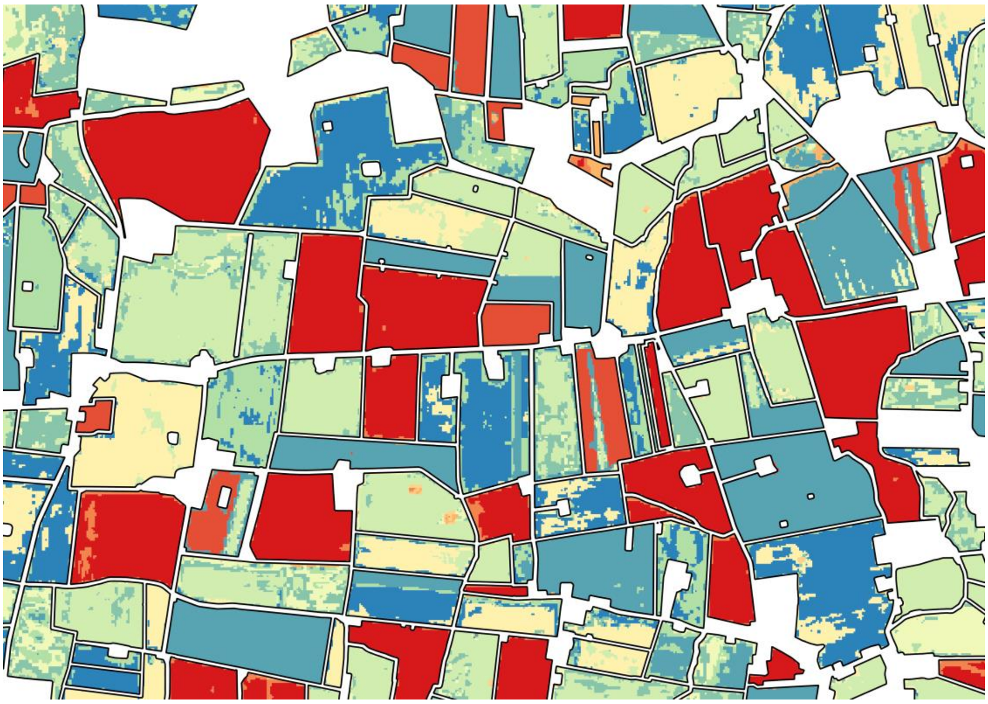


Figure 5.11 1:20.000 scale of K-means unsupervised classification (12 classes)

By combining both checks - index values vs. variance and NHI - it is possible to find fields that are internally homogenous, but outliers regarding vegetation indices scores. That is fields that appear consistent but are outliers regarding average vegetation indices scores. These types of fields could be indicative of non-compliance with harvest and ploughing periods.

5.3. Machine Learning

Using Machine Learning to build a classifier capable of identifying crop types and vegetation states would be beneficial for a range of the regulation outlined in the CAP. Using WorldView-2 VHR imagery to assess compliance with greening regulation in Denmark, has already been studied by researchers at Aarhus University in (Chellasamy et al., 2016). There they report a 90.2% overall accuracy across 18 crop types. In (Immitzer, Vuolo, & Atzberger, 2016) the research team used sentinel 2 imagery and applied Random Forest machine learning algorithms to 7 classes of crops in central Europe and reported an overall accuracy of 76% for crop types.

This chapter seeks to use the Random Forest classification method (Breiman, 2001) distributed through the SHARK Machine Learning Library and developed by Copenhagen University, to classify a single image as well as a multi-temporal image classification of the project area (Igel et al., 2008).

5.3.1. Training

Before beginning the training of the classifier, the project area is divided into three: Lolland 2015, Falster 2015 and both 2016. First Lolland (red) is used as ground truth and training material and tested on Falster (blue), and then both islands in 2015 are used as training material and ground truth for classifying the 2016 image.

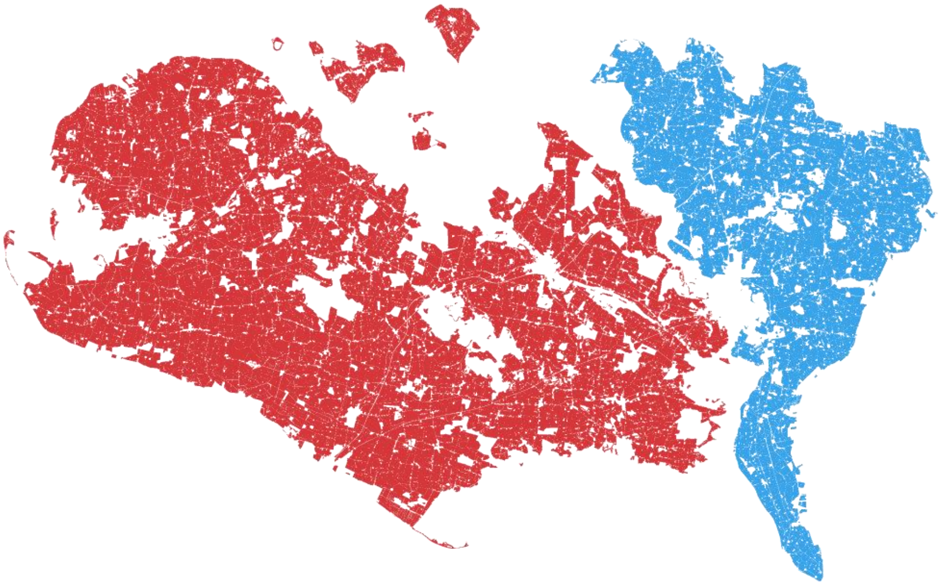


Figure 5.12 Training sets. Red: Training and ground truth. Blue: Validation set

Since the images are captured late in the season, few crops are in the fields. The most abundant crop is sugar beets.

The crops were categorised according to the crop codes published on the Danish AgriFish Agency's website (Ministry of Environment and Food - Danish Agrifish Agency, 2017b). Crops with above 100 fields and 500 hectares in total in both 2015 and 2016 were chosen for the initial sift. Below is their categorisation.

Crop Code	Initial Category	Final Category	2015 samples	2016 samples
583	Pine tree	Pine tree (1)	12876	16338
528	Apple tree	Apple tree (2)	7877	10641
308	Fallow land	Fallow land (3)	17036	30190
276	Grass (1)	Grass (4)	22428	27429
263	Grass (2)	Grass (4)	21537	31342
260	Grass (3)	Grass (4)	28231	32077
254	Grass (4)	Grass (4)	78794	112547
252	Grass (5)	Grass (4)	57564	69223
251	Grass (6)	Grass (4)	11177	14485
216	Maise	Maise (5)	46184	97559
160	Sugar beets	Sugar beets (6)	1384222	1830446
108	Fescue seeds	Fescue seeds (7)	185189	236933
22	Winter canola	Bare soil (8)	307611	432488
13	Winter barley (1)	Bare soil (8)	321074	389384
11	Winter barley (2)	Bare soil (8)	2498929	3398437
10	Winter wheat	Bare soil (8)	72816	129901
1	Spring wheat	Bare soil (8)	2918032	4032791

Tabel 5.4 Crop categories and sample count

Besides the initial field count and area sift, fields that had a z-score above or below two in either internal variance or vegetation indices were excluded from the training sets. The script that processed this can be seen in Indices and Variation. Furthermore, a 10-meter negative buffer corresponding to the sentinel 2 GSD (European Spacy Agency, 2015), was applied to the training fields, to ensure that all pixels were completely within the drawn field.

5.3.2. Single Image

The training sets prepared in the previous sub-chapter was used to train the classifier. A random sampling strategy was used, taking the class with the least number of samples and sample other classes at the corresponding rate. The options set for the random forest classifier was 500 trees (nTree) and the mTry parameter set to the square root of the number of features tested, as recommended in (Belgiu & Dragut, 2016).

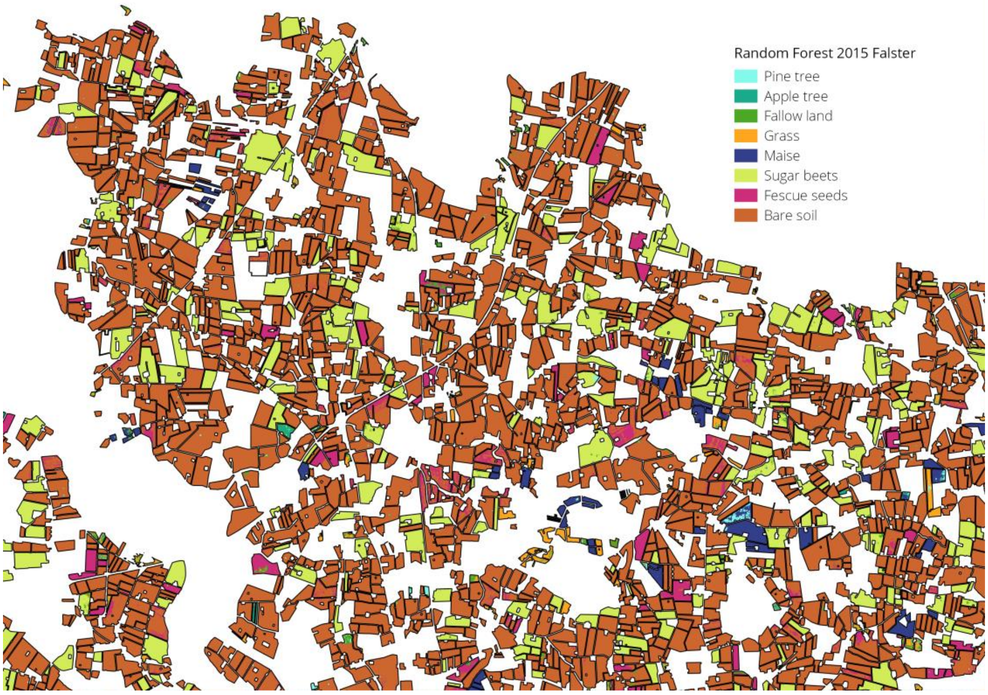


Figure 5.13 Random Forest Classification 2015 Falster (8 classes)

The majority class determined within each field by zonal statistics was used to classify the fields. Calculating the accuracy of the field classification was done with the following script:

```
const fs = require('fs')
const input = 'randomForest2015Falster'
const groundTruth = 'cat2'
const randomForestClass = 'rfClass_ma'

fs.readFile(input + '.geojson', 'latin1', function (err,
data) {
  if (err) { return console.log(err) }
  var geojson = JSON.parse(data)
  var features = geojson.features
  var unique = {}

  for (var i = 0; i < features.length; i += 1) {
    var properties = features[i].properties
    var truth = properties[groundTruth]
    var guess = properties[randomForestClass]
    if (!unique[truth]) {
      unique[truth] = {
        total: 1,
        correct: 0,
        incorrect: {}
      }
    } else {
      unique[truth].total += 1
    }

    if (truth === guess) {
      unique[truth].correct += 1
    } else {
      if (!unique[truth].incorrect[guess]) {
        unique[truth].incorrect[guess] = 1
      } else {
        unique[truth].incorrect[guess] += 1
      }
    }
  }

  var str = JSON.stringify(unique, null, 2)

  fs.writeFile('accuracy.txt', str, function (err) {
    if (err) return console.log(err)
  })
})
```

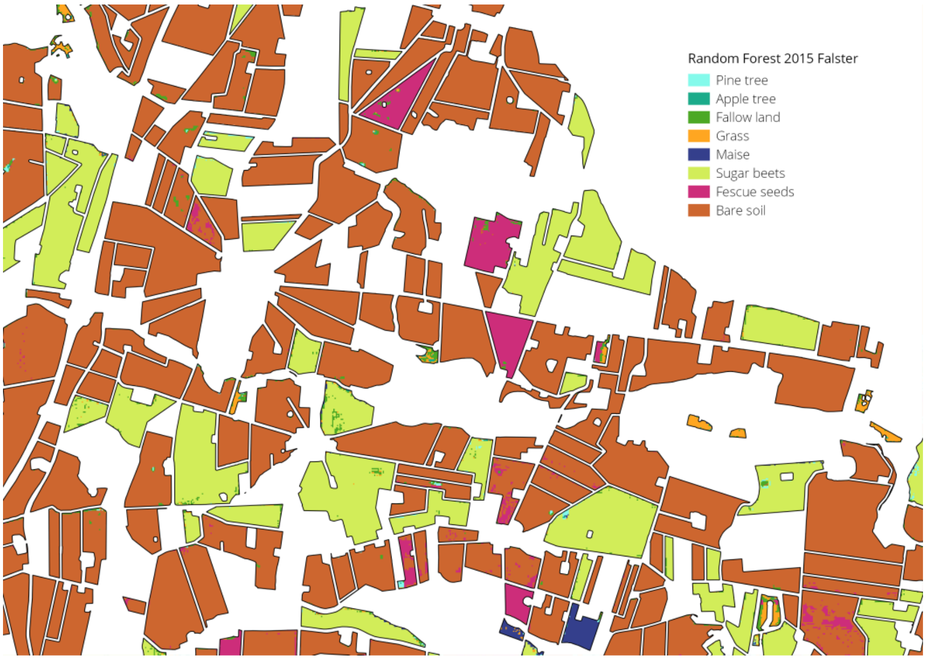


Figure 5.14 Random Forest 2015 - Close-up with buffer

The overall accuracy was 96.6%, which is slightly better than the estimate of 90-95% accuracy expected in crop declaration by farmers (Ministry of Environment and Food - Danish Agrifish Agency, 2017c).

In the below error matrix the producer's accuracy refers to how well an area can be classified, and the user's accuracy refers to the likelihood of the classified pixel being equal to the in-field truth (Congalton, 1991).

	Pine tree	Apple tree	Fallow land	Grass	Sugar beets	Maize	Fescue seeds	Bare soil	User accuracy
Pine tree	**10**	0	1	0	0	0	0	0	90.9%
Apple tree	0	**4**	0	0	0	0	0	0	100%
Fallow land	1	0	**18**	5	0	0	10	1	51.4%
Grass	0	4	19	**106**	1	0	1	0	80.1%
Sugar beets	1	1	0	0	**63**	0	0	0	96.9%
Maize	0	0	0	0	0	**369**	0	0	100%
Fescue seeds	0	0	0	0	0	0	**56**	0	100%
Bare soil	0	0	2	0	0	0	45	**1964**	97.7%
Prod. accuracy	83.3%	44.4%	45.0%	95.5%	98.4%	100%	50.0%	100%	96.6%

Table 5.5 Error matrix of 2015 single image classification (per field)

The reason for the confusion between fescue seeds and bare soil could be due to the image being taken before the fescue seed growth spurt in autumn (SEGES, 2017a) as the fescue seeds have a low average NDVI (Table 5.7 Yearly Average NDVI). Fallow lands getting mistaken for grass types is expectable as the CAP regulation allows for seeding temporary grass for fallow land. For the CAP control the areas of uncharacteristal deviation should be investigated, such as the pine tree and the apple trees mistaken for sugar beets.

5.3.3. Multi-temporal

The process for the multi-temporal classification was similar to the process for the single image. Both of the islands in the 2015 data sets were used for training the classifier with the same options set. Upon completing the classification, it was evident that it is not accurate and not useful for regulation control.

	Pine tree	Apple tree	Fallow land	Grass	Sugar beets	Maize	Fescue seeds	Bare soil	User accuracy
Pine tree	**16**	0	0	23	1	0	0	1	39.1%
Apple tree	0	**0**	1	22	0	0	0	0	0%
Fallow land	26	0	**4**	40	2	0	2	3	5.2%
Grass	52	2	72	**411**	0	0	5	18	73.4%
Sugar beets	28	0	6	42	**2**	0	0	51	1.6%
Maize	15	2	36	27	3	**0**	0	1294	0%
Fescue seeds	59	1	6	13	0	0	**22**	69	12.9%
Bare soil	91	5	232	1722	2	0	61	**4803**	69.4%
Prod. accuracy	5.6%	0%	1.1%	23.9%	20%	0%	24.4%	77%	56.6%

Table 5.6 Error matrix of 2016 multi-temporal classification (per field)

Looking closer at the result of the multi-temporal classification shows that the model appears to overly emphasise grass and bare soil classes in the 2016 image.

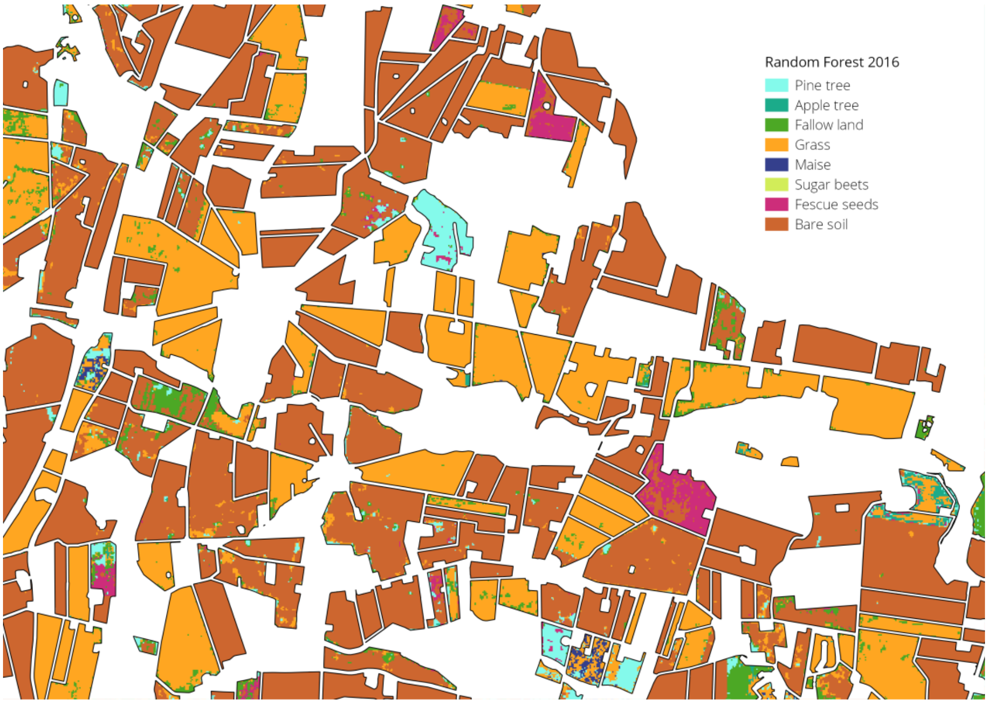


Figure 5.15 Random forest 2016 classification close-up

There could be multiple reasons for the poor results of this classification. The crops in the 2015 image had a lower average NDVI than the 2016 image, despite being later in the season. It could be that 16 days is too far apart in the growth cycle for classifier training purposes and the problem could be meteorological in nature due to rainy periods or the project area experiencing a particular dry period in 2015.

Crop type	NDVI 09/08-2015	NDVI 24/07-2016
Winter barley	0.2235	0.2507
Suger beets	0.8339	0.8684
Maise	0.8839	0.9020
Fescue seeds	0.3658	0.4965
Spring wheat	0.2913	0.3166

Table 5.7 Yearly Average NDVI

6. Conclusion

This thesis set out to determine the spatial requirements of the Common Agricultural Policy that could be suitable for control with medium resolution satellite imagery. The second research question asked for methods for handling the large amount of data generated by the Copernicus, Landsat and ASTER programmes. Finally, the last research question sought a workflow example for assisting in the control of the relevant regulation.

An overview of the spatial CAP requirements was provided and their susceptibility to control with mid-resolution satellite imagery investigated. The review the requirements and the control processes led to the creation of methods to ease some of the aspects of the control – especially regarding field heterogeneity and greening.

An application and management system was created to handle the substantial amount of data generated. The Vågen application is described, and the thesis shows ways in which its possible to connect the various open API's thereby establishing a geospatial mashup of satellite imagery. The methodology behind Vågen ensures that no imagery is required server-side and only stores metadata and references. This guarantees a small-scale setup that can be deployed locally to enable monitoring with few resources.

The thesis proposes a method of calculating statistical outliers regarding weighted average vegetation index values and internal variance for two vegetation indices. The results are combined with an unsupervised classification and the creation of a heterogeneity index. These methods provides a tool for the inspection fields that are significantly different than other fields of their crop type. Each method has accompanying scripts with default values that can be adjusted to comply with the local knowledge of the controlling agencies. By comparing heterogeneity scores and vegetation indicies outliers, it is possible to find incorrectly declared fields.

Despite the unsuitability of the multi-temporal classification, the high overall accuracy of the single image classification proves that it could be a useful tool for testing compliance with the CAP regulation. The classification is especaially useful for testing heterogeneity, using the heterogeneity or majority ratio index, declared crops versus grown crop and greening regulation.

7. Discussion

By combining a rigorous selection of training samples from the declared crop data set and combining it with the decision tree nature of the random forest methodology, it was possible to achieve a high degree of accuracy in crop type detection. However, the control could be improved further by maintaining a year-round ground truth dataset for crop types in multiple locations throughout individual member states. Such a system would not need to include many fields per type, as at 10-meter GSD sentinel 2 provide many samples per field. Having this live ground truth dataset would significantly improve the crop type classification.

If, instead of relying on standard deviation, a database could be established containing expected tolerance levels for different crop types regarding NDVI, RENDVI and heterogeneity at different times throughout the year. These absolute values could improve the selection for control areas.

A negative buffer was not applied for the indices approach described in 5.2. Indices and K-Means Classification. In hindsight, this could be beneficial – especially regarding testing heterogeneity of small fields. A solution to this problem could be introducing a buffer function from TurfJS as part of the indices creation script.

I propose the following change to the current control process: Instead of selecting a bulk of the fields at the beginning of the season – reserve fields for continuous control throughout the year. Doing so would allow nation-wide control, and by looking at outliers in mid-resolution imagery, it is possible to select outlier fields for these in-field controls. Whenever a field is found to be compliant, atmospherically corrected imagery and the field geometry should be archived in a public database to serve as ground truth for other analysis.

The European Union is currently running a programme called RECAP under Horizon 2020. The goal is to create an intergrated platform for control with remote sensing and mobile devices. There is not yet a public application available for testing, but I hope this thesis can contribute to some of the consideration of the controlling agencies and the consortium behind RECAP.

I am developing Vågen as an open source platform. Contributions and forks are highly encouraged

8. Bibliography

Augére-Granier, M.-L. (2015). *EU Rural Development Policy*. Retrieved from http://www.europarl.europa.eu/RegData/etudes/BRIE/2015/568340/EPRS_BRI(2015)568340_EN.pdf

Bannari, A., Morin, D., Bonn, F., & Huete, A. R. (1995). A review of vegetation indices. *Remote Sensing Reviews*, *13*(1–2), 95–120. https://doi.org/10.1080/02757259509532298

Belgiu, M., & Dragut, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS Journal of Photogrammetry and Remote Sensing*. https://doi.org/10.1016/j.isprsjprs.2016.01.011

Bio Intelligence Service. (2010). *Environmental Impacts of Different Crop Rotations in the European Union*. Retrieved from http://ec.europa.eu/environment/agriculture/studies.htm

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning. Pattern Recognition* (Vol. 4). https://doi.org/10.1117/1.2819119

Bourke, P. (1993). Polygon Types. Retrieved from http://paulbourke.net/geometry/polygonmesh/

Breiman, L. (2001). Random forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Butler, H., Cadcorp, M. D., Mit, A. D., Unc-chapel, S. G., Opengeo, T. S., & Metacarta, C. S. (2008). The GeoJSON Format Specification Contents. Retrieved from http://geojson.org/geojson-spec.html

Callegati, F., Cerroni, W., & Ramilli, M. (2009). Man-in-the-middle attack to the HTTPS protocol. *IEEE Security and Privacy*. https://doi.org/10.1109/MSP.2009.12

Chellasamy, M., Ferré, T. P. A., & Greve, M. H. (2016). Evaluating an ensemble classification approach for crop diversity verification in Danish greening subsidy control. *International Journal of Applied Earth Observation and Geoinformation*, *49*, 10–23. https://doi.org/10.1016/j.jag.2016.01.008

Claverie, M., Masek, J. G., & Ju, J. (2016). *Harmonized Landsat-8 Sentinel (HLS) Product User's Guide*. Retrieved from https://hls.gsfc.nasa.gov/documents/

Congalton, R. G. (1991). A review of assessing the accuracy of classifications of remotely sensed data. *Remote Sensing of Environment*, *37*(1), 35–46. https://doi.org/10.1016/0034-4257(91)90048-B

Congedo, L. (2016). *Semi-Automatic Classification Plugin Semi-Automatic Classification Plugin Documentation. SCP.* https://doi.org/10.13140/RG.2.2.29474.02242/1

Copernicus Observer. (2017). *Copernicus - Europe's Eyes on Earth*.

Copernicus Open Access Hub. (2017). Pilot production of Sentinel-2 L2A products over Europe available on the Open Access Hub. Retrieved May 5, 2017, from https://scihub.copernicus.eu/news/News00179

COWI. (2015). *Vækst og nabotjek af krydsoverensstemmelse (KO) på miljø og GLM-området*. Retrieved from http://lfst.dk/landbrug/krydsoverensstemmelse/nabotjek-af-miljoe-og-glm-omraadet/#c50027

Dech, S., Leutner, B., & Wegmann, M. (2016). *Remote Sensing and GIS for Ecologist - A textboox Using Open Source Software*. Pelagic Publishing.

Directorate General - Agriculture and Rural Development. (2017). *CAP expenditure: European Commission, DG Agriculture and Rural Development (Financial Report)*. Retrieved from http://ec.europa.eu/agriculture/sites/agriculture/files/cap-post-2013/graphs/graph1_en.pdf

ECMA International. (2015). *ECMA-262 ECMAScript 6th Edition Language Specification. JavaScript Specification*. Retrieved from http://www.ecma-international.org/ecma-262/6.0/

ESA EarthObservation. (2017). Tweet: "Average daily production of #Sentinel core products: 12 TB; 4 times the original planning https://spacedata.copernicus.eu/." Retrieved from https://twitter.com/ESA_EO/status/872068460603486210

EU Science Hub. (2017). EU Science Hub. In *Monitoring Agricultural ResourceS (MARS)*. Retrieved from https://marswiki.jrc.ec.europa.eu/wikicap/index.php/Use_of_RS_for_control_of_Cross-Compliance

European Commission. (2015). *CAP Explained - Direcy Payments for Farmers 2015-2020*. Retrieved from http://ec.europa.eu/agriculture/sites/agriculture/files/direct-support/direct-payments/docs/direct-payments-schemes_en.pdf

European Commission. (2016). *Review of greening after one year* (1/6). Retrieved from https://ec.europa.eu/agriculture/sites/agriculture/files/direct-support/pdf/2016-staff-working-document-greening_en.pdf

European Commission. (2017a). Greening. Retrieved May 3, 2017, from https://ec.europa.eu/agriculture/direct-support/greening_en

European Commission. (2017b). Integrating environmental concerns into the CAP. Retrieved May 8, 2017, from https://ec.europa.eu/agriculture/envir/cap

European Spacy Agency. (2013). *Sentinel-1 User Handbook*. Retrieved from https://earth.esa.int/documents/247904/685163/Sentinel-1_User_Handbook

European Spacy Agency. (2015). *Sentinel-2 User Handbook* (1st ed.). Retrieved from https://sentinels.copernicus.eu/documents/247904/685211/Sentinel-2_User_Handbook

Fernández-Manso, A., Fernández-Manso, O., & Quintano, C. (2016). SENTINEL-2A red-edge spectral indices suitability for discriminating burn severity. *International Journal of Applied Earth Observation and Geoinformation*, *50*, 170–175. https://doi.org/10.1016/j.jag.2016.03.005

Fielding, R. T. (2000). Architectural Styles and the Design of Network-based Software Architectures. *Building*, *54*, Chapter 5. https://doi.org/10.1.1.91.2433

Fu, P., & Sun, J. (2011). *Web GIS - Principles and Applications* (1st ed.). Esri Press.

Gdal. (2012). GDAL - Geospatial Data Abstraction Library. *Official Web Page*. Retrieved from http://www.gdal.org/

Gil, P. (2017). What Is "SaaS" (Software as a Service')? Retrieved May 1, 2017, from https://www.lifewire.com/what-is-saas-software-2483600

Gitelson, A., & Merzlyak, M. N. (1994). Quantitative estimation of chlorophyll-a using reflectance spectra: Experiments with autumn chestnut and maple leaves. *Journal of Photochemistry and Photobiology, B: Biology*, *22*(3), 247–252. https://doi.org/10.1016/1011-1344(93)06963-4

Hadjimitsis, D. G., Papadavid, G., Agapiou, a., Themistocleous, K., Hadjimitsis, M. G., Retalis, a., ... Clayton, C. R. I. (2010). Atmospheric correction for satellite remotely sensed data intended for agricultural applications: impact on vegetation indices. *Natural Hazards and Earth System Science*, *10*(1984), 89–95. https://doi.org/10.5194/nhess-10-89-2010

Hansen, M. (2017). Integrating Landsat 7, 8 and Sentinel 2 data in improving crop type identification and area estimation. Retrieved from http://www.glad.umd.edu/projects/integrating-landsat-7-8-and-sentinel-2-data-improving-crop-type-identification-and-area

Haverbeke, M. (2015). *Eloquent JavaScript. Javascript Programming Language* (2nd ed.).

Hegazy, I. R., & Kaloop, M. R. (2015). Monitoring urban growth and land use change detection with GIS and remote sensing techniques in Daqahlia governorate Egypt. *International Journal of Sustainable Built Environment*, 4(1), 117–124. https://doi.org/10.1016/j.ijsbe.2015.02.005

Henrich, V., Krauss, G., Götze, C., & Sandow, C. (2017). A database for remote sensing indices. Retrieved from http://www.indexdatabase.de/db/i.php

Holmes, S. (2013). *Getting MEAN with Mongo, Express, Angular, and Node*.

Online. https://doi.org/10.1109/MCSE.2011.99

Igel, C., Heidrich-Meisner, V., & Glasmachers, T. (2008). Shark. *Journal of Machine Learning Research*, *9*, 993–996.

Immitzer, M., Vuolo, F., & Atzberger, C. (2016). First Experience with Sentinel-2 Data for Crop and Tree Species Classifications in Central Europe. *Remote Sensing*, *8*(166), 2–27. https://doi.org/10.3390/rs8030166

Inglada, J., & Christophe, E. (2009). The Orfeo Toolbox remote sensing image processing software. *IEEE International Geoscience and Remote Sensing Symposium*, (November), 733–736. https://doi.org/10.1109/IGARSS.2009.5417481

Inglada, J., Vincent, A., Arias, M., & Marais-Sicre, C. (2016). Improved early crop type identification by joint use of high temporal resolution sar and optical image time series. *Remote Sensing*, *8*(5). https://doi.org/10.3390/rs8050362

Jarvis, A., Reuter, H. I., Nelson, A., & Guevara, E. (2008). SRTM 90m Digital Elevation Database v4.1. Retrieved from http://www.cgiar-csi.org/data/srtm-90m-digital-elevation-database-v4-1

Jet Propulsion Laboratory. (2017). Requesting New Acquisitions. Retrieved from https://asterweb.jpl.nasa.gov/NewReq.asp

Liu, P. (2015). A survey of remote-sensing big data. *Frontiers in Environmental Science*, *3*. https://doi.org/10.3389/fenvs.2015.00045

Lockhart, J. A. R., & Wiseman, A. J. L. (1983). *Introduction to Crop Husbandry*. *Introduction to Crop Husbandry*. https://doi.org/10.1016/B978-0-08-029793-4.50011-X

Lovec, M. (2016). The European Union's Common Agricultural Policy Reforms. *Central and Eastern European Perspectives on International Relations*. https://doi.org/10.1057/978-1-137-57278-3

Ludlow, N. P. (2005). The Making of the CAP: Towards a Historical Analysis of the EU's First Major Policy. *Contemporary European History*, *14*(3), 347–371. https://doi.org/10.1017/S0960777305002493

Mariam-Webster. (2017). Definition of midlatitudes. Retrieved from https://www.merriam-webster.com/dictionary/midlatitude

Matteo, G. Di. (2016). Web Application Updates and the Sentinel-2 Alert Module (p. Control and management of agricultural land in IAC). European Commission, Joint Research Centre. Retrieved from https://ec.europa.eu/jrc/sites/jrcsh/files/3_S1_DiMatteo_G4CAPwa.pdf

Matteo, G. Di. (2017). Image Acquisition. Retrieved May 28, 2017, from https://g4cap.jrc.ec.europa.eu/g4cap/Default.aspx?tabid=327

Meyer, C., Matzdorf, B., Müller, K., & Schleyer, C. (2014). Cross Compliance as payment for public goods? Understanding EU and US agricultural policies. *Ecological Economics*, *107*, 185–194. https://doi.org/10.1016/j.ecolecon.2014.08.010

Mike Cantelon, Marc Harter, Holowaychuk, N. R. (2014). *Node.js in Action* (1st ed.).

Miljø- og Fødevareministeriet. (2017a). *Vejledning Om Direkte Arealstøtte 2017: Grundbetaling, Grønne Krav, ø-Støtte Og Støtte Unge Nyetablerede Landbrugere*. Retrieved from http://lfst.dk/fileadmin/user_upload/naturerhverv/filer/tilskud/arealtilskud/direkte_stoette_-__grundbetaling_mm/2017/vejledning___om_direkte_arealstoette_2017.pdf.

Miljø- og Fødevareministeriet. (2017b). *Vejledning om krydsoverensstemmelse 2017*. Retrieved from http://lfst.dk/landbrug/krydsoverensstemmelse/

Ministry of Environment and Food. (2016). *Vejledning om Gødsknings- og Harmoniregler.*

Ministry of Environment and Food - Danish Agrifish Agency. (2017a). Hvordan får du adgang til data? Retrieved May 29, 2017, from https://kortdata.fvm.dk/download/

Ministry of Environment and Food - Danish Agrifish Agency. (2017b). Koder til Fællesskemaet. Retrieved May 5, 2017, from http://lfst.dk/tilskud-selvbetjening/kom-i-gang-med-selvbetjening/stoetteordninger-i-faellesskemaet/koder-til-faellesskemaet/

Ministry of Environment and Food - Danish Agrifish Agency. (2017c). Satellite Image Analyses for Agricultural Control 2017. Danish AgriFish Agency.

Morris, R., & Thompson, K. (1979). Password security: a case history. *Communications of the ACM*, *22*(11), 594–597. https://doi.org/10.1145/359168.359172

Mueller-Wilm, U. (2016). Sen2Cor Configuration and User Manual. Retrieved from http://step.esa.int/main/third-party-plugins-2/sen2cor/

Muro, J., Canty, M., Conradsen, K., Hüttich, C., Nielsen, A., Skriver, H., … Menz, G. (2016). Short-Term Change Detection in Wetlands Using Sentinel-1 Time Series. *Remote Sensing*, *8*(10), 795. https://doi.org/10.3390/rs8100795

NASA. (2017). Active Fire Data. Retrieved May 14, 2017, from https://earthdata.nasa.gov/earth-observation-data/near-real-time/firms/active-fire-data

Nyborg, L. (2017, April 7). Successful demonstration of using Sentinel 1 & 2 in support of Common Agricultural Policy in Denmark. *Http://www.dhi-Gras.com/news/*, p. 1. Retrieved from http://www.dhi-gras.com/news/2017/4/7/successful-demonstration-of-using-sentinel-1-and-2

Nyborg, L., & Eskesen, S. (2017). Sentinel 1 og 2 lige på kornet.. In *Smart Agriculture*. Retrieved from http://workshop.copernicus.eu/sites/default/files/content/attachments/form-qANuASXJjBug9rR3aBjkFKDoLC63Fhy4NtXysfBQz18/6_sentinel-1-2_agricultural_monitoring_sannelotte.pdf

Pereira, M. J., Ramos, A., Nunes, R., Azevedo, L., & Soares, A. (2016). Geostatistical Data Fusion: Application to Red Edge Bands of Sentinel 2. In *2016 International Conference on Computational Science and Computational Intelligence (CSCI)*. https://doi.org/10.1109/CSCI.2016.0147

PostgreSQL Documentation. (2017). Array Functions and Operators. Retrieved from https://www.postgresql.org/docs/9.6/static/functions-array.html

Provos, N., & Mazieres, D. (1999). A future-adaptable password scheme. *USENIX Annual Technical Conference, …*, 1–12. Retrieved from https://www.usenix.org/legacy/event/usenix99/full_papers/provos/provos.pdf

Rouse, J. W., Haas, R. H., Schell, J. A., & Deering, D. W. (1973). Monitoring Vegetation Systems in the Great Okains with ERTS. *Third Earth Resources Technology Satellite-1 Symposium*, *1*, 325–333.

Saraceno, E. (2003). Rural Development Policies and the Second Pillar of the Common Agricultural Policy. *Studies in Spatial Development*, *4*, 197–223. Retrieved from http://shop.arl-net.de/media/direct/pdf/ssd/ssd_4.pdf#page=204

SEGES. (2015). Vinterhvede. Retrieved from https://www.landbrugsinfo.dk/Planteavl/Plantevaern/IPM/Filer/Vinterhvede.pdf

SEGES. (2017a). *Dyrkningsvejledning - Rødsvingel frø*. Retrieved from https://www.landbrugsinfo.dk/Planteavl/Plantevaern/IPM/Filer/Dyrkningsvejledning_Roedsvingel_til_froe.pdf

SEGES. (2017b). *Dyrkningsvejledning - Vinterraps*. Retrieved from https://www.landbrugsinfo.dk/Planteavl/Plantevaern/IPM/Filer/vinterraps.pdf

SEOS. (2005). Resolution. Retrieved from http://www.seos-project.eu/modules/remotesensing/remotesensing-c03-p02.html

Smith, R., Adams, M., Maier, S., Craig, R., Kristina, A., & Maling, I. (2007). Estimating the area of stubble burning from the number of active fires detected by satellite. *Remote Sensing of Environment*, *109*(1), 95–106. https://doi.org/10.1016/j.rse.2006.12.011

Storey, J., Choate, M., & Dekota, S. (2014). Geometric and Spatial

Performance of Landsat 8. ASPRS 2014 Annual Conference & co-located JACIE Workshop.

Styrelsen for Dataforsyning og Effectivisering. (2017). Danmarks Højdemodel. Retrieved May 6, 2017, from http://sdfe.dk/hent-data/danmarks-hoejdemodel/

Sukkerroer.nu. (2017). Jord og sædskifte. Retrieved from https://www.sukkerroer.nu/irj/portal/nordzucker/da?NavigationTarget=navurl://d346a6a404b5f8f0e89947d8ad38f9a7

Sørensen, K. (2016). Dyrkning af sukkerroer. Retrieved from http://www.lf.dk/viden-om/landbrugsproduktion/planter/roer#

Telespazio. (2015a). *Level 2A Input Output Data Definition*. Retrieved from http://forum.step.esa.int/uploads/default/original/2X/4/438372b4348cc41d0dcbf74ead539922f49d4ea7.pdf

Telespazio. (2015b). *Level 2A Products Algorithm Theoretical Basis Document*. Retrieved from http://forum.step.esa.int/uploads/default/original/2X/9/93f71f2525db80beb28b5bcceece2700f8b364aa.pdf

Toller, G., & Isaacman, A. (2009). MODIS Level 1B Product User's Guide. *NASA*, 63. Retrieved from http://ccplot.org/pub/resources/Aqua/MODIS%5CnLevel%5Cn1B%5CnProduct%5CnUser%5CnGuide.pdf

Vajsova, B., & Åstrand, P. J. (2015). *New Sensors Benchmark Report on Sentinel-2A*. European Commission, Joint Research Centre. https://doi.org/10.2788/544302

Verhegghen, A., Eva, H., Ceccherini, G., Achard, F., Gond, V., Gourlet-Fleury, S., & Cerutti, P. (2016). The Potential of Sentinel Satellites for Burnt Area Mapping and Monitoring in the Congo Basin Forests. *Remote Sensing*, *8*(12), 986. https://doi.org/10.3390/rs8120986

Weisstein, E. W. (2017). z-Score. Retrieved from http://mathworld.wolfram.com/z-Score.html

Willis, K. S. (2015). Remote sensing change detection for ecological monitoring in United States protected areas. *Biological Conservation*. https://doi.org/10.1016/j.biocon.2014.12.006

Xue, J., & Su, B. (2017). Significant Remote Sensing Vegetation Indices: A Review of Developments and Applications. *Journal of Sensors*, *2017*(1353691), 1–17. https://doi.org/https://doi.org/10.1155/2017/1353691

Ya'acob, N., Azize, A. B. M., Mahmon, N. A., Yusof, A. L., Azmi, N. F., & Mustafa, N. (2014). Temporal Forest Change Detection and Forest Health Assessment using Remote Sensing. https://doi.org/10.1088/1755-1315/19/1/012017

Yadav, M., Prawasi, R., Jangra, S., Rana, P., Kumari, K., Lal, S., ... Hooda, R. S. (2014). Monitoring seasonal progress of rice stubble burning in major rice growing districts of Haryana, India, using multidate AWiFS data. In *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives* (Vol. XL-8, pp. 1003–1009). https://doi.org/10.5194/isprsarchives-XL-8-1003-2014

9. Appendix

I. Regulation (EU) No 1306/2013 Annex II

SMR : Statutory management requirement

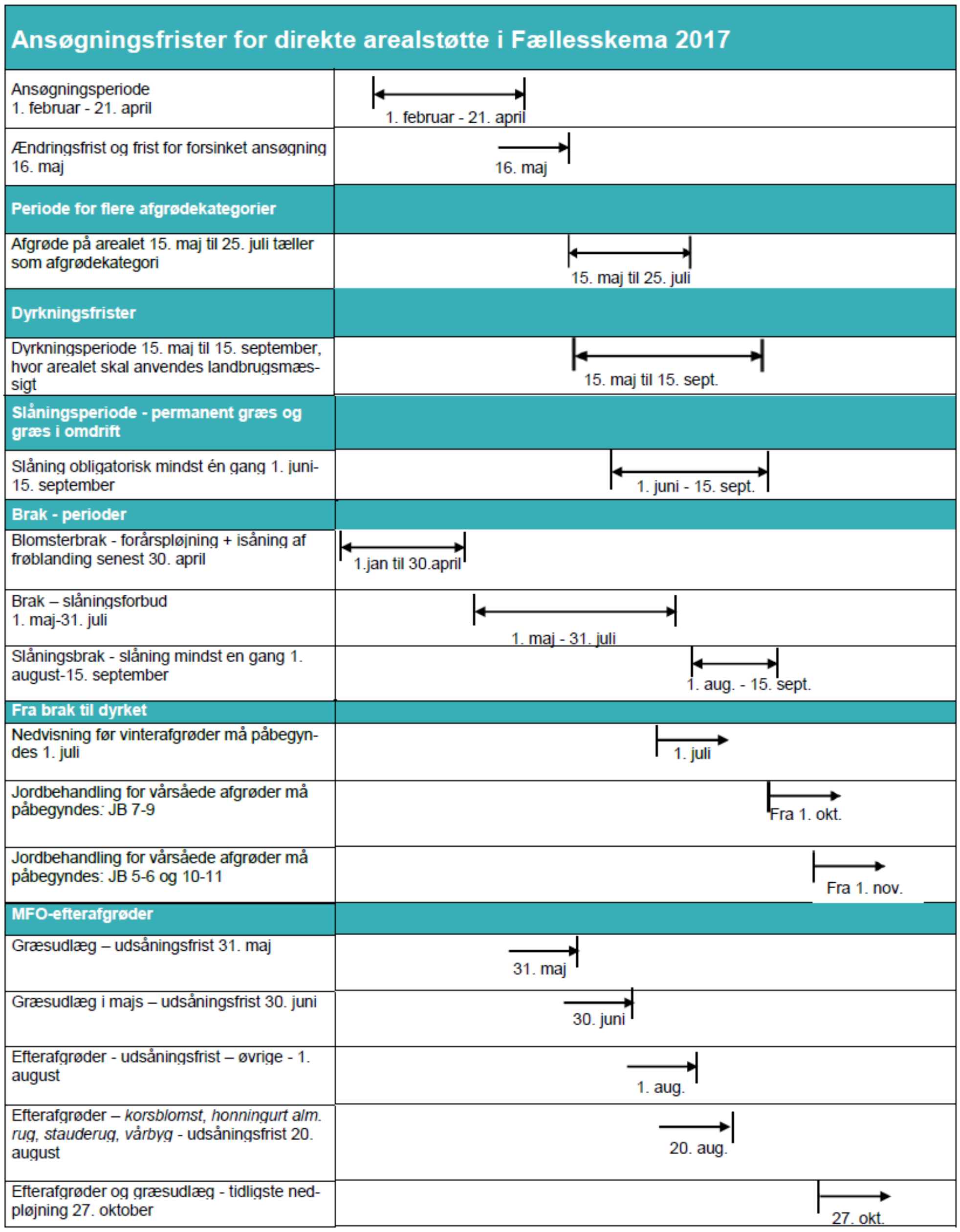
GAEC : Standards for good agricultural and environmental condition of land

Area	Main Issue	Requirements and standards		
Environment, climate change, good agricultural condition of land	Water	SMR 1	Council Directive 91/676/EEC of 12 December 1991 concerning the protection of waters against pollution caused by nitrates from agricultural sources (OJ L 375, 31.12.1991, p. 1)	Articles 4 and 5
		GAEC 1	Establishment of buffer strips along water courses [1]	
		GAEC 2	Where use of water for irrigation is subject to authorisation, compliance with authorisation procedures	
		GAEC 3	Protection of ground water against pollution: prohibition of direct discharge into groundwater and measures to prevent indirect pollution of groundwater through discharge on the ground and percolation through the soil of dangerous substances, as listed in the Annex to Directive 80/68/EEC in its version in force on the last day of its validity, as far as it relates to agricultural activity	
	Soil and carbon stock	GAEC 4	Minimum soil cover	
		GAEC 5	Minimum land management reflecting site specific conditions to limit erosion	
		GAEC 6	Maintenance of soil organic matter level through appropriate practices including ban on burning arable stubble, except for plant health reasons [2]	
	Biodiversity	SMR 2	Directive 2009/147/EC of the European Parliament and of the Council of 30 November 2009 on the conservation of wild birds (OJ L 20, 26.1.2010, p. 7)	Article 3(1), Article 3(2)(b), Article 4(1), (2) and (4)
		SMR 3	Council Directive 92/43/EEC of 21 May 1992 on the conservation of natural habitats and of wild flora and fauna (OJ L 206, 22.7.1992, p. 7)	Article 6(1) and (2)
	Landscape, minimum level of	GAEC 7	Retention of landscape features, including where appropriate, hedges, ponds, ditches, trees in line, in group or isolated, field margins and terraces, and including a ban on cutting hedges and trees during	

	maintenance		the bird breeding and rearing season and, as an option, measures for avoiding invasive plant species	
Public health, animal health and plant health	Food safety	SMR 4	Regulation (EC) No 178/2002 of the European Parliament and of the Council of 28 January 2002 laying down the general principles and requirements of food law, establishing the European Food Safety Authority and laying down procedures in matters of food safety (OJ L 31, 1.2.2002, p. 1)	Articles 14 and 15, Article 17(1) [3] and Articles 18, 19 and 20
		SMR 5	Council Directive 96/22/EC of 29 April 1996 concerning the prohibition on the use in stockfarming of certain substances having a hormonal or thyrostatic action and beta-agonists, and repealing Directives 81/602/EEC, 88/146/EEC and 88/299/EEC (OJ L 125, 23.5.1996, p. 3)	Article 3(a), (b), (d) and (e) and Articles 4, 5 and 7
	Identification and registration of animals	SMR 6	Council Directive 2008/71/EC of 15 July 2008 on identification and registration of pigs (OJ L 213, 8.8.2005, p. 31)	Articles 3, 4 and 5
		SMR 7	Regulation (EC) No 1760/2000 of the European Parliament and of the Council of 17 July 2000 establishing a system for the identification and registration of bovine animals and regarding the labelling of beef and beef products and repealing Council Regulation (EC) No 820/97(OJ L 204, 11.8.2000, p. 1)	Articles 4 and 7
		SMR 8	Council Regulation (EC) No 21/2004 of 17 December 2003 establishing a system for the identification and registration of ovine and caprine animals and amending Regulation (EC) No 1782/2003 and Directives 92/102/EEC and 64/432/EEC (OJ L 5, 9.1.2004, p. 8)	Articles 3, 4 and 5
	Animal diseases	SMR 9	Regulation (EC) No 999/2001 of the European Parliament and of the Council of 22 May 2001 laying down rules for the prevention, control and eradication of certain transmissible spongiform encephalopathies (OJ L 147, 31.5.2001, p. 1)	Articles 7, 11, 12, 13 and 15
	Plant protection products	SMR 10	Regulation (EC) No 1107/2009 of the European Parliament and of the Council of 21 October 2009 concerning the placing of plant protection products on the market and repealing Council Directives 79/117/EEC and 91/414/EEC (OJ L 309, 24.11.2009, p. 1)	Article 55, first and second sentence
Animal welfare	Animal welfare	SMR 11	Council Directive 2008/119/EC of 18 December 2008 laying down minimum standards for the protection of calves (OJ L 10, 15.1.2009, p. 7)	Articles 3 and 4
		SMR 12	Council Directive 2008/120/EC of 18 December 2008 laying down minimum standards for the	Article 3 and Article 4

			protection of pigs (OJ L 47, 18.2.2009, p. 5)	
		SMR 13	Council Directive 98/58/EC of 20 July 1998 concerning the protection of animals kept for farming purposes(OJ L 221, 8.8.1998, p. 23)	Article 4

II. Crop Deadlines



Ansøgningsfrister for direkte arealstøtte i Fællesskema 2017	
Ansøgningsperiode 1. februar - 21. april	1. februar - 21. april
Ændringsfrist og frist for forsinket ansøgning 16. maj	16. maj
Periode for flere afgrødekategorier	
Afgrøde på arealet 15. maj til 25. juli tæller som afgrødekategori	15. maj til 25. juli
Dyrkningsfrister	
Dyrkningsperiode 15. maj til 15. september, hvor arealet skal anvendes landbrugsmæssigt	15. maj til 15. sept.
Slåningsperiode - permanent græs og græs i omdrift	
Slåning obligatorisk mindst én gang 1. juni-15. september	1. juni - 15. sept.
Brak - perioder	
Blomsterbrak - forårspløjning + isåning af frøblanding senest 30. april	1.jan til 30.april
Brak – slåningsforbud 1. maj-31. juli	1. maj - 31. juli
Slåningsbrak - slåning mindst en gang 1. august-15. september	1. aug. - 15. sept.
Fra brak til dyrket	
Nedvisning før vinterafgrøder må påbegyndes 1. juli	1. juli
Jordbehandling for vårsåede afgrøder må påbegyndes: JB 7-9	Fra 1. okt.
Jordbehandling for vårsåede afgrøder må påbegyndes: JB 5-6 og 10-11	Fra 1. nov.
MFO-efterafgrøder	
Græsudlæg – udsåningsfrist 31. maj	31. maj
Græsudlæg i majs – udsåningsfrist 30. juni	30. juni
Efterafgrøder - udsåningsfrist – øvrige - 1. august	1. aug.
Efterafgrøder – *korsblomst, honningurt alm. rug, stauderug, vårbyg* - udsåningsfrist 20. august	20. aug.
Efterafgrøder og græsudlæg - tidligste nedpløjning 27. oktober	27. okt.

III. Parsing Request

```
const request = require('request')
const bodyParser = require('body-parser')
const sunCalc = require('suncalc')
const turf = require('@turf/turf')
const xmldoc = require('xmldoc')
const database = require('../database.js')
const credentials = database.credentials
const NASAkey = database.NASAkey
const utc = require('../geom/utc')
const helper = require('./helpers')

var external = function (obj, callback) {
  var timeOut = setTimeout(function () {
    callback({
      'status': 'error',
      'message': 'Waiting for images took too long..'
    })
  }, 1000 * 60 * 2) // Two minutes

  var returnArray = []
  var key = NASAkey
  var params = {
    'date': '2017.01.01',
    'sentinel1': false,
    'sentinel2': true,
    'sentinel3': false,
    'landsat8': false,
    'geometry': {}
  }

  // CHECK: Does the request have a geometry column?
  if (obj && obj.geometry) {
    params.geometry.geojson = obj.geometry
  } else {
    clearTimeout(timeOut)
    callback({
      'status': 'error',
      'message': 'geometry invalid'
    })
  }

  // Shortcut
  var geom = params.geometry.geojson

  // Convert the strings of the array to numbers
  helper.toNumber(geom.geometry.coordinates[0])

  // Calculate geometry for requests
  params.geometry.array = geom.geometry.coordinates[0]
  params.geometry.wkt =
helper.arr2wkt(params.geometry.array)
   params.geometry.center = turf.centroid(geom)

   // Find timezone of geometry and add it to the params
   for (var j = 0; j < utc.features.length; j += 1) {
    if (turf.inside(params.geometry.center, utc.features[j])
=== true){
       params.geometry.timezone =
utc.features[j].properties.zone
       break
     }
   }

   // Count how many requests to make
   var totalSatellites = 0
   if (params.sentinel1 === true ||
params.sentinel2 === true) {
     totalSatellites += 1
   }
   if (params.sentinel3 === true) { totalSatellites += 1 }
   if (params.landsat8 === true) { totalSatellites += 1 }

   // Each time a request is finished
   var finished = 0
   var finishCheck = function () {
     finished += 1
     if (finished === totalSatellites) {
       clearTimeout(timeOut)
       callback({
         'status': 'success',
         'message': returnArray
       })
     }
   }
```

IV. Fetch Sentinel 1 & 2

```
  if (params.sentinel2 === true ||
    params.sentinel1 === true) {
     var startRow = 0
     var baseUrl =
'https://scihub.copernicus.eu/dhus/search?q='
     var end = `&start=${startRow}&rows=100`
     var platforms
     if (params.sentinel1 === true &&
       params.sentinel2 === true) { platforms = '' }
     if (params.sentinel1 === true &&
       params.sentinel2 === false) { platforms = ' AND
       platformname:Sentinel-1' }
     if (params.sentinel1 === false &&
       params.sentinel2 === true) { platforms = ' AND
       platformname:Sentinel-2' }
     var esaRequest =
`${baseUrl}footprint:"Intersects(${params.geometry.wkt})"${pl
atforms} AND beginposition:[${params.date}T00:00:00.000Z TO
NOW]${end}`
     var entries = []

     request.get(esaRequest, {
       'auth': credentials.main,
       'gzip': true
     }, function (error, response, result) {
       if (!error && response.statusCode === 200) {
         var esa = new xmldoc.XmlDocument(result)
         var nrEntries = Number(esa.childNamed(
           'opensearch:totalResults').val)

         var nrSearches = helper.getSearches(nrEntries) - 1
         var completed = 0
         entries = esa.childrenNamed('entry')

         if (nrSearches === 0) {
           entries = helper.parseXML(entries,
             params.geometry.timezone)

           for (var i = 0; i < entries.length; i += 1) {
             returnArray.push(entries[i])
           }
           finishCheck()
         } else {
           for (var j = 0; j < nrSearches; j += 1) {
             var newStartRow = (j + 1) * 100
             var newEnd = `&start=${newStartRow}&rows=100`
             var newEsaRequest =
`${baseUrl}footprint:"Intersects(${params.geometry.wkt})"
 ${platforms} AND beginposition:[${params.date}T00:00:00.000Z
 TO NOW]${newEnd}`
```

```
            request.get(newEsaRequest, {
              'auth': credentials.secondary,
              'timeout': 1200000,
              'gzip': true
            }, function (error, response, result) {
              if (!error && response.statusCode === 200) {
                completed += 1
                var esa = new xmldoc.XmlDocument(result)
                entries = entries.concat(
                  esa.childrenNamed('entry'))

                if (completed === nrSearches) {
                  entries = helper.parseXML(entries,
                    params.geometry.timezone)
                  for (var i = 0; i < entries.length;
                    i += 1) {
                      returnArray.push(entries[i])
                  }
                  finishCheck()
                }
              } else { console.log('error: ' + error) }
            })
          }
        }
      } else { console.log('error: ' + error) }
    })
  }
}
```

V. Fetch Landsat

```
if (params.landsat8 === true) {
  var landsat = {
    'count_NASA': 0,
    'count_AMAZON': 0,
    'id': []
  }

  // MINUS ONE BECAUSE GEOJSON ARRAYS REPEATS THE LAST POINT
TO CLOSE POLYGONS
  for (var i = 0; i < params.geometry.array.length - 1;
    i += 1) {
    var link =
`https://api.nasa.gov/planetary/earth/assets?lon=${params.geo
metry.array[i][0]}&lat=${params.geometry.array[i][1]}&begin=$
{params.date}&api_key=${key}`

    // Request NASA for images of each of
    // the points in the geometry
    request({
      method: 'GET',
      uri: link
    }, function (error, response, body) {
      if (error) { console.log(error) }

      // Parse the reply and add it to the list of unique IDs
      var bodyParse = JSON.parse(body)
      for (var j = 0; j < bodyParse.results.length; j += 1) {
        var post = bodyParse.results[j].id.split('/')[1]
        if (landsat.id.indexOf(post) === -1) {
          landsat.id.push(post)
        }
      }

      // Keeps track of how many points are returned.
      landsat.count_NASA += 1

      // MINUS ONE BECAUSE GEOJSON ARRAYS
      // REPEATS THE LAST POINT
      if (landsat.count_NASA === params.geometry.array
        .length - 1) {
        if (landsat.id.length === 0) { finishCheck() }

        for (var i = 0; i < landsat.id.length; i += 1) {
          var id = landsat.id[i]
          var row = id.slice(3, 6)
          var path = id.slice(6, 9)

          var metaLink = `http://landsat-
pds.s3.amazonaws.com/L8/${row}/${path}/${id}/${id}_MTL.txt`

          request({
```

```
      method: 'GET',
      uri: metaLink
     }, function (error, response, body) {
      if (error) { console.log(error) }
      var preFormat = helper.parseLandsatMetadata(body)

      if (preFormat.METADATA_FILE_INFO
       .LANDSAT_SCENE_ID) {
       var _id = preFormat.METADATA_FILE_INFO
        .LANDSAT_SCENE_ID
       var _row = _id.slice(3, 6)
       var _path = _id.slice(6, 9)
       var main = `http://landsat-
pds.s3.amazonaws.com/L8/${_row}/${_path}/${_id}/index.html`
       var thumbnail = `http://landsat-
pds.s3.amazonaws.com/L8/${_row}/${_path}/${_id}/${_id}_thumb_
large.jpg`

       var UTCTime = preFormat.PRODUCT_METADATA
        .DATE_ACQUIRED + 'T' +
        preFormat.PRODUCT_METADATA +
        .SCENE_CENTER_TIME
       var localTime = helper.toLocaltime(
        UTCTime, params.geometry.timezone)

       var S_PM = preFormat.PRODUCT_METADATA

       var NW = [Number(S_PM.CORNER_UL_LON_PRODUCT),
        Number(S_PM.CORNER_UL_LAT_PRODUCT)]
       var NE = [Number(S_PM.CORNER_UR_LON_PRODUCT),
        Number(S_PM.CORNER_UR_LAT_PRODUCT)]
       var SW = [Number(S_PM.CORNER_LL_LON_PRODUCT),
        Number(S_PM.CORNER_LL_LAT_PRODUCT)]
       var SE = [Number(S_PM.CORNER_LR_LON_PRODUCT),
        Number(S_PM.CORNER_LR_LAT_PRODUCT)]
       var polygon = turf.polygon([[NW, NE, SE, SW,
       NW]])
      // GeoJSON always repeats first and last entry

       var replyLandsat = helper.cloneObject(
        helper.defaultReply)
       replyLandsat.id = _id
       replyLandsat.satellite.name = 'Landsat-8'
       replyLandsat.satellite.sensor = 'OLI'
       replyLandsat.date.UTC = UTCTime
       replyLandsat.date.local = localTime
       replyLandsat.footprint = turf.truncate(
        turf.convex(polygon), 5, 2)
        // precision 5, no z-coordinates
       replyLandsat.clouds.radar = false
       replyLandsat.clouds.cover = preFormat.
        IMAGE_ATTRIBUTES.CLOUD_COVER
       replyLandsat.sun.altitude = preformat
```

```
                .IMAGE_ATTRIBUTES.SUN_ELEVATION
              replyLandsat.sun.azimuth = preformat
                .IMAGE_ATTRIBUTES.SUN_AZIMUTH

              returnArray.push(replyLandsat)
            } else {
              var message = new xmldoc
                .XmlDocument(response.body)
              var messageID = message.childNamed('Key').val;
                messageID = messageID.split('/')[3]
              console.log('Error getting: ' + messageID)
            }

            landsat.count_AMAZON += 1
            if (landsat.count_AMAZON === landsat.id.length){
              finishCheck()
            }
          }
          )
        }
      }
    }
   )
  }
}
```

VI. Fetch ASTER

```
var options = {
    'token': '23FFE8BF-CDC1-486B-B5E3-F2B7F55CE151',
    'center': 'LPDAAC_ECS',
    'shortname': 'AST_L1T',
    'version': '003',
    'begin': '2017-03-01',
    'end': '2017-03-20',
    'mode': 'coordinates',
    'urlat': 60,
    'urlon': 16,
    'lllat': 51,
    'llon': 6,
    'minhoriz': 0,
    'maxhoriz': 35,
    'minvert': 0,
    'maxvert': 17,
    'metadata': 'on'
  }

  // request via post
  $.ajax({
    'url': 'https://dartool.cr.usgs.gov/cgi-
bin/Daac2Disk.cgi',
    'type': 'POST',
    'data': data,
    'dataType': 'xml',
    'crossDomain': true,
    'timeout': 600000,
    'accepts': {
      'xml': 'text/xml',
      'text': 'text/xml'
    }
  })
     .done(function (xml) {
       var metadata = []
       $(xml).find('metadata').each(function () {
         metadata.push($(this).text())
       })
       parseMeta (metadata)
     })
     .fail(function (jqxhr, textStatus, errorThrown) {
       alert('ERROR: ' + textStatus + ' | ' + errorThrown)
     })

  // Parse Metadata
  var parseMeta = function (arr) {
    var totalCount = arr.length
    var count = 0
    var dayImages = []
    for (var i = 0; i < arr.length; i += 1) {
```

```
    var metadataLink = arr[i]

    $.ajax({
      'url': metadataLink,
      'type': 'get',
      'dataType': 'xml',
      'crossDomain': true
    })
       .done(function (xml) {
         count += 1
         $(xml).find('DayNightFlag').each(function () {
           var dayOrNight = $(this).text()
           if (dayOrNight === 'day') {
             dayImages.push(metadataLink)
           }
         })
         if (count === totalCount) { return dayImages }
       })
       .fail(function (jqxhr, textStatus, errorThrown) {
         alert('ERROR: ' + textStatus + ' | ' +
           errorThrown)
       })
  }
}

// Prepare URL for encoding
var data = ''
for (var prop in options) {
  if (options.hasOwnProperty(prop)) {
    data += prop + '=' + options[prop] + '&'
  }
}
data = data.slice(0, data.length - 1)
```

VII. Indices and Variation

```
// Calculate Outliers
const fs = require('fs')

const options = {
  inputFile: 'projectFields2015.geojson',
  outputGeoJSON: 'disp2015.geojson',
  outputSummary: 'summ2015.csv',
  cropName: 'AfgNavn',
  cropArea: 'IndtAreal',
  cropNDVI: 'ndvi_mean',
  cropRENDVI: 'rendvi_mea',
  cropNDVI_std: 'ndvi_stdev',
  cropRENDVI_std: 'rendvi_std',
  encoding: 'utf8',
  decimals: 5,
  cutoffCount: 25,  // at least 10 of field type
  cutoffArea: 10,   // single field at least 10ha
  cutoffIndex: 2,
  // >= 2 standard deviations from weighted mean
  cutoffVariance: 2
  // >= 2 standard deviations from weighted mean
}

// Calculate trainingSet
const trainingOptions = {
  standardDeviation: 2,
  // <= std in internal std and index std
  fieldsCount: 100,   // atleast 100 fields of type
  fieldsTotal: 200,   // atleast 200 ha of type
  fieldSingle: 2      // atleast 2 ha
}

var sum_crop_CSV =
'crop;count;totalArea;NDVIweightedMean;NDVIstandardDevi
ation;RENDVIweightedMean;RENDVIstandardDeviation;intern
alNDVIstandardDeviation;internalRENDVIstandardDeviation
;\n'
var uniques = {}

fs.readFile(options.inputFile, options.encoding,
function (err, data) {
  if (err) { return console.log(err) }
  var geojson = JSON.parse(data)
  var features = geojson.features

  // FIND UNIQUES AND AREAS
  for (var i = 0; i < features.length; i += 1) {
    var cropProp = features[i].properties
    var cropName = cropProp[options.cropName]
    if (!uniques[cropName]) {
      uniques[cropName] = {
        count: 1,
        totalArea: cropProp[options.cropArea],
        NDVI: {
          variance: 0,
          weightedMean: 0,
          standardDeviation: 0,
          numbers: []
        },
        RENDVI: {
          variance: 0,
          weightedMean: 0,
          standardDeviation: 0,
          numbers: []
        },
        internal: {
          NDVI: {
            weightedMeanVariance: 0,
            standardDeviation: 0,
            numbers: []
          },
          RENDVI: {
            weightedMeanVariance: 0,
            standardDeviation: 0,
            numbers: []
          }
        }
      }
    } else {
      uniques[cropName].count += 1
      uniques[cropName].totalArea +=
cropProp[options.cropArea]
    }
  }

  // CALCULATE WEIGHTED MEAN
  for (var j = 0; j < features.length; j += 1) {
    var cropProp = features[j].properties
    var cropName = cropProp[options.cropName]

    var ndviMean = cropProp[options.cropNDVI]
    var ndviVariance = Math.pow(cropProp[
      options.cropNDVI_std], 2)

    var rendviMean = cropProp[options.cropRENDVI]
    var rendviVariance = Math.pow(cropProp[
      options.cropRENDVI_std], 2)

    if (cropProp[ndviMean] === '' || cropProp[
      rendviMean] === '') {
      cropProp[ndviMean] = null
      cropProp[rendviMean] = null
    } else {
      var typeArea = uniques[cropName].totalArea
      var weight = cropProp[options.cropArea] /
        typeArea
      var ndvi_weight = ndviMean * weight
      var rendvi_weight = rendviMean * weight
      var internalWeightNDVI = ndviVariance * weight
      var internalWeightRENDVI = rendviVariance *
        weight


uniques[cropName].internal.NDVI.weightedMeanVariance +=
  internalWeightNDVI
      uniques[cropName].internal.NDVI.numbers.push(
        ndviVariance)
      uniques[cropName].NDVI.weightedMean +=
        ndvi_weight
      uniques[cropName].NDVI.numbers.push(ndviMean)
      uniques[cropName].internal.RENDVI
        .weightedMeanVariance += internalWeightRENDVI
      uniques[cropName].internal.RENDVI.numbers.push(
        rendviVariance)
      uniques[cropName].RENDVI.weightedMean +=
        rendvi_weight
      uniques[cropName].RENDVI.numbers.push(rendviMean)
    }
  }

  // CALCULATE STANDARD DEVIATION TO WEIGHTED MEAN
  for (var key in uniques) {
    if (uniques.hasOwnProperty(key)) {
      var ndviArray = uniques[key].NDVI.numbers
      var ndviMean = uniques[key].NDVI.weightedMean
      var rendviArray = uniques[key].RENDVI.numbers
      var rendviMean = uniques[key].RENDVI.weightedMean

      var ndviArrayVariance = uniques[key]
        .internal.NDVI.numbers
      var ndviMeanVariance = uniques[key].internal.NDVI
        .weightedMeanVariance
      var rendviArrayVariance =
        uniques[key].internal.RENDVI.numbers
      var rendviMeanVariance = uniques[key]
        .internal.RENDVI.weightedMeanVariance

      if (ndviArray.length < options.cutoffCount) {
        uniques[key].NDVI.variance = null
        uniques[key].NDVI.standardDeviation = null
      } else {
        uniques[key].NDVI.variance = mth
          .varianceCustom(ndviArray, ndviMean)
        uniques[key].NDVI.standardDeviation = Math
          .sqrt(uniques[key].NDVI.variance)
      }
      if (rendviArray.length < options.cutoffCount) {
        uniques[key].RENDVI.variance = null
        uniques[key].RENDVI.standardDeviation = null
      } else {
        uniques[key].RENDVI.variance = mth
          .varianceCustom(rendviArray, rendviMean)
        uniques[key].RENDVI.standardDeviation = Math
          .sqrt(uniques[key].RENDVI.variance)
      }

      if (ndviArrayVariance.length < options.
        cutoffCount) {
        uniques[key].internal.NDVI
          .standardDeviation = null
      } else {
        uniques[key].internal.NDVI.
          standardDeviation = Math.sqrt(mth
          .varianceCustom(ndviArrayVariance,
          ndviMeanVariance))
      }

      if (rendviArrayVariance.length < options
        .cutoffCount) {
        uniques[key].internal.RENDVI
          .standardDeviation = null
      } else {
        uniques[key].internal.RENDVI
        .standardDeviation = Math.sqrt(mth.
        varianceCustom(rendviArrayVariance,
        rendviMeanVariance))
      }

      delete uniques[key].NDVI.numbers
      delete uniques[key].RENDVI.numbers
      delete uniques[key].internal.NDVI.numbers
      delete uniques[key].internal.RENDVI.numbers

      var crop = uniques[key]
      sum_crop_CSV +=
`${key};${crop.count};${crop.totalArea};${crop.NDVI.wei
ghtedMean};${crop.NDVI.standardDeviation};${crop.RENDVI
.weightedMean};${crop.RENDVI.standardDeviation};${crop.
internal.NDVI.standardDeviation};${crop.internal.RENDVI
.standardDeviation};\n`
    }
  }

  var zScore = function (mean, std, number) {
    return (number - mean) / std
  }

  // Rounding function by Jack Moore (@StackOverflow)
  function round (value, decimals) {
```

```
  return Number(Math.round(value + 'e' + decimals) +
   'e-' + decimals)
 }

 var trainingSet = {
  'type': 'FeatureCollection',
  'features': []
 }

 for (var q = 0; q < geojson.features.length; q += 1)
{
  var cropProp = geojson.features[q].properties
  var cropName = cropProp[options.cropName]
  var ndviMean = cropProp[options.cropNDVI]
  var rendviMean = cropProp[options.cropRENDVI]
  var ndviMeanVariance = Math.pow(cropProp[
   options.cropNDVI_std], 2)
  var rendviMeanVariance = Math.pow(cropProp[
   options.cropRENDVI_std], 2)

  cropProp.t_Count = uniques[cropName].count
  cropProp.t_NDVI_Mean = uniques[cropName]
   .NDVI.weightedMean
  cropProp.t_NDVI_STD = uniques[cropName]
   .NDVI.standardDeviation
  cropProp.t_RENDVI_Mean = uniques[cropName]
   .RENDVI.weightedMean
  cropProp.t_RENDVI_STD = uniques[cropName]
   .RENDVI.standardDeviation

  cropProp.t_NDVI_varMean = uniques[cropName]
   .internal.NDVI.weightedMeanVariance
  cropProp.t_NDVI_varStd = uniques[cropName]
   .internal.NDVI.standardDeviation
  cropProp.t_RENDVI_varMean = uniques[cropName]
   .internal.RENDVI.weightedMeanVariance
  cropProp.t_RENDVI_varStd = uniques[cropName]
   .internal.RENDVI.standardDeviation

  cropProp.controlIndex = false
  cropProp.controlVariance = false
  cropProp.controlOBS = false

  cropProp.z_NDVI = zScore(cropProp.t_NDVI_Mean,
   cropProp.t_NDVI_STD, ndviMean)
  cropProp.z_NDVI_int = zScore(
   cropProp.t_NDVI_varMean, cropProp.t_NDVI_varStd,
   ndviMeanVariance)
  cropProp.z_RENDVI = zScore(
   cropProp.t_RENDVI_Mean, cropProp.t_RENDVI_STD,
   rendviMean)
  cropProp.z_RENDVI_int = zScore(
   cropProp.t_RENDVI_varMean,
   cropProp.t_RENDVI_varStd, rendviMeanVariance)

  if (uniques[cropName].count > options
   .cutoffCount && cropProp[options.cropArea] >
   options.cutoffArea) {
   if (cropProp.z_NDVI > options.cutoffIndex ||
     cropProp.z_NDVI < (options.cutoffIndex * -1)) {
     cropProp.controlIndex = true
   }
   if (cropProp.z_RENDVI > options.cutoffIndex ||
   cropProp.z_RENDVI < (options.cutoffIndex * -1)) {
     cropProp.controlIndex = true
   }
   if (cropProp.z_NDVI_int >
     options.cutoffVariance || cropProp.z_NDVI_int <
     (options.cutoffVariance * -1)) {
     cropProp.controlVariance = true
   }
   if (cropProp.z_RENDVI_int > options.
     cutoffVariance || cropProp.z_RENDVI_int <
    (options.cutoffVariance * -1)) {
     cropProp.controlVariance = true
   }
   if (cropProp.controlIndex === true && cropProp.
     controlVariance === false) {
     cropProp.controlOBS = true
   }
  }

  // ROUND ALL NUMBERS TO DECIMALS
  for (var key in cropProp) {
   if (cropProp.hasOwnProperty(key)) {
     if (typeof (cropProp[key]) === 'number') {
       cropProp[key] = round(cropProp[key],
       options.decimals) }
   }
  }

  // CHECK AND ADD TO TRAINING SET
  if (cropProp.z_NDVI < trainingOptions
   .standardDeviation && cropProp.z_NDVI >
   (trainingOptions.standardDeviation * -1) &&
   cropProp.z_RENDVI < trainingOptions
   .standardDeviation && cropProp.z_RENDVI >
   (trainingOptions.standardDeviation * -1) &&
   cropProp.z_NDVI_int < trainingOptions
   .standardDeviation && cropProp.z_NDVI_int >
   (trainingOptions.standardDeviation * -1) &&
   cropProp.z_RENDVI_int < trainingOptions
   .standardDeviation && cropProp.z_RENDVI_int >
   (trainingOptions.standardDeviation * -1)) {
   if (uniques[cropName].count > trainingOptions
    .fieldsCount && uniques[cropName].totalArea >
```

```
        trainingOptions.fieldsTotal &&
        cropProp[options.cropArea] >
        trainingOptions.fieldSingle) {
        trainingSet.features.push(geojson.features[q])
      }
    }
  }

  // WRITE SUMMARY
  fs.writeFile(options.outputSummary, sum_crop_CSV,
  function (err) {
    if (err) return console.log(err)
  })

  // WRITE GEOJSON
  fs.writeFile(options.outputGeoJSON,
  JSON.stringify(geojson), function (err) {
    if (err) return console.log(err)
  })

  // WRITE TRANINGSET
  fs.writeFile('trainingSet.geojson',
    JSON.stringify(trainingSet), function (err) {
    if (err) return console.log(err)
  })
})

var mth = {
  sum: function (array) {
    var num = 0
    for (var i = 0; i < array.length; i += 1) {
      num += array[i]
    }
    return num
  },
  mean: function (array) {
    return this.sum(array) / array.length
  },
  varianceCustom: function (array, weightedMean) {
    return this.mean(
      array.map(function (num) {
        return Math.pow(num - weightedMean, 2)
      }))
  }
}
```

VIII. Calculating Normalised Homogeneity Index

```
const fs = require('fs')

fs.readFile('falsterIntersect2015.geojson', 'latin1',
function (err, data) {
  if (err) { return console.log(err) }
  var geojson = JSON.parse(data)
  var features = geojson.features
  var uniques = {}

  for (var i = 0; i < features.length; i += 1) {
    var properties = features[i].properties

    if (uniques[properties.FID_2015_f]) {
      uniques[properties.FID_2015_f].sum +=
      properties.segArea
    } else {
      uniques[properties.FID_2015_f] = {
        sum: properties.segArea
      }
    }
  }

  for (var j = 0; j < features.length; j += 1) {
    var properties = features[j].properties
    var thisSum = uniques[properties.FID_2015_f].sum
    var thisScore = properties.segArea /
      (2 * thisSum - properties.segArea)
    if (uniques[properties.FID_2015_f].score) {
      uniques[properties.FID_2015_f].score +=
      thisScore
    } else {
      uniques[properties.FID_2015_f].score = thisScore
    }
  }

  var csv = 'nFID;score;\n'

  for (var key in uniques) {
    if (uniques.hasOwnProperty(key)) {
      uniques[key].score = (uniques[key].score - 0.5)
      / 0.5
      csv += `${key};${uniques[key].score};\n`
    }
  }

          // WRITE TRANINGSET
  fs.writeFile('score.csv', csv, function (err) {
    if (err) return console.log(err)
  })
})
```