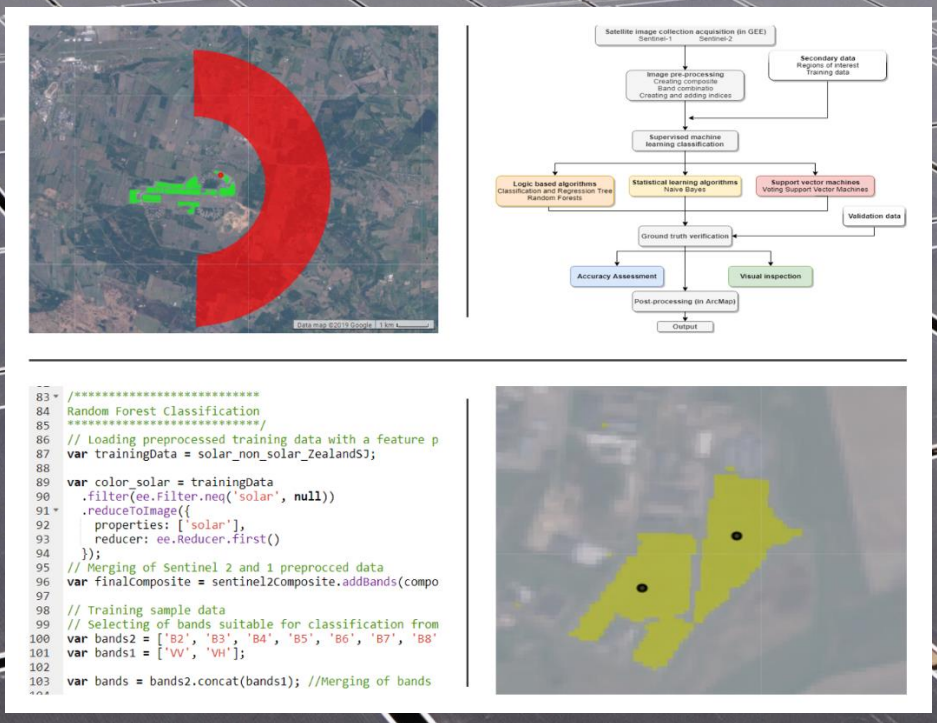
AALBORG UNIVERSITY
STUDENT REPORT

An exploration of automatic detection of large-scale solar plants: application of machine learning-based image classification in Google Earth Engine

(Master Thesis)



Marek Vasku
Surveying, Planning and Land Management
Geoinformatics MSc. studies, 4th Semester
Copenhagen, Summer 2019

Abstract:

This thesis contains a study of automatic detection of large-scale solar plants from satellite imagery (Sentinel-1 and Sentinel-2) using geospatial cloud-based platform – Google Earth Engine. The thesis investigates to what extent and with what accuracy is possible to detect large-scale solar plants using specific remote sensing data. The study is connected with the existing research and approach and tries to provide a more accessible way using available processing tools. The approach and process are commented using coding samples, workflow scheme, and the results are compared using different machine learning classifiers. The analysis is conducted using Denmark as a study area with known ground-truth data, but the approach is ultimately transferred for detecting large-scale solar plants in the selected Germany state.

Title:

An exploration of automatic detection of large-scale solar plants: application of machine learning-based image classification in Google Earth Engine

Keywords:

Remote Sensing, Supervised Classification, Machine Learning, Earth Engine, Solar Energy

Author:

Marek Vasku

Supervisor:

Jamal Jokar Arsanjani

Project period: Feb 2019 – Jun 2019 (4th semester of Master programme)

Education: MSc in Geoinformatics

University: Aalborg University Copenhagen, Denmark

TABLE OF CONTENTS

List of figures, tables, equations and codes

List of acronyms

API - application programming interface

CART - Classification and Regression Tree

CNN - Convolutional Neural Network

DAWA - Denmark Addresses Web API

ESA – European Space Agency

GEE – Google Earth Engine

GMES - Global Monitoring for Environment and Security

GRD - Ground Range Detected

NDVI – Normalized Difference Vegetation Index

NDWI - Normalized Difference Water Index

PV – photovoltaics

SAR - synthetic-aperture radar

SPOT - *Satellite Pour l'Observation de la Terre* (French) - "Satellite for observation of Earth"

SVM – Support Vector Machines

1 INTRODUCTION

Humans have always been attracted by viewing the world from above. Every single day, there is a vast number of satellite images comprehensively taken by dozens of satellites orbiting Earth and containing a large amount of information. However, the broad range and amount of data, which is updated daily, exceeds the opportunity to process these data with manual analysis. It is only possible to manage to work with this data using entirely or semi-automated tools (Ishii et al. 2017). Another fact is that with still a growing rise of global warming of Earth, also the investments on clean energy have been increasing, not excluding solar plants investments and constructions. The photovoltaic (PV) capacity worldwide exceeded 270 GW between years 2010 and 2016 (Imamoglu et al. 2017).

There is an increased demand for the objects detected from space using remote sensing in recent years, including large solar installations in the academia and simultaneously by commercial companies and businesses. Companies usually use high-resolution aerial imagery and analyzing only the small portion of the area (for example cities or just part of cites). Not extensive academic research, on the other hand, uses less precise spatially resolution imagery but conduct their analysis using Convolutional Neural Network (CNN) on large datasets without using capabilities of cloud computing. This research is usually lead by the departments of Computer Science / Informatics without using the geography perspective on the subject.

These facts lead to establishing a thesis project idea whether and to which extent we can observe, monitor, and collect data about solar energy objects, especially large-scale solar (PV) plants. Solar power plants are in a key position in renewable energy planning. Detection of solar power plants empowers authorities to plan and estimate energy production. Moreover, the direct light is needed for optimal functionality of solar power plants; thus, these objects are usually located in a place fully visible for satellite imagery (Ishii et al. 2017). Furthermore, the data for these installations often lacks precise geospatial information as they are registered to address to an adjacent building site or farm, and this makes the potential geospatial analysis using these data difficult with biased results. There is a fast-paced building-up of solar

plants, and data might become overdue quickly. With modern tools such as Google Earth Engine (GEE), it became easier and straightforward to monitor the changes in the land cover and the methods for usage rising invariably. These assumptions lead to defining main research questions and sub-questions of this thesis project.

1.1 Problem statement and research questions

The purpose of this study is to fill a gap in the ongoing research of the object detection of solar plants from geoinformatics perspective with a focus on the description of overall workflow from data collection to discussion of results, and examination of feasibility using freely accessible tools and satellite imagery collections.

Main research question

- To what degree of accuracy can we use freely available remote sensing images and the cloud processing platform, e.g., GEE to detect large-scale solar plants?

Research sub-questions:

- To what extent is, in terms of area size and installed capacity, feasible to get information for large-scale solar plants from Sentinel images?
- What are the advantages and disadvantages of using binary supervised classification in order to detect large-scale solar plants?
- Which machine learning algorithm available in GEE is the most suitable for the detection of large-scale solar plants? Is GEE capable enough to do this job?

The structure of the thesis is as follows: firstly, the theory about researched subjects will be described. Secondly, the data processing and the overall approach for using GEE and its supervised classification along with commented script will be explained and written. The last part will consist of results commentary and discussion/conclusion part with a focus on answering the primal research questions. This project was conducted as a master thesis for Geoinformatics study program. The overall workflow for a thesis procedure ordered by the time frame from the thesis proposal to the final writing is presented in Figure 1.

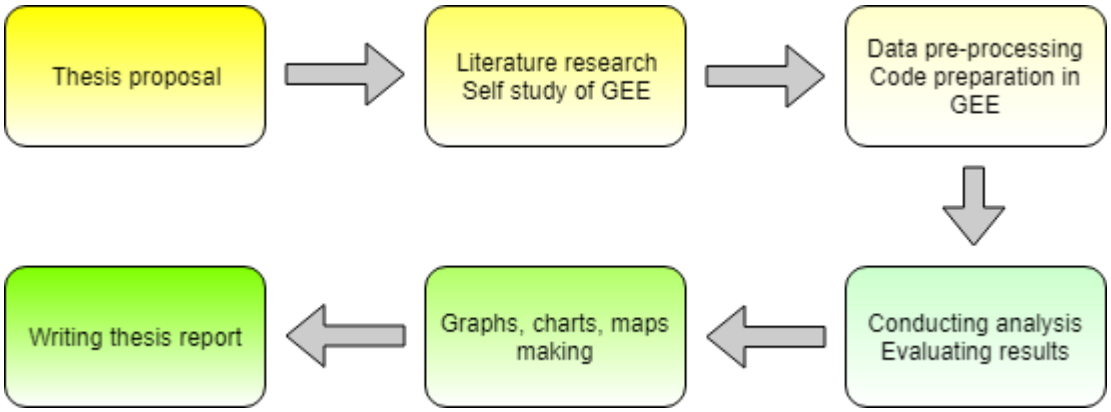


Figure 1. The workflow of the thesis procedure

1.2 Existing research on large-scale solar plants detection using RS

There is still limited research on detection of large-scale solar plants or solar panels in the scientific literature. Many of the existing researches on object detection focus solely on high spatial resolution imagery within the small study area (incomparable with the scope of the study area of this project). Malof et al. (2015) conducted research to develop a computer algorithm to detect rooftop PV installations using high resolution (0.3 m) orthophoto imagery. Their algorithm shows exciting results with 94% detection rate and only 4 out of 57 PV installations falsely detected but mapping only one specific part of the city, Lemoore, CA.

Ishii et al. (2017) investigate the trained CNN using Landsat 8 satellite images (30 m spatial resolution) for the detection of solar power plants. Similarly, as the approach of this thesis project, they use the existing solar plant's dataset to annotate the results. Associated research from Imamoglu et al. (2017) uses CNN (FB-net model to a baseline CNN) and Landsat 8 imagery, this time with a weakly supervised feedback and the satisfactory results for pixel-wise detection tasks. Also, few blog posts present the research on a similar topic. Padarian (2018) small project consists of detection of solar rooftop installations within a city neighborhood. The research uses GEE to demarcate solar panels, but the CNN model is run elsewhere. Many of these researches have the common idea of particularly exploring the feasibility of their approach farther development and potential future research.

2 BACKGROUND AND THEORY

2.1 Google Earth Engine

There are several tools for enabling the processing of large-scale geospatial data – GeoMesa, GeoSpark, or Hadoop. Unfortunately, in many cases these tools require extensive technical expertise, knowledge of handling advanced data acquisition and storage and working with complex data and file formats, not to mention machine allocations or CPUs and GPUs complications among many (Gorelick et al. 2017). These difficulties discouraged many scientists from solving their potential research questions. The remote sensing research community, therefore, appreciated the initiation of the new platform – Google Earth Engine. Before GEE creation, it was theoretically described the needfulness for large-scale cloud computing product with a high-performance system to leverage the power and to get user-defined temporal and spatial high-quality remote sensing images (Li et al. 2019).

GEE is a cloud-based, freely accessible and available Google product and processing imagery platform for facilitating remote sensing data extraction and processing without the need for extensive data downloading and immersed data processing. Data are mostly gathered from NASA`s Earth Observing Satellites (mainly MODIS and LANDSAT) and ESA`s Sentinel satellites (Wasson et al. 2018)(Li et al. 2019). GEE is, for the time being of writing this master thesis, still a new platform used for scientific analysis and geographical visualization of large-scale (petabyte) spatial datasets including historical images (Liu et al. 2018). The present data archive consists not only from remote sensing data but is also complemented by vector data sets about demographic, social, weather, climate, and digital elevation models features (Mutanga and Kumar 2019). GEE takes profit not only from its free access but also for the growing user base, which is especially useful when exchanging information, code samples, and applicable approach (Healey et al. 2018).

Google Earth Engine framework and platform overview

GEE along with its catalog are reachable through Code Editor and GEE Explorer (both web-based platforms) after approved user registration. The GEE

Explorer is used to analyze and view satellite images with a set of tools and Code Editor is used to fully customizable analysis using JavaScript and Python programming language. Several spatial or mathematical operations can be executed on satellite imagery using Code Editor (Liss, Howland, and Levy 2017).

GEE has a relatively user-friendly front-end solution for interactive data and algorithms exploration and processing (Mutanga and Kumar 2019). The access is provided after the approval of the development team when submitting the form of potential usage and the background of the user. It is expected that the GEE will provide a solution for not only scientists and researchers, but also for hobbyist and remote sensing enthusiasts. The diagram of all available components of GEE is displayed in Figure 2.

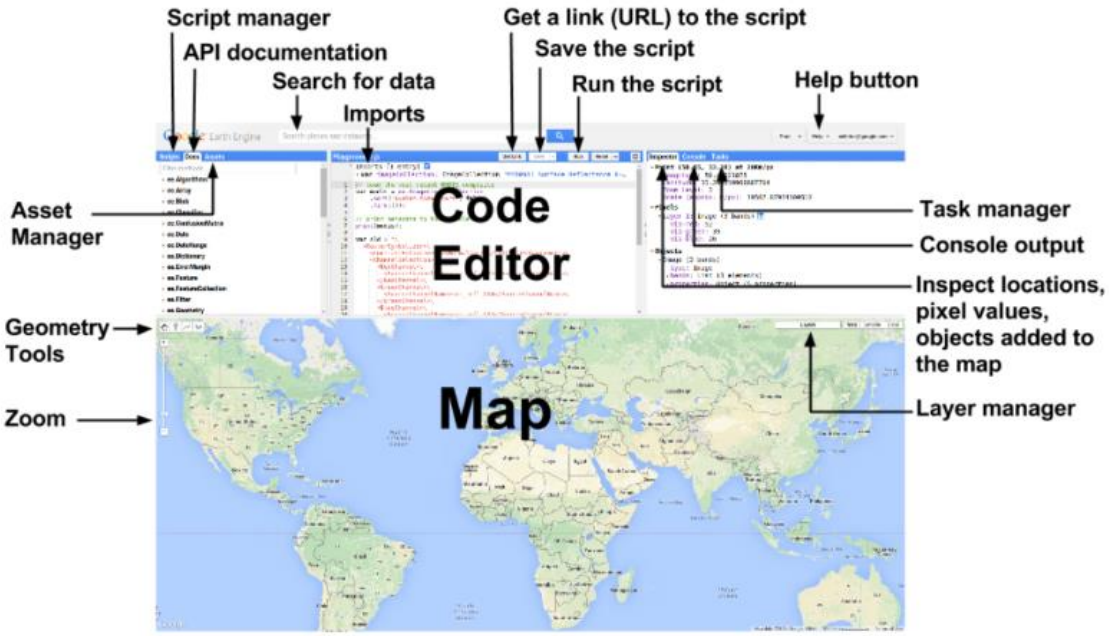


Figure 2. Components of the Earth Engine Code Editor ("API Tutorials | Google Earth Engine API | Google Developers" n.d.)

For many previous years, earth observation datasets were labor-intensive, required long-term acquisition for application and subsequent analysis with a dependent on geometric and radiometric correction due to systematic sensor errors and atmospheric impact (Hansen and Loveland 2012). Furthermore, the preparation of large-scale remote sensing data was inefficient as a result of principally for individuals since for example the Landsat data were accessed from the United States

Geological Survey's archive after vast inspecting, filtering and downloading these big data (Woodcock et al. 2008).

2.2 Satellite imagery

Remote sensing and satellite imagery if used correctly can provide a beneficial point of view to explore the surface of the earth (Liss, Howland, and Levy 2017). As contrasted to the standard imagery, the remote sensing imagery significantly differs with the satellite images containing extra spectral information beyond the scope of the RGB spectrum. Spectral bands (infrared), which are not visible to the human eye, are used in remote sensing (Ishii et al. 2017). These bands work as different channels of the image and can help to get 'for eyes' hidden information. Satellite imagery is considered to be a useful data source more and more and is getting accessible easily every year more often. Amongst the application and usage of satellite imagery belong the assisting in urban planning, navigation, disaster planning, and recovery and getting data about large objects and energy. The intersection of object detection and energy are the main content of this thesis project.

Sentinel-2

Copernicus, formerly recognized as Global Monitoring for Environment and Security (GMES) is a Europe based monitoring Earth system which collects data from various sources (Earth observation satellites, ground stations, airborne sensors, etc.). Copernicus operates these data and provides them to end users (public authorities and policymakers), mainly related to environmental and security concerns. The services of this system contain six particular areas: atmosphere, climate change, land, marine, along with security as mentioned above and emergency management (Pasco et al. 2016).

Sentinel's missions as a part of Copernicus Space Component are used in emergency management for mapping flooded areas, burn scars, earthquakes or landslides; nonetheless, their resolution is too coarse for urban area`s disaster and damage mapping (Pasco et al. 2016).

Sentinel-2 imagery consists of two sensors collecting data between 443 nm and 2190 nm (thirteen bands with different spatial resolution) at revisiting time of five days.

There are also other three bands (the channels of the atmospheric correction – B1, B9, and B10) with 60 m spatial resolution (Bomans et al. 2018).

Sentinel-2 is a high-resolution, wide swath and multi-spectral satellite image European mission consisting of twin satellites flying at the same orbit with a frequency of revisit five days. Sentinel-2 has optical instrument equipment which samples 13 spectral bands. Sentinel-2 mission is a follow up to SPOT (Satellite for observation of Earth) and Landsat missions, compared to them, however, stands out with better spatial resolution and a greater wide-swath (Figure 3) (ESA 2019b).

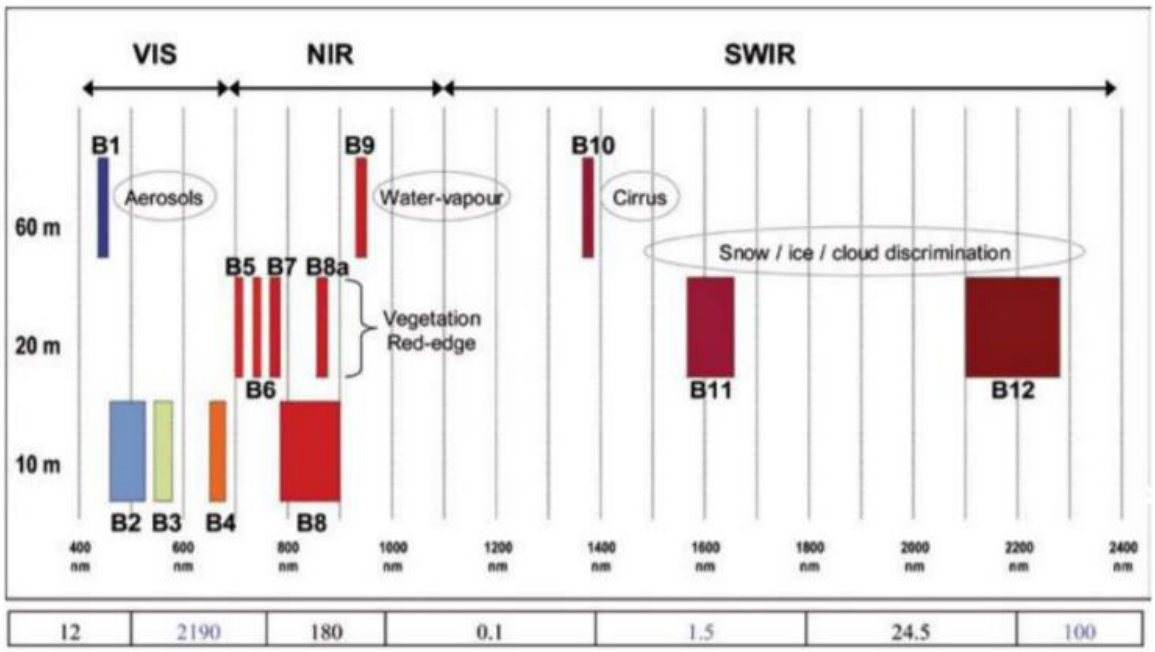


Figure 3. Overview of Sentinel-2 spectral bands (ESA 2019b)

Sentinel-1

The mission of Sentinel-1 is even older (April 2014) than Sentinel-2 mission and consists of two satellites orbiting polar operating C-band synthetic aperture radar remote imaging. The radar imaging is particularly useful because the image acquisition is disregarding weather. Sentinel-1 has C-band performing in four exclusionary modes with distinctive resolution (5 m being the highest resolution). It supplies short revisit times, dual polarization capability, and fast product delivery.

Synthetic Aperture Radar (SAR) operates at wavelengths not disrupted by lack of illumination or cloud cover (ESA 2019a). Sentinel-1 is monitoring the Earth surface

in 4 modes: Strip Map Mode (SM), Interferometric Wide Swath Mode (IW), Extra Wide Swath Mode (EW) and Wave Mode (WM) displayed in Figure 4.

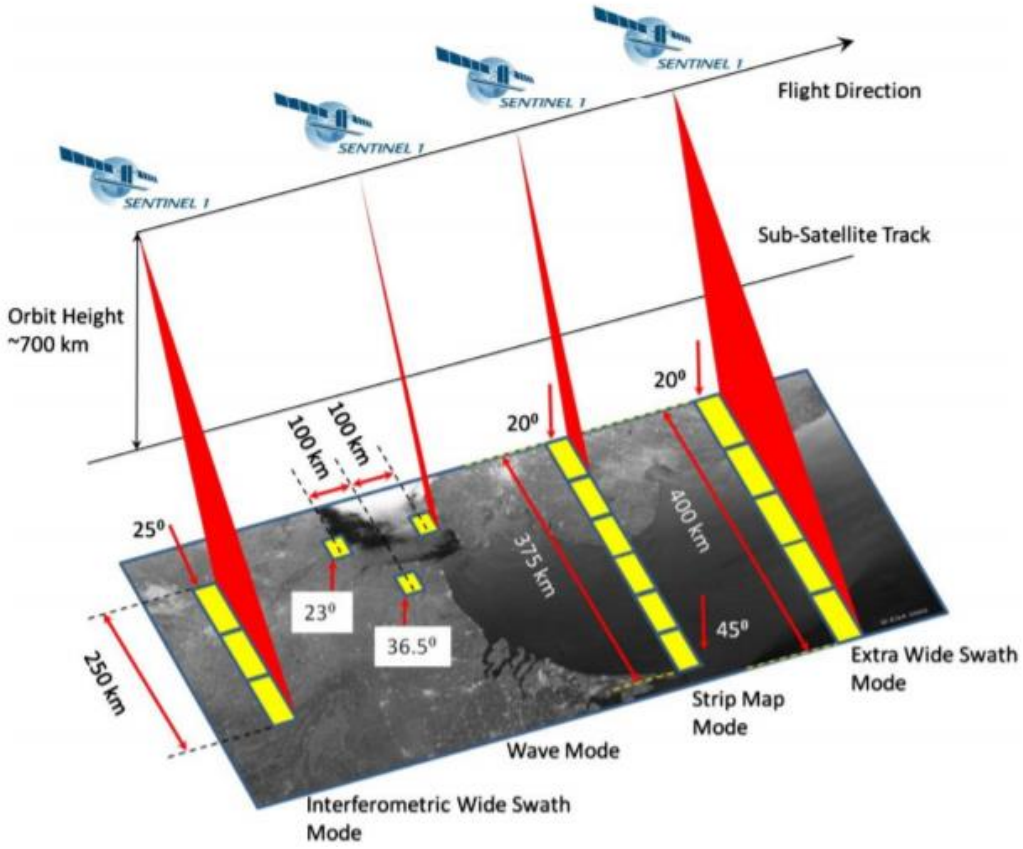


Figure 4. Different Sentinel-1 SAR modes (ESA 2019a)

2.3 Object detection in remote sensing

Object detection is a part of remote sensing science area in last years and represented a growing challenge as an open research topic being a significant challenge for decades (mostly within the automatic building extraction) (Chen, Li, and Li 2018). The goal is to determine whether certain satellite image contains one or more objects of a given class along with the geographical position of these features. The term "object," in this specific meaning, refers mostly to a human-made feature like vehicle, ship, building or in the context of this thesis solar panels or solar plant. This object should locate in the independent background environment and have sharp boundaries (Cheng and Han 2016). Successful object detection from remote sensing

can play an essential role in updating and creating geospatial databases, environmental or geological monitoring, or even in agriculture.

Object detection applicable in remote sensing suffers from several challenges such as massive diversity of the visual appearance of the object, background clutter, shadow, and illumination, to name a few. This topic has been studied within the geospatial field since the 1980s. The earliest satellite images, including Landsat images, lacked the spatial resolution needed for the detection of separate natural or human-made features. The studies targeted only to extracting region properties from these images. After certain advances in remote sensing, the very high-resolution satellites were introduced (Quickbird, IKONOS, and SPOT-5) and the pictures from these satellites can provide more detailed textural and spatial information to the point where human-made objects appeared observable and recognizable and could be identified. That opened new challenges linked with automatic detection of objects with geospatial information (Cheng and Han 2016).

There are several object categories for object detection in remote sensing, which were described and summed up by Cheng and Han (2016). These categories are template matching-based object detection methods, knowledge-based object detection methods, object-based image analysis detection methods, and machine learning-based object detection methods. In recent years, the scientific field of object detection from imagery has been dominated by CNN (Ishii et al. 2017).

2.4 Solar energy and solar plants overview

The increasing human consumption end to end with the future exhaustibility of the traditional resources of energy has caused growing energy issues for the world economy in the last years. To avoid energy deficiencies, a gradual transition to another alternative source of energy (water, wind, biomass, etc.) is one of the few conceivable solutions (Navratilova 2013). Being considered the future of the world's global success, solar power plants are on the rise as a sustainable, significant energy source (SpaceKnow 2016). Solar energy is branched into solar PV, concentrating solar power and solar heating.

Solar energy is considered socio-economical beneficial and environmentally friendly with a few negative impacts as cultivable land loss, site size of large solar installations, and impacts on the ecosystem. The positive effects as the emission of greenhouse and toxic gases, reclamation of degraded land and many others, however, prevail (Tsoutsos, Frantzeskaki, and Gekas 2005; Phillips 2013). From the beginning of the 21st century, solar plants were growing rapidly due to the efforts of research and development, particularly in Germany, China, Spain, Australia, and the United States (Behar 2018).

There were nearly 74 GW solar energy installations in 2016 but is anticipated by the current projection that the total capacity will exceed 1300 GW by 2025 and will pass wind energy total installed capacity around 2022 (Nyheim and Bruhn 2017).

Large-scale solar plants in the world and Denmark

There is no direct definition what can be considered as a large-scale solar plant (often also used as a solar park, solar power station or with a term 'photovoltaic' instead of 'solar') in terms of capacity and area size. Piyatadsananon (2016) writes about large-scale solar plant following: "system designed for the supply of merchant power into the electricity grid at the utility level. It has been promoted to invest this kind of renewal energy in several countries over the world." The biggest solar installation in the world, at the time of writing this thesis report, exceeds 1 GW of installed capacity and the largest by area size is big around 25 km^2. There are two kinds of large-scale solar plants: ground-mounted solar plants, which are the main objective of this study and are located in broad areas of agriculture, private or public areas. The other type is rooftop solar installations which are located on the rooftops of large commercial buildings or factories (Piyatadsananon 2016).

Nowadays, Denmark is one of the leaders on large-scale solar plants which are directly linked to the district heating systems. The number of solar plants in Denmark expanded rapidly over the last years (Furbo et al. 2018). The goal of reaching 200 MW of installed capacity distributed by PV was reached in 2012 (although the set year was 2020). Denmark has the largest solar plant in the whole Scandinavia – Solar park Vandel in Jutland with a capacity of 75 MW and is providing sufficient electricity to power around 20 000 Danish homes in a year (EuropeanEnergy.dk 2015).

Table 1 demonstrates the generation and capacity of solar power in Denmark from 2012 to 2017. However, also the values for small solar plants and rooftop installations are contained in the table, not only values for large-scale solar plants.

Table 1. Solar PV electricity capacity and generation values for Denmark for period 2012 – 2017 (source: IRENA (2018, Renewable Energy Statistics 2018)

Country/area	Technology	Indicator	2012	2013	2014	2015	2016	2017
Denmark	Solar photovoltaic	Electricity capacity (MW)	402	571	607	782	851	906
		Electricity generation (GWh)	104	518	596	604	744	

2.5 Solar plants data of Denmark

The Danish Energy Agency provided the initial dataset of all registered solar plants (from the small rooftop solar panels to large-scale PV installations) as the author of this thesis worked as an intern for 4 months before thesis period there (on the project unrelated to the topic of this thesis project, except focus on the solar plants). This dataset was preprocessed, some parts of the same installations were grouped, and for the purpose of the project, the original dataset was filtered out to leave only solar plants with an installed capacity larger than 200 kW. Even though there is no direct definition and threshold for what can be considered as 'large-scale' installation, it is not in the interest of this project to detect solar plants with a smaller installed capacity. The source for the geographical location of the solar plants from this dataset was based on the geocoded web scrapping from DAWA (Denmark Addresses Web API). The coordinates for solar plants from this register are in many cases not accurate as the registered address of solar installation might be linked to an owner's address (in many cases of large-scale solar plants it is the adjacent farm); thus the real geolocation of that solar plants is biased. This dataset contains point features, and together with the location, also the installed capacity is provided.

The map showing the locations of the biggest solar installations in Denmark (valid for September 2018) is presented in Figure 5. These data will be used mainly for a visual check, drawing input training data and validation data for accuracy assessment of the classification. From the map (Figure 5) where the location is demonstrated, it can be observed the largest solar installations in Denmark are spread

all over the country with largest one being Solar Park Vandel in Jutland with the installed capacity of 75 MW.

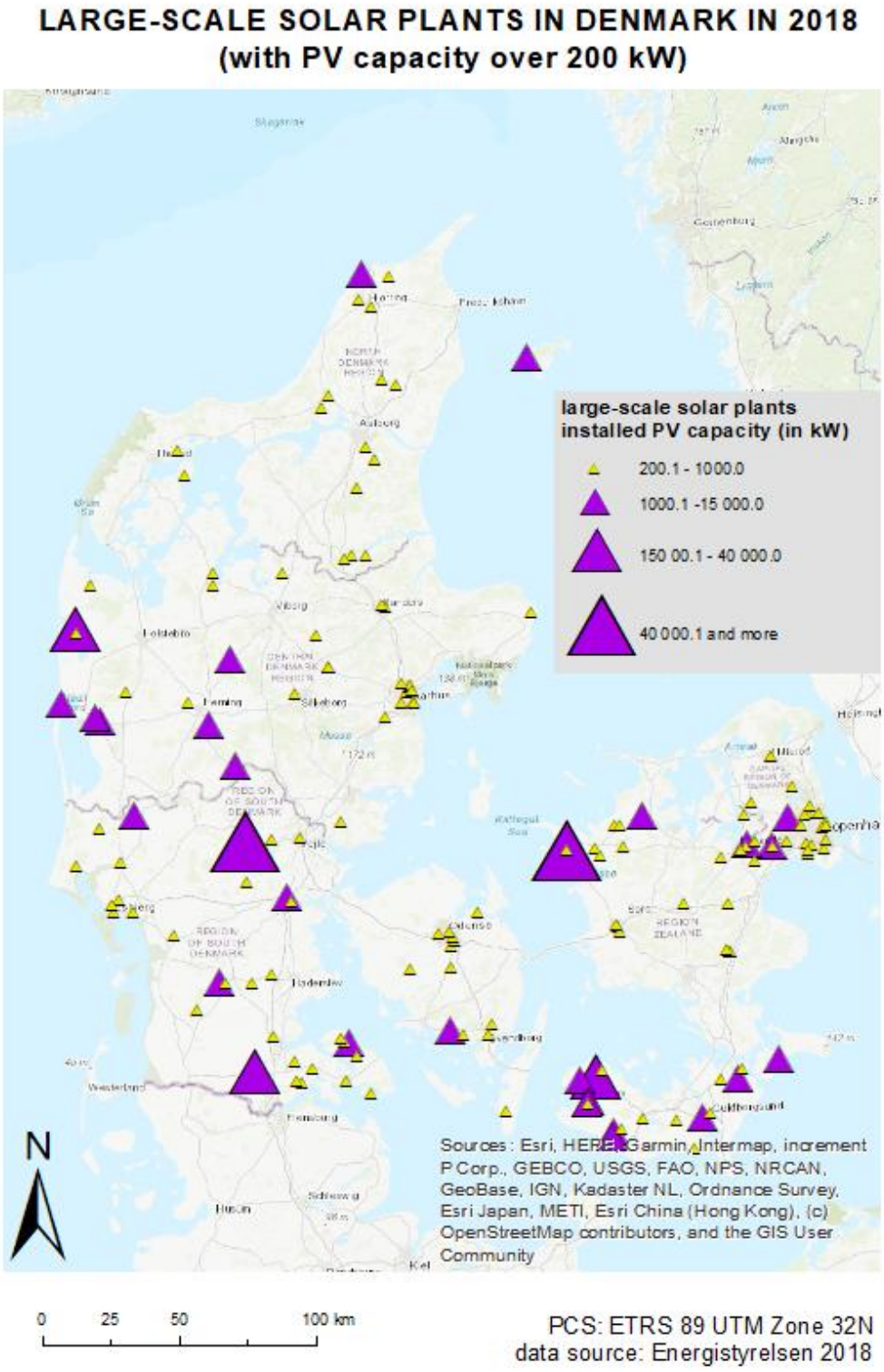


Figure 5. Locations of large-scale PV plants in Denmark with a capacity of 200 kW and more

2.6 Machine learning algorithms

In supervised classification in remote sensing imagery, the selection of suitable machine learning algorithm is crucial. Machine learning in GIS is used not only in supporting classifications in remote sensing but is being currently used to solve many issues, including 3D visualization, geospatial economic modeling, and all different variety of problems. Usage of machine learning in remote sensing dates back to the late 1990s. These algorithms learn the pattern based on the training data, and with this example, can generalize the whole study area (Farda 2017). When this algorithm is trained, sequentially can be applied to the entire image in the researched region of interest, and the result classification is retrieved. The selection of machine learning algorithms used in this thesis project was evaluated by literature research (which are the most effective and used algorithms in remote sensing imagery supervised classification) and by the GEE API (application programming interface) availability. GEE provides in the time of this thesis processing 10 machine learning algorithms (CART, Fast Naïve Bayes, GMO Max Entropy, Multi-Class Perceptron, Random Forest, Margin Support Vector Machines (SVM), Voting SVM, Pegasos, IKPamir, Winnow), which can be divided into several groups (Farda 2017). In the following part, the machine learning algorithms which were used in this thesis project will be described theoretically. The accuracy assessment results from these different classifiers will be compared in the 4.1 section of this thesis paper.

Support Vector Machines

SVM algorithm targets entirely on the training data samples that are located closest in the classification space to the optimal boundary between the feature classes (Maxwell, Warner, and Fang 2018) and can be used both for classification and regression. These samples are support vectors. The objective of this classification is to detect the suitable boundary which expands the segregation between these vectors. This classifier is binary, thus detects a single boundary between surveyed classes. SVM classifiers are more frequently used in remote sensing applications due to their capability to generalize with a high level of accuracy even with a small number of training samples (Mountrakis, Im, and Ogole 2011). GEE contains two decision procedures – ‘Voting’ and ‘Margin’ whereas Voting decision procedure is the default.

Classification and Regression Tree

Classification and Regression Tree (CART) algorithm is along with Random Forest belonging to the logic-based machine learning algorithms applicable in GEE (Farda 2017). CART is a binary tree-structured algorithm and has a non-parametric approach for recognition of patterns (Bittencourt and Clarke 2004). CART affects the identification and construction of decision trees using pretrained data (Thamilselvana and Sathiaseelan 2015). The advantage of this classifier is that it can handle the vast amount of processed data.

Random Forest

Random Forest classifier (belongs under ensemble methods for classification) uses a broad number of decision trees to overrun the deficiency of a single decision tree (Maxwell, Warner, and Fang 2018). The final class of every pixel of this classification is assigned based on the majority 'vote' of the entire set of trees. Usually, approximately two-thirds of training data samples are used to train trees, and the remaining third is used for internal cross-validation for estimating how well the model itself performed. This classifier is also regularly used in remote sensing applications for detection of land cover, greenery and is especially advisable to use this classifier when working with multispectral imagery (Belgiu and Drăgu 2016).

Naive Bayes

Naive Bayes classifier is an efficient and simple method of classification for remote sensing image supervised classification. It is based on probability theory and is one of the six models of Bayesian networks. So as Random Forest classifier is especially capable tool when working with multispectral satellite imagery classification (Solares and Sanz 2007). This classifier is based on the assumption that the effect of each attribute on a researched class is separated from another attribute. Naive Bayes classification algorithm works with conditional independence assumption and then has the least possible chance of error. Based on that, the algorithm examines every attribute with an identical effect on the attribute of its class (Yang et al. 2018). In GEE, Fast Naive Bayes classifier is available and falls under the statistical learning algorithm class.

2.7 Accuracy assessment theory

The results of classification can be visualized in the form of the map when post-processing the results, but only accuracy assessment can quantify the legitimacy of results or if the classification results competent or incompetent. The validation of classification provides quality of data information both for producers and users of the analysis. These results can also guide which of the classifier is the most suitable. The most accepted method of expressing accuracy of classification (in this case supervised) is confusion matrix (error matrix) and various descriptive measures acquired from this confusion matrix to provide the accuracy information for each of the classified classes (Topaloğlu, Sertel, and Musaoğlu 2016).

Confusion matrix and overall accuracy

Confusion matrix compares relations among validation ground truth data and the corresponding classification results class by class. This matrix is used to calculate all adjacent values and coefficient. Overall accuracy value is calculated by dividing the number of accurately classified pixels with an overall number of reference pixels (Equation 1).

$$Overall\ accuracy = \frac{Total\ number\ of\ correctly\ classified\ px}{Total\ number\ of\ pixels} x\ 100$$

Equation 1. Overall accuracy equation

Producer's and user's accuracy

In addition to overall accuracy, there are producer's and user's accuracy calculated similarly and help to evaluate the accuracy of classification. Producer's accuracy expresses the quality of the mapped classification and is calculated by division of the overall number of pixels which were classified accurately by the total number of all referenced "known" (ground truth) pixels. The value of producer's accuracy, therefore, means how strong the reference data are classified or can be interpreted as a probability that a randomly chosen point in the region of interest has the same property value as on the final classification map. On the other hand, the user's accuracy expresses the quality of the classification of training dataset and is calculated by the division of the accurately classified pixels of every category by the total number of pixels which were classified for this category. It shows how well the classification

represents the chance that the pixels classified into a specific category represents this category in the real world based on the "ground truth" data. User's accuracy can also be interpreted as the probability for each class that a randomly chosen point on the result classification map has the same property as in the real world. Other indicators are omission and commission error, which are the values conversed from producer's, respectively user's accuracy and calculated as the derived values from 100 % from their accuracies (Hasmadi 2005). The formulas explaining these indicators are in Equation 2.

$$Producer's\ accuracy\ (\%) = 100\ \% - omision\ error\ (\%)$$

$$User's\ accuracy\ (\%) = 100\ \% - commision\ error\ (\%)$$

Equation 2. Producer's and user's accuracy equations

Kappa coefficient

Kappa coefficient is a value to measure the original training pixels to the validation data (ground truth data). The coefficient differs between -1 and +1 values, where the negative value displays the low accuracy, values around 0 no correlation at all and positive values the high accuracy (Taufik, Ahmad, and Khairuddin 2017). The mathematical equation for kappa is presented below (Equation 3) where *n* is the total number of pixels *p* is the number of classes and Σx_{ii} signs the absolute number of elements in the confusion matrix.

$$K = \frac{n\sum_{i=1}^{p} x_{ii} - \sum_{i=1}^{p} x_{io}x_{oi}}{n^2 - \sum_{i=1}^{p} x_{io}x_{oi}}$$

Equation 3. Kappa coefficient equation (Taufik, Ahmad, and Khairuddin 2017).

3 DATA COLLECTION AND PROCESSING

3.1 Process summary

The whole process of the following classification using GEE and summarized in sections from 3.2 to 3.6 is presented in Figure 6.

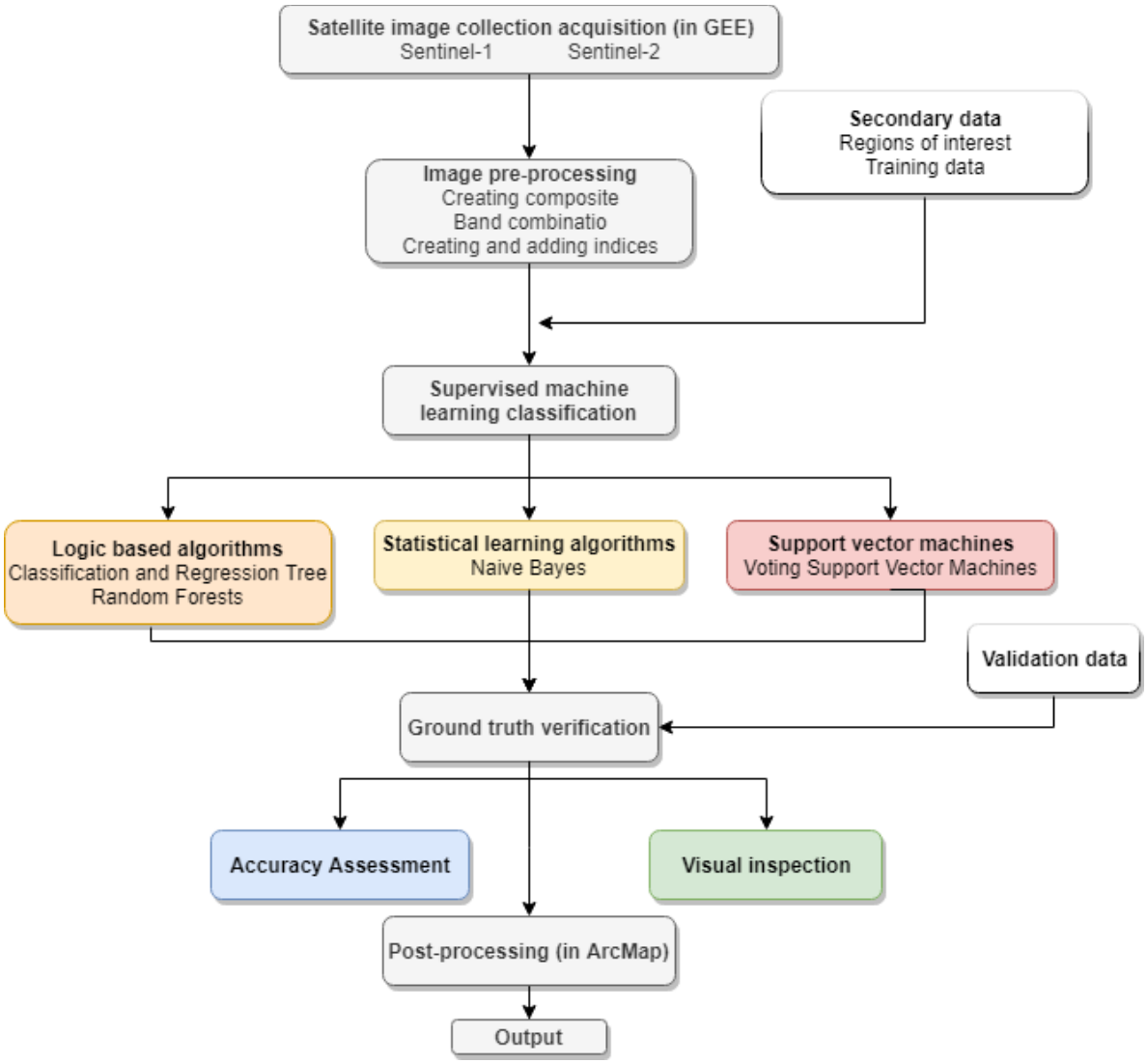


Figure 6. The flow chart of the process of the classification

3.2 Training input data for image classification

To use the function to classify satellite images collections into the suitable analysis information, creation of sample training data is required. It is possible to upload preprocessed data or create point or polygon features directly in GEE or combine both methods. In this project, several potential options were carried out to compare, and the optimal procedure is written down. The choice of the training data is the predominant factor for successful classification. As object-based classification depends solely on two classes, feature collection with not the only property of solar plants occurrence, but also non-solar occurrence sampling containing as various land classes (urban areas, water bodies, crops, etc.) in the surrounding regions around trained samples of solar plants. The class property had to be in a binary numerical form so for the complete analysis and throughout the used datasets, the property solar was used with values 1 = solar plants training areas or 0 = areas without solar plants or even small solar panels appearance. In certain aspects, the supervised classification used in this analysis can be viewed both as a one class classifier and binary classifier at the same time as even though the analysis is executed for two classes, the goal is to extract the information just from one (solar). In Mack and Roscher (2014) one-class classification, the classifier requires reference data exclusively for a class of interest. This classification (without using contrarily training data – nonsolar), however, can be biased and is not, at the time of this thesis writing, supported by GEE.

Training data for solar plants were drawn using geometry tools in GEE on top of Sentinel-2 composite image for the first half of 2018 using true colors composite. These training patches were extracted as feature collection and drawn on the largest (by installed capacity) large-scale solar plants in the original dataset also considering their geographical distribution in the investigated region (Denmark), so the patches of solar plants from different parts of Denmark were used. The critical aspect is the number of training pixels to avoid overfitting and underfitting in the consecutive analysis. The classification works by exploring the pixel values in predefined training sites to gather statistical signature for each of the training sites. To choose the solar plants where the training patches will be drawn upon, the idea that for the future reproducibility of the research is always easier to find information and data about the

most spacious large-scale power plants and then to detect the smaller ones. Hence, the training patches were drawn on the largest one (by capacity) concerning their spatial distribution. The samples of the training sites are seen in Figure 7 (in green) where training patches are drawn upon solar plants near Vandel in Southern Denmark region (left) and the solar plant near Gislinge in Zeland region (right). Even though the recognition of ground truth data is crucial and difficult in supervised classification in remote sensing, solar plant detection by user view is possible by significant visual features even without using high-resolution imagery. The patches were drawn and later trained on the inner part of arrays of solar plants to distinguished clearly from the surrounding areas and not to include non-solar areas. The red point symbols in Figure 7 denote the original latitude and longitude attributes from the original datasets.

Figure 7. Example of training sites made upon Sentinel-2 image

The same level of importance is given for defining areas with non-solar attribute. Although it might seem that more non-solar patches are created, the more precise the final classification in terms of quantifying accuracy will be, the limitation of GEE must be considered conjointly. With geometry tools in GEE are limited by their function, for creating training patches for solar class = 0 ArcMap software was used. These areas were created as an intersection between two polygon circles with a radius of 4,000 m of a smaller one and 8,000 m for a larger one where the center of the circle is the centroid of the solar plant. The example of these areas is in Figure 8. These proportions were given that these polygon features would not include the area of solar plants. The visual analysis of these areas must have been conducted as it was necessary to not include any single solar plants (even the larger rooftop solar arrays) in these polygons. Based on a later execution of code for analysis, later the polygons were cut in half due to a computation capacity exceeding of GEE.

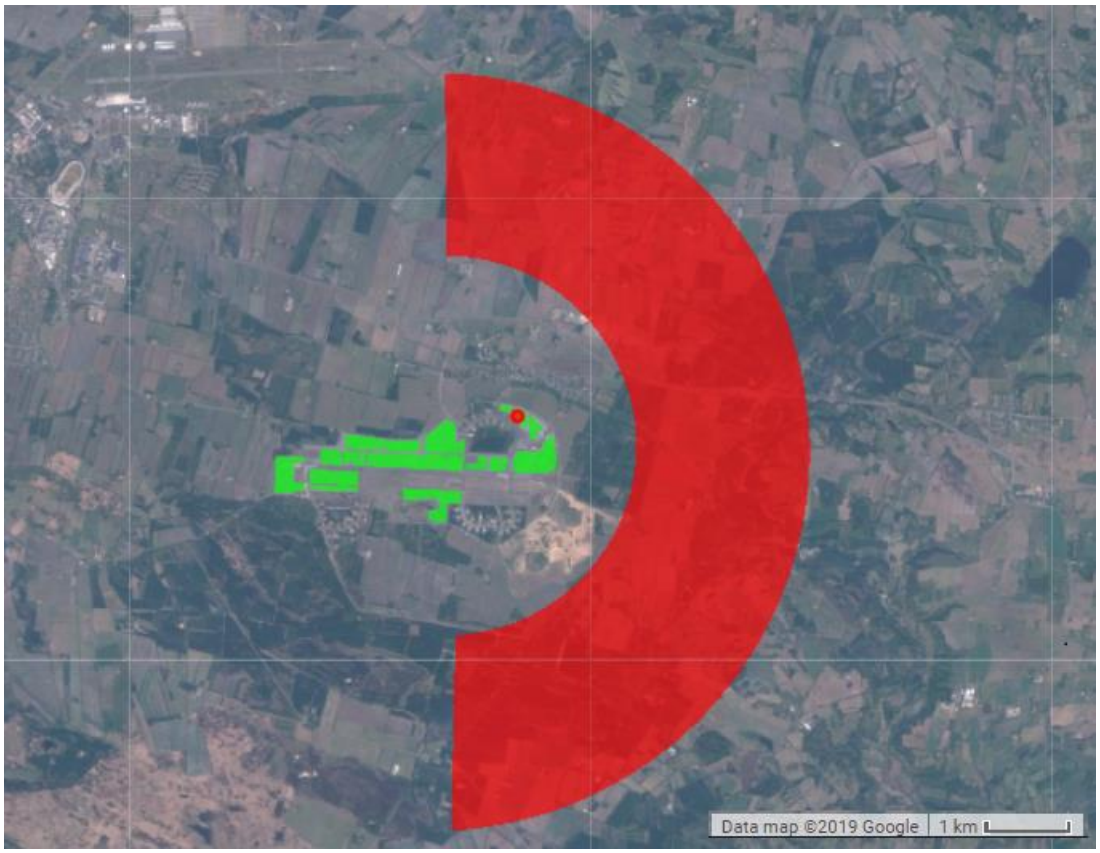


Figure 8. Non-solar areas (red color) training sample in the area of Vandel solar plant

3.3 Pre-processing of satellite imagery data

As all the computational operation are carried out on cloud in GEE, and the satellite imagery is provided besides, the pre-processing must be executed there. For the forthcoming classification, both imageries from Sentinel-1 and Sentinel-2 missions were processed as for the challenging classification with a specific two classes input the best results will be needed. To classify the imagery with higher accuracy, the function that calculates the normalized difference vegetation index (NDVI) that quantifies vegetation and normalized difference water index (NDWI) that indicates the moisture content of the land surface were written (Code 1). Code 1 and all following code samples are written using JavaScript programming language. These indices must be renamed as seen in the complete code in Appendix 1. Adding these indices as additional bands to classification support the classification with higher overall accuracy and better kappa coefficient as conducted in research by Taufik, Ahmad, and Khairuddin (2017) on Landsat 8 satellite data.

```
//Creating function
that creates NDVI and NDWI indices (later used for image collection)
var addNDVIBands = function(image) {
var NDVI = image.addBands(image.normalizedDifference(['B8', 'B4']));
var NDWI = NDVI.addBands(NDVI.normalizedDifference(['B3', 'B8']));
return NDWI.addBands(NDWI.metadata('system:time_start'));
};
```

Code 1. NDVI and NDWI adding function written in JavaScript

Concerning the size of the study region and incomparable areas of large-scale solar plants, the highest spatial resolution imagery was needed, and thus the data from Sentinel-2 of 10 m and 20 m (dependent on the bands) were used. The image collection of Sentinel-2 top of atmosphere data was loaded (Code 2), filtered by date to get the median image from the whole collection, filtered by region of interest, using cloud filtering function and generating NDVI and NDWI indices using the function written in Code 1.

```
// Loading Sentinel-2 TOA reflectance data.
// Map the function over half a year of data and take the median.
var sentinel2Collection = ee.ImageCollection('COPERNICUS/S2')
                    .filterDate('2018-01-01', '2018-06-30')
                    // Pre-filter to get less cloudy granules.
                    .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))
                    .filterBounds(region2)
                    .map(addNDVIBands);
```

Code 2. Loading of Sentinel 2 data in GEE (JavaScript GEE script)

With the image collection loaded, the true color composite from Sentinel-2 imagery using bands 4, 3 and 2 is adequate and pertinent for the training data sampling described in 3.2 section.

The similar approach was used when importing data from Sentinel-1 before stacking the layers for classification from these two collections. Because unlike Sentinel-2, Sentinel-1 Ground Range Detected (GRD) scene data are radar data and requires specially designed algorithms to get imagery orthorectified and calibrated.

```
// Loading the Sentinel-1 image collection
var sentinel1Collection = ee.ImageCollection('COPERNICUS/S1_GRD')
                    .filterDate('2018-01-01', '2018-06-30')
                    .filterBounds(region2);

// Filtering by metadata properties.
var metaSentinel1 = sentinel1Collection
  .filter(ee.Filter.listContains('transmitterReceiverPolarisation', 'VV'
))
```

```
    .filter(ee.Filter.listContains('transmitterReceiverPolarisation', 'VH'
))
  .filter(ee.Filter.eq('instrumentMode', 'IW'));

// Filtering to get images from different look angles.
var vhAscending = metaSentinel1.filter(ee.Filter.eq('orbitProperties_pas
s', 'ASCENDING'));
var vhDescending = metaSentinel1.filter(ee.Filter.eq('orbitProperties_pa
ss', 'DESCENDING'));

// Create a composite from means at different polarizations and look ang
les.
var composite = ee.Image.cat([
  vhAscending.select('VH').mean(),
  ee.ImageCollection(vhAscending.select('VV').merge(vhDescending.select(
'VV'))).mean(),
  vhDescending.select('VH').mean()
]).focal_median();
```

Code 3. Loading of Sentinel 1 radar data in GEE (with the help of "Sentinel-1 Algorithms | Google Earth Engine API | Google Developers" n.d.)

Sentinel-1 data is gathered up with a different resolution, configuration, and band combination, and by that reason, it is necessary to filter these data before working them further on. These data have different geometry and strong speckle of images. In Code 3 above the pre-processing steps in GEE are written down including loading of data, filtering to get images both with VV and VH dual polarization and to get image collection in interfered wide swath mode (Sentinel-1 Algorithms, GEE). After initiating these steps, the Sentinel-1 composite can be created, used for display and most importantly, for layer stacking with Sentinel-2 data to be used as a final composite for classification.

3.4 Classification

The supervised classification in remote sensing has with the ability to monitor and identify objects from a satellite image. Supervised classification is a process where the computer is trained to detect a distinct object or specific land cover area on its unique reflected energy at distinctive wavelengths (Liss, Howland, and Levy 2017) and the process of performing it in GEE environment will be described further on.

The next and significant part is to model and run a classifier when Sentinel-1 and Sentinel-2 data are preprocessed. This procedure described subsequently comments on using random forest classifier with 20 trees, but the same approach is

used analyzing different machine learning classifiers (described in 2.6). To begin with, the training data was loaded to GEE (both with solar areas and non-solar areas), and Sentinel-1 and Sentinel-2 data were merged. Consequently, the bands for the classification were selected from both data sources. To generate a training based on a training data where 'solar' is the main parameter, each polygon from training data is overlaid by the within the Sentinel stacked data and extracting the data values for each sample of the training dataset. Once the data values for each band and indices are captured, then the classification model could be run. Furthermore, the classifier model was applied for the input parameters and could be visualized for image classification visualization or exported in raster format (GeoTIFF). This process in GEE is written in Code 4.

```
/**************************
Random Forest Classification
**************************/
// Loading preprocessed training data with a feature property solar = 0 or 1
var trainingData = solar_non_solar_ZealandSJ;

var color_solar = trainingData
  .filter(ee.Filter.neq('solar', null))
  .reduceToImage({
    properties: ['solar'],
    reducer: ee.Reducer.first()
  });
// Merging of Sentinel 2 and 1 preprocced data
var finalComposite = sentinel2Composite.addBands(composite);

// Training sample data
// Selecting of bands suitable for classification from Sentinel 2 and 1

var bands2 = ['B2', 'B3', 'B4', 'B5', 'B6', 'B7', 'B8', 'B11', 'B12', 'NDVI', 'NDWI'];
var bands1 = ['VV', 'VH'];

var bands = bands2.concat(bands1); //Merging of bands from Sentinel 2 and Sentinel 1

var input = finalComposite.select(bands);

var classifierTraining = input.select(bands)
      .sampleRegions({
        collection: trainingData,
        properties: ['solar'],
        scale: 20
      });

// Instantiating Random Forest (or different) classifier and training it:
var classifier = ee.Classifier.randomForest(10).train({
```

```
  features: classifierTraining,
  classProperty: 'solar',
  inputProperties: bands
});

//Classifying the image filtered by region
var classified = input.select(bands).classify(classifier).clip(region2);

Export.image.toDrive({
  image: classified,
  description: 'Denmark_classification_raster',
  region: region1.geometry().bounds(),
  scale: 20
});
```

Code 4. Machine learning supervised classification model code in GEE

3.5 Accuracy assessment in GEE

Regardless of classified visualization, the important factor which decides about the success of classification is accuracy assessment, which is a quantification of results. This can also be done using GEE code editor.

Accuracy assessment measures the consilience and compare the result of the classification with other data, optimally ground truth data from a different source or comparing with high-resolution imagery. In the thesis, two different accuracy assessments were made and used.

The first variation of accuracy assessment (Code 5) works by partitioning the set of known data values (training data bot for solar plant areas and non-solar areas) to training and testing set, in this case in the ratio 70:30. The classification training set is split randomly in by randomColumn function and the results, therefore, might be slightly different every time the code is run. The classifier must remain the same as in the previous part (in this case random forest classifier), and the training set which was made by splitting from the original training set is used to train a model. The testing set (30% split of original training data) serves as validation and is used to deliver confusion matrix. This confusion matrix is made comparing the ‘solar’ property with the newly created ‘classification’ property. From the confusion matrix, GEE is able to calculate overall accuracy, producer’s accuracy (omission error), consumer’s/user’s accuracy (commission error) and kappa – coefficient of agreement and order value. These results can be printed into GEE console or saved as CSV file into Google Drive

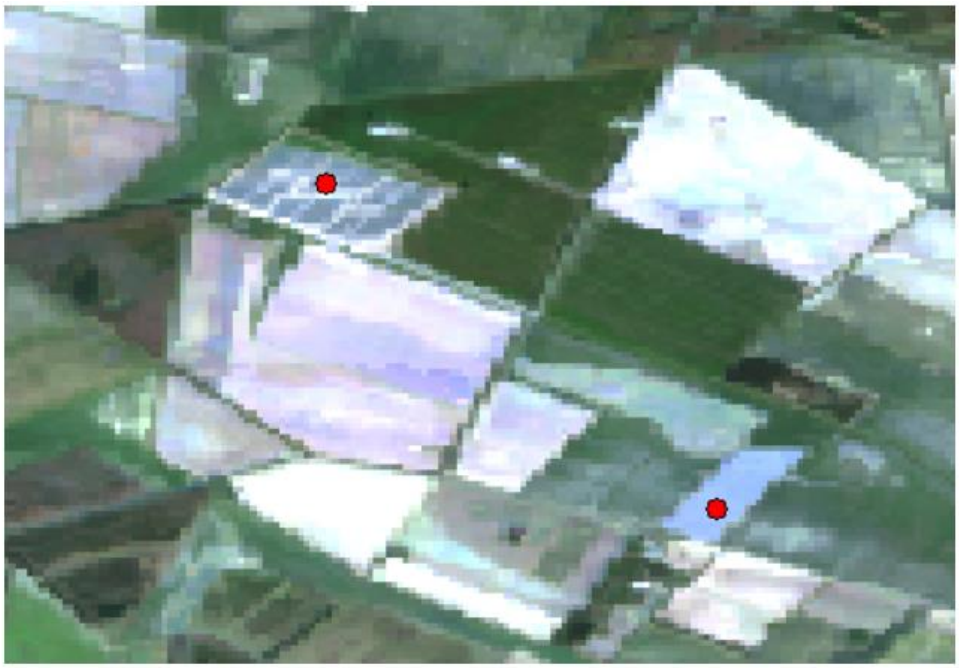
which is a more appropriate method, in this case, concerning the training set data size and the area of the region of interest, printing into the console of large volume computation caused computation timed out error.

```
/******************
Accuracy assessment (1)
*******************/

var testTraining = classifierTraining.randomColumn();
var trainedSet = testTraining
.filter(ee.Filter.lessThan('random', 0.7));
var testingSet = testTraining
.filter(ee.Filter.greaterThanOrEquals('random', 0.7));

// Training the classifier with the trainedSet:
var trained = ee.Classifier.randomForest(10).train({
  features: trainedSet,
  classProperty: 'solar',
  inputProperties: bands
});

// classifying the testingSet and get a confusion matrix
var confusionMatrix = ee.ConfusionMatrix(testingSet.classify(trained)
      .errorMatrix({
        actual: 'solar',
        predicted: 'classification'
      }));

var confMat = ee.Feature(null, {matrix: confusionMatrix.array()});
var overAccu = ee.Feature(null, {matrix: confusionMatrix.accuracy()});
var prodAccu = ee.Feature(null, {matrix: confusionMatrix.producersAccura
cy()});
var consAccu = ee.Feature(null, {matrix: confusionMatrix.consumersAccura
cy()});
var kappa = ee.Feature(null, {matrix: confusionMatrix.kappa()});

// Printing the results to console in GEE (sample)
print('Confusion Matrix:', confMat);
print('Overal Accuracy:', overAccu);

// Exporting the results in the CSV format to Google Drive (sample)
Export.table.toDrive({
  collection: ee.FeatureCollection(confMat),
  description: 'confMat_Zealand',
  fileFormat: 'CSV'
});
```

Code 5. Accuracy assessment 1

The second method and the more suitable is to compare the classification results with ground truth data of original dataset. This dataset was filtered to contain only data for large-scale solar plants. Thus the threshold for installed capacity was made to be 500 kW. The solar plants with smaller installed capacity are not considered “large-

scale" for this accuracy assessment, though the definition of the exact capacity threshold does not exist. Moreover, they hardly could be detected by satellite imagery of spatial resolution of Sentinel-1 and Sentinel-2 datasets. This data was valid for the second half of 2018, and the number of these large-scale power plants in the region of interest – Denmark was 47. Since these data might lack the precise location, the manual validation was made using ArcMap software, and their position was changed upon the same Sentinel-2 image collection median as was used for the training data set in GEE. These 47 points were manually validated and, in some cases, relocated into the potential centroid of the large-scale solar plant area. This dataset was merged with the dataset of 42 randomly distributed points in the region of interest made by Create Random Points function. For each of these merged points from final validation datasets property value, 1 or 0 was added for 'solar' class. The example for these validation data points is seen in Figure 9 (the spatial resolution of this Sentinel-2 satellite composite median was scaled for 30 m when exporting to raster format to be used in ArcMap environment due to the computation limitation of GEE). These two sample validation points are located in the centers of the solar plants, and then the property value of 1 is assigned in the attribute table.



	idrift	id_geo	instkw	anlvej	anlpostnr	x	y	source	sola
	14-May-18	99531	6800	Langelinie 3C	4970	11.30235	54.724523	DAWA	1
	30-Dec-13	87642	3200	Langs Vejlevej 17	4970	11.300813	54.705563	DAWA	1

Figure 9. Sample of validation data points for accuracy assessment

The final validation dataset was taken and from shapefile transferred into Fusion Table format to be fully cooperative in GEE. The accuracy assessment process (Code 6) is simpler and more straightforward but works according to the same principle as accuracy assessment from Code 5 by the fact that compares the classification results with existing ground truth data, additionally manually validated.

```
/******************
Accuracy assessment (2)
********************/

var trainingTesting2 = 92points_for_validation;

var validation = classified.sampleRegions({
  collection: trainingTesting2,
  properties: ['solar'],
  scale: 20
});

var testAccuracy = validation.errorMatrix('solar', 'classification');

var confMatrix = ee.Feature(null, {matrix: testAccuracy.array()});
var overAccuracy = ee.Feature(null, {matrix: testAccuracy.accuracy()});

var prodAccracy = ee.Feature(null, {matrix: testAccuracy.producersAccura
cy()});
var consAccuracy = ee.Feature(null, {matrix: testAccuracy.consumersAccur
acy()});
var kappa2 = ee.Feature(null, {matrix: testAccuracy.kappa()});

// Printing the results to console in GEE (sample)
print('Confusion Matrix:', confMatrix);
print('Overal Accuracy:', overAccuracy);

// Exporting the results in the CSV format to Google Drive (sample)
Export.table.toDrive({
  collection: ee.FeatureCollection(overAccuracy),
  description: 'overalAccuracy_Denmark_assesment2',
  fileFormat: 'CSV'
});
```

Code 6. Accuracy assessment 2

3.6 Post-processing of the classified maps

GEE capabilities are extensive, but as for the post-processing of the raster, the desktop GIS software (both licensed and open source) can help process the result data and add additional value. GEE has a function to vectorize raster and use it for another analysis straight in the API, but unlike the supervised classification, pre-processing of

a vast collection of satellite imagery. Converting raster (of a bigger area, which the whole country of Denmark is) to vector is a time-consuming process in GEE. However, transferring the results of a classification in a raster format to Google Drive in a georeferenced TIFF format is straightforward and quick. Then the data can be generalized using filter to straighten the visual perception of the result and make congruous patches suitable for vectorizing. This vectorized dataset can be used as a polygon feature shapefile with the information about the location of the large-scale solar plant as contrasted to original, incorrect point datasets. Also, the point centroid of the polygon feature can be created, leading to better longitude and latitude information of the solar plant even when needed to be working with point dataset.

The overall workflow of this processing method is transferable to many available GIS software or tools. The described process can be worked on using capabilities of modern programming languages with the ability to work with spatial data such as Python, and R. The post-processing of the classification result data in this thesis project was utilized in ArcMap, the main component of ArcGIS programs for geospatial processing especially for the possibility to process large raster file quickly and for the experience, the author of this thesis has with this software.

The process leading to consequently usable format with valuable metadata and information can be seen in Figure 10, where the process is demonstrated using ModelBuilder application within ArcMap. The input raster is created using GEE code editor when performing supervised classification; the part of the code can be seen on the lines 35-40 in Code 4. This raster in georeferenced TIFF format (GeoTIFF) is firstly clipped using known polygon area (in the case of this study Denmark) with raster processing clip function considering the tile form of the exported raster from GEE. The majority filter is applied to filtered raster applying four neighbors to use and majority replacement threshold. Consequently, this filtered raster is vectorized to polygon shapefile using Raster to Polygon function with simplifying polygon option. There the selection of solar = 1 attributes can be made to distinguish only solar areas. Due to the fact that some of these solar area polygons are composed of few closely detached patches, the Aggregate Polygon function is used with aggregation distance of 1 km. Lastly, these already aggregated polygons are completed of geometry attribute, specifically the area of the solar plant polygon in square kilometers and

potentially transferred to point shapefile in particular when the point dataset is required and the location of this point feature is the centroid of previously created solar plant polygon feature. Even though the above-described process was created in ArcGIS environment, the same approach can be applied to many varieties of existing GIS processing tools with these functions which might just have different naming. When having the processed vector dataset is advisable to overlay it with primal raster classification result and compare them visually to detect any possible contradiction arisen during the processing.

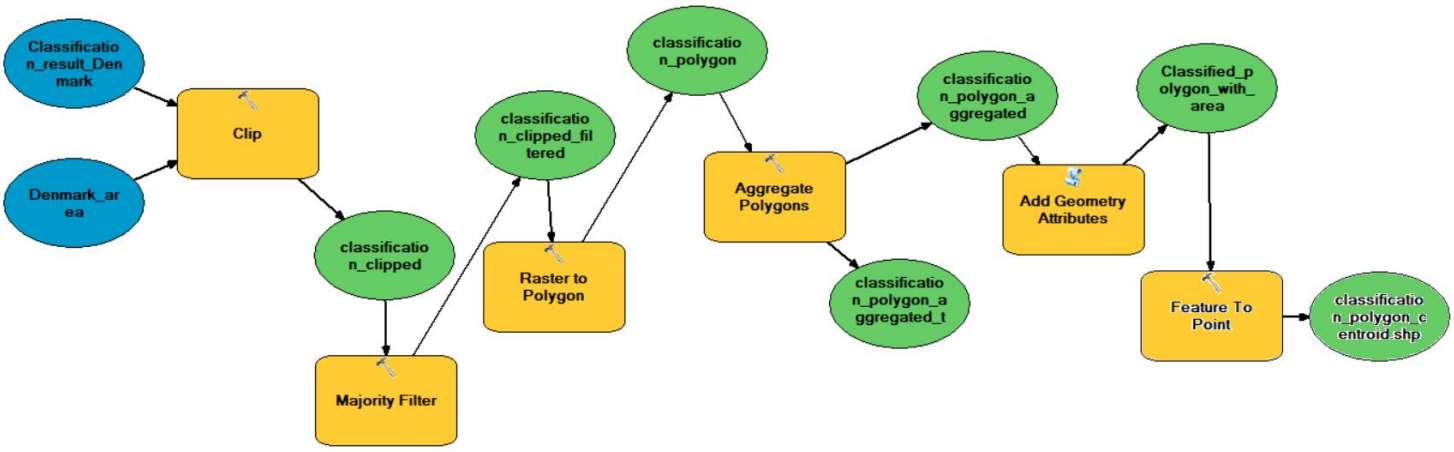


Figure 10. Model (in ArcMap SW) for post-processing of raster classification

4 RESULTS

4.1 Accuracy assessment results of different models

The classification of data was conducted in GEE, and together with the classification map, the accuracy assessment results for various machine learning classifiers were calculated. The same input training data were put into classifications with drawn patches in the five samples of large-scale solar plants (based on their capacity and geographical location) and neighboring areas without solar plants occurrence in the proximity of solar plants. Following results are conducted from accuracy assessment 2 (Code 6) using ground truth manually inspected validation data points. This accuracy assessment reflects the external validity acceptably. Assessment from Code 5 is re-substitution accuracy and must be viewed with attention as these classifiers can emulate the training set explicitly. The accuracy assessment from Code 5 was even so conducted, but the results (overall accuracy in almost all cases exceeding 99 %) were not decisive in this type of supervised classification. The following results are subject of Code 5 solely.

These results and accuracy assessment methods use independent validating datasets. As these models were chosen based on the literature research on the supervised classification with multi-classes, not the similar study was found where the goal is to detect two (respectively one when an only class of solar plants occurrence is demanded. Regarding the results captured in Table 2 and visualized in the bar chart in Figure 11, the figures for particular accuracies and kappa coefficient are not prominent.

Table 2. Accuracy assessment results of different classifiers

	overall accuracy	producer's accuracy	consumer's accuracy	kappa
CART	77.98%	52.94%	70.73%	0.545
Random Forest (20)	78.90%	54.90%	71.60%	0.564
SVM	78.90%	52.94%	71.60%	0.564
Naive Bayes	68.81%	37.25%	63.64%	0.351

The lowest overall results are in classification using Naive Bayes classifier. This machine learning classifier based on the numerical results is least likely to be efficient for two class supervised classification, and even the visual inspection of the classifier map shows wrong assumption to classify pixels which are not seen in the training phase. The numerical results for the other three classifiers are approximately the same. CART and Random Forest (with 20 trees) classifiers belong to the same class of logical machine learning classifiers. Random Forest has slightly better performance, especially at the kappa coefficient, which is the most critical indicator of the classification effectiveness. Results for SVM classifier are surprisingly identical as for the Random Forest. The number of trees in Random Forest classifier is an important variable, but for this comparison, the number of trees = 20 was chosen as it is widely used in satellite imagery classification based on the literature review. It must be however noted, that the validation data are in a limited number (90) and there are only two classes in the classification. These results are showing essential information about classification and its classifiers but to be fully interpretable, the visual inspection analysis must be provided. This is crucial, particularly with the fact that the validation data was point dataset, and some of these points were classified in the opposite class even though most of the solar plant might have been labeled right.

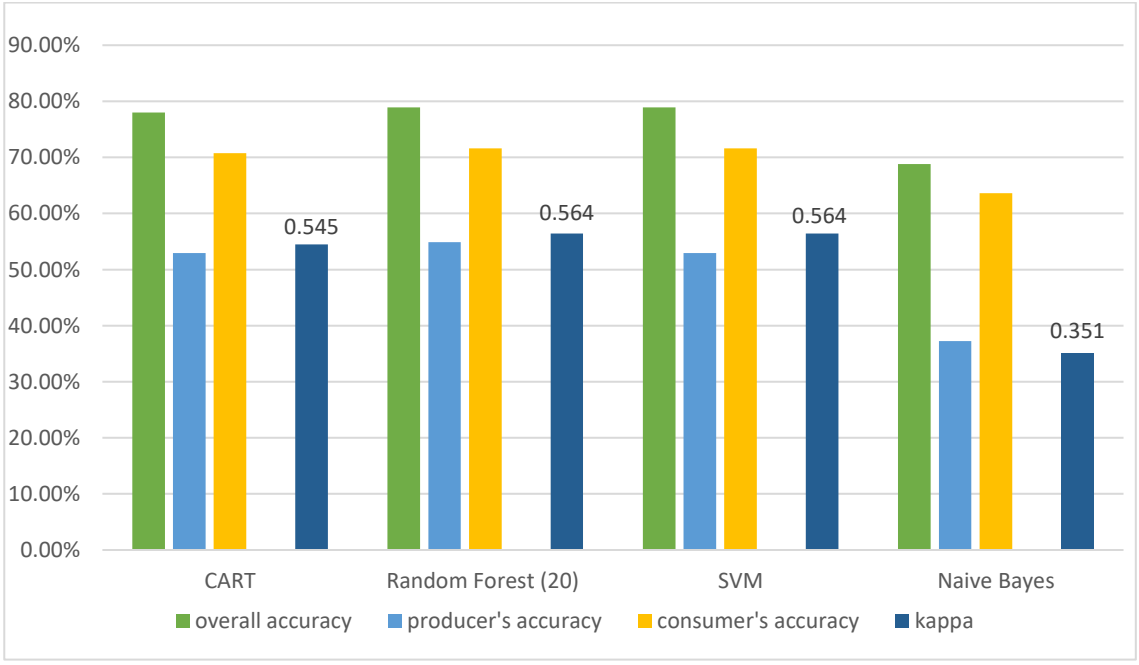


Figure 11. Models comparison for different accuracy assessments

The results are for accuracies are not extremely high must the specification of the two-class classification must be considered. Kappa coefficient interprets how an outcome classification is significantly better than the map generated by random where values between 0.40 and 0.80 represent the moderate agreement (Lee et al. 2016). This must, however, be confronted with another interpretation of results (visual analysis).

Computation of confusion matrix and consecutive numerical results were carried out in GEE using export to table function as printing the results into the console exceeded the limited time available. Nevertheless, the computation into the CSV tables on Google Drive consumed a significant amount of time, different for each classifier. The average time for GEE to compute this classification using CART classifier was around 12 minutes (all results including kappa coefficient were calculated and available always in the timeframe of a minute), Naive Bayes 14 minutes and Random Forest with 20 trees 17 minutes. The time needed for calculating the results for SVM classifier exceeds 1 hour. It must be considered, however, the extent of the area of interest and the quantum of training and validation data.

4.2 Visual inspection of the classification results

As the values of accuracy assessment can tell certain information, it is suggested to visually inspect the results by comparing the results of the classification with the real ground truth data or with a base satellite imagery in true colors. Even though that SVM machine learning classifier is widely used in the supervised classification of remote sensing images, especially when classifying urban green areas (Kranjčić et al. 2019), for the purpose of two class classification and solar plants detection is impracticable. The first look at the classification for whole Denmark shows approximately equal distribution of solar and non-solar areas, which is incongruousness. However, this is seen only on the results classification using SVM classifier. Considering previously mentioned, the longest calculation time also the least accurate visual classification is seen in the case of a solar plant near Sundby in Zealand. The classification in Figure 12, where the green circle point indicates the middle location of the solar plant and the green color of classification shows the solar areas. While the area of the solar plant (below in Figure 15) is classified correctly, the surrounding fields and urban areas are also classified as solar areas; however, in the

researched area of Figure 12, there is only one solar plant of a larger scale. Similar cases are seen throughout the classification, and while the numerical accuracy assessment results might seem high, it may be caused by the coincidence factor (only two class in classification and under 100 validation points dataset).

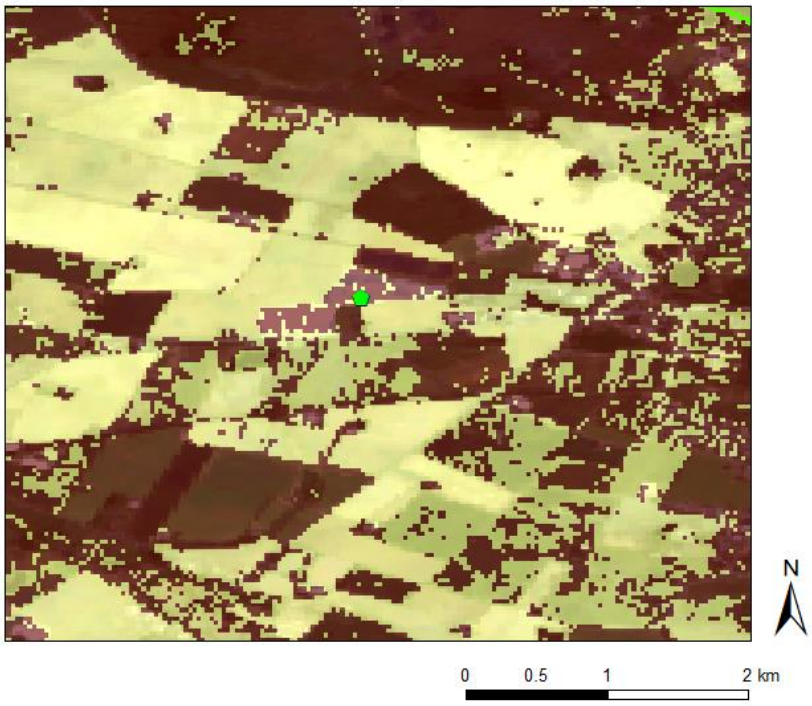


Figure 12. SVM Sundby classification visual inspection

In the following lines, the comparison of visual classification for CART, Random Forest of 10 trees and Naive Bayes classifiers will be compared and commented on the example of three selected large-scale solar plants. The solar plant in Figure 13 is located at the south of Region of Southern Denmark near the town Tinglev close to the borders with Germany. The solar plant was put into operation in 2015 and with its installed capacity around 20 MW was the fifth largest solar power plant by capacity in Denmark in 2018. At first glance, there is a significant observable difference between the results using Naive Bayes classifier and the other two. Despite capturing some of the solar panels into the right class, the overall observed results are poor. The much-improved situation is observable when confronting the classification results with Sentinel-2 true composite image of CART and Random Forest with ten trees classifiers. In brief, the results look very similar, but there are a more explicit

structure, fewer gaps, and falsely classified solar areas in Random Forest classification. Additionally, the black mark indicates the verification point from validation dataset.

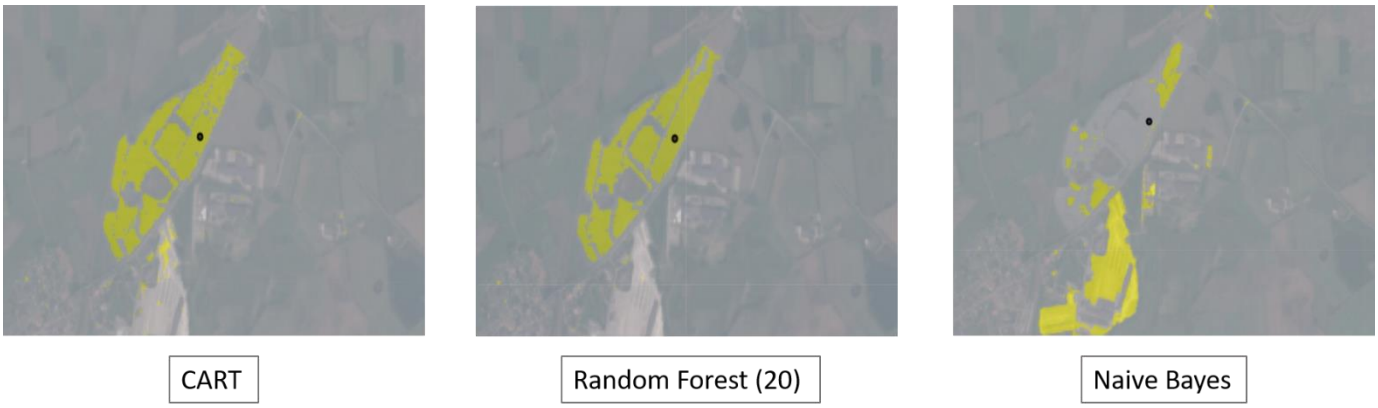


Figure 13. Classification visual results comparison shown in the case of the solar plant near Tingley

Subsequently, the two adjacent installations which form one smaller large-scale solar power plant near the city Nakskov on the west of Lolland island with a capacity of 2.8 MW (northern installation) and 2.4 MW (southern) constructed in 2014, respectively in late 2013. Even though this solar plant is much smaller by area than the one near Tingley from Figure 13, the visual results for CART and Random Forest classifier are overall satisfactory. In Figure 14, Random Forest results exceed the CART classification because of the lack of random false positive pixels outside the solar areas. Despite the positive classification of the southern part of the Nakskov solar plant, the northern part is classified wrongly, and this indicated how small the threshold is for such a classification of solar plants. The high-resolution imagery shows that the solar panels arrays from northern part have the different size or tilt, but the overall visual classification is not of good quality, because there are a lot of false positive solar samples in the rectangular cut of the region of interest.

Figure 14. Classification visual results comparison shown in the case of the solar plant near Nakskov

The last inter-model comparison is on the example of the chosen solar plant is in Figure 15. From the first instance, the classification results from Naive Bayes classifier seem the most fitting in terms of covering the area of the ground truth soar plant, but the results from this small cut of the area also indicate the high rank of false positive samples of solar pixels classified.

Additionally, in this case, the results from CART classifiers exceeds the Random Forest classifier, but the visual record is not flawless. This solar installation lies in the east of Lolland island and was put into operation in early 2018. Its capacity was 7.0 MW, and that can be considered as the lower half of the large-scale solar plants of what can be treated as a 'large-scaled.' The imperfections in the CART and Random Forest classifiers results, however, can be improved when post-processing these results (as suggested using the model in Figure 10). The validation point in the Random Forest results pinpoints the wrong classification in the validation phase and stress the uncertainty with using point dataset for validating supervised classification.

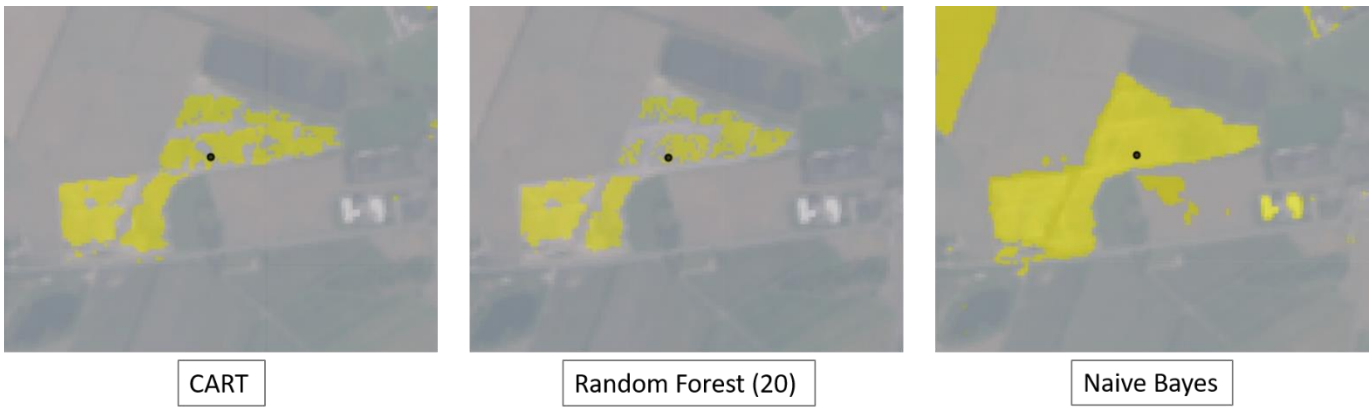


Figure 15. Classification visual results comparison shown in the case of the solar plant near Sundby

On the above commented and interpreted visual results, it is apparent, that the CART and Random Forest classifiers are superior to Naive Bayes and SVM classifiers when interpreting the results visually. When comparing CART and Random Forest, the second mentioned exceeds the CART for the lesser number of false positive solar patches and the overall smoothness of the solar plant areas with a higher level of smoothness of the classified solar areas.

4.3 Results comparison of different number of trees for RF classifier

Based on the results from 4.1 and 4.2, Random Forest classifier was chosen to be the most appropriate and suitable machine learning classifier of GEE for this thesis research. The default number of decision trees to create per class in GEE algorithm function is 1, but when running the classifier, the number of trees was chosen 20 as this setting is widely seen in related GEE supervised classification. When Random Forest classifier was demonstrated to over proceed the other classifiers, the optimal number of trees was researched. The code was run with the same training data and same preprocessed imagery only with a different number of trees (default 1, 5, 10, 20, 25). The results are shown in Table 3. Even though the classification was made for a different number of trees, the results showed that if 5 or fewer trees are used, the results remained the same and the same for using 20 trees and more (even the attempt of using 50 trees presented the same results as for 20). The kappa coefficient, which might be considered as the most important significant factor for supervised classification results, do not differ distinctively. Although for the consequential practice, the 20 trees used in Random Forest classifier will be considered as the optimal result in this thesis project and will be commented and analyzed further on.

Table 3. Comparison of a different number of trees in Random Forest classifier.

	overall accuracy	producer's accuracy	consumer's accuracy	kappa
Random Forest (5 trees and less)	77.06%	50.98%	69.88%	0.525
Random Forest (10)	77.98%	52.94%	70.73%	0.545
Random Forest (20 and more)	78.90%	54.90%	71.60%	0.564

4.4 Results comparison without using Sentinel 1 data

It was assumed that using the Sentinel-1 radar imagery data would improve the overall classification result, but for the verification, the classification was done using bands from Sentinel 2 imagery (B2, B3, B4, B5, B6, B7, B8, B11, B12 plus NDVI, and NDWI indices) solely.

Table 4. Classification accuracy assessment values for classification with and without Sentinel 1 data

	overall accuracy	producer's accuracy	consumer's accuracy	kappa
Random Forest (20) S1 + S2	75.23%	47.06%	68.24%	0.486
Random Forest (20) - only S2	78.90%	54.90%	71.60%	0.564

As can be seen from Table 4, there is a more significant difference between values from the classification using additional Sentinel-1 data and not using the same classifier with the same number of trees. This observation states that the usage of Sentinel-1 data to classification improve the results from the accuracy assessment perspective not significantly, but a certain level of enhancement can be found. Moreover, the visual inspection on the example of the solar plant near Nakskov is in Figure 16. The particular improvement using a stacked layer from both Sentinel-1 and Sentinel-2 imagery is seen on the right picture, particularly in the northern part. Besides, without Sentinel-1 data addition (on the left picture) there are more false positive pixel patches in the cut. These patches might be disposed of in the post-processing phase using the majority filter, but it can be stated that using Sentinel-2 together with Sentinel-1 data is preferred and shows better results visually and accurately.

Figure 16. Visual inspection of classification results using Sentinel 1 data

4.5 Results using less amount of training sites

The previous analysis and results commentary were made on the results using training data of 7 selected large-scale solar plants chosen by their capacity (the largest were chosen) and geographical location throughout the study area (that the training data would be collected not only from one part of Denmark). Despite that, the lower number of training solar data samples was used to compare how feasible the results will be in terms of accuracy assessment related to the original training data input. In this research, the non-solar area remained the same – 4 half circles in the cut hole in the middle with an area of approximately 75 $km^{2,}$ as shown in Figure 8. The classifications were conductor from 7 training large-scale solar areas to 3 where the results started to be insufficient.

Table 5. Comparison of different number of solar plants training samples

	overall accuracy	producer's accuracy	consumer's accuracy	kappa
7 solar plants training sites	78.90%	54.90%	71.60%	0.564
6 solar plants training sites	76.15%	49.02%	69.05%	0.525
5 solar plants training sites	75.23%	47.06%	68.24%	0.486
4 solar plants training sites	74.31%	45.10%	67.44%	0.466
3 solar plants training sites	64.22%	23.53%	59.79%	0.247

When lowering the number of training solar data, the classification results are getting subtly worse up to the four training solar areas, however when using only three large-scale solar plants for training data input, the numbers for accuracy measurement are significantly worse (Table 5). Additionally, the bar chart in Figure 17 shows the steep of kappa coefficient between using three and four solar training patches.

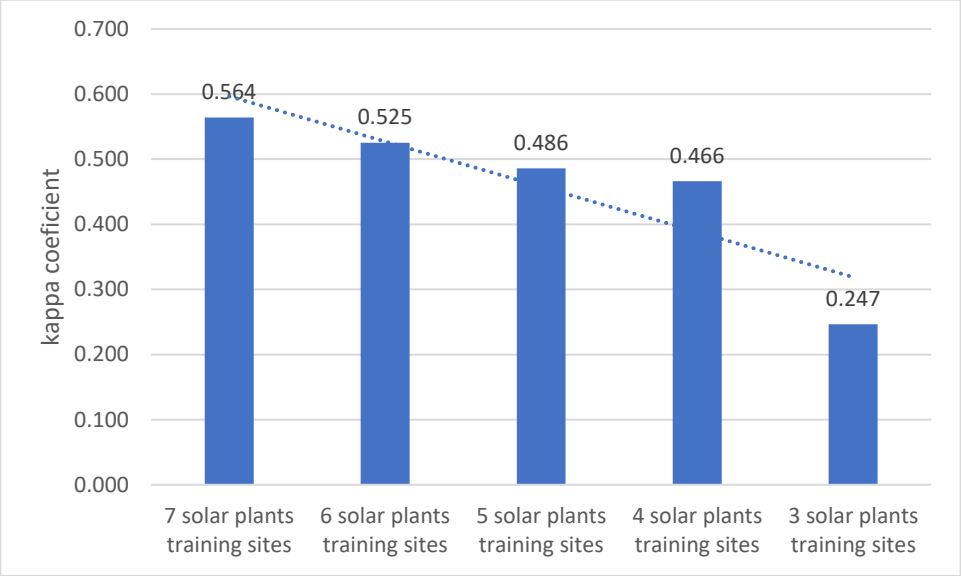


Figure 17. Comparison of kappa coefficient for different number of solar plants training samples

4.6 Use case on Schleswig-Holstein

This analysis was so far conducted on the region of interest with a previously known data (even though with certain limitation, and low spatial precise accuracy). One of the questions asked when performing the classification was whether it is possible and feasible to extend the region of interest and have the detection of solar plants information available also for the regions with the unknown data. Based on the geographical proximity, the northernmost state of Germany – Schleswig-Holstein was chosen. The analysis was conducted the same as for Denmark using GEE, only the region of interest was expanded. The input training solar and non-solar data remained the same. The results classification layer was exported out of GEE environment and processed using the workflow and model from Figure 10.

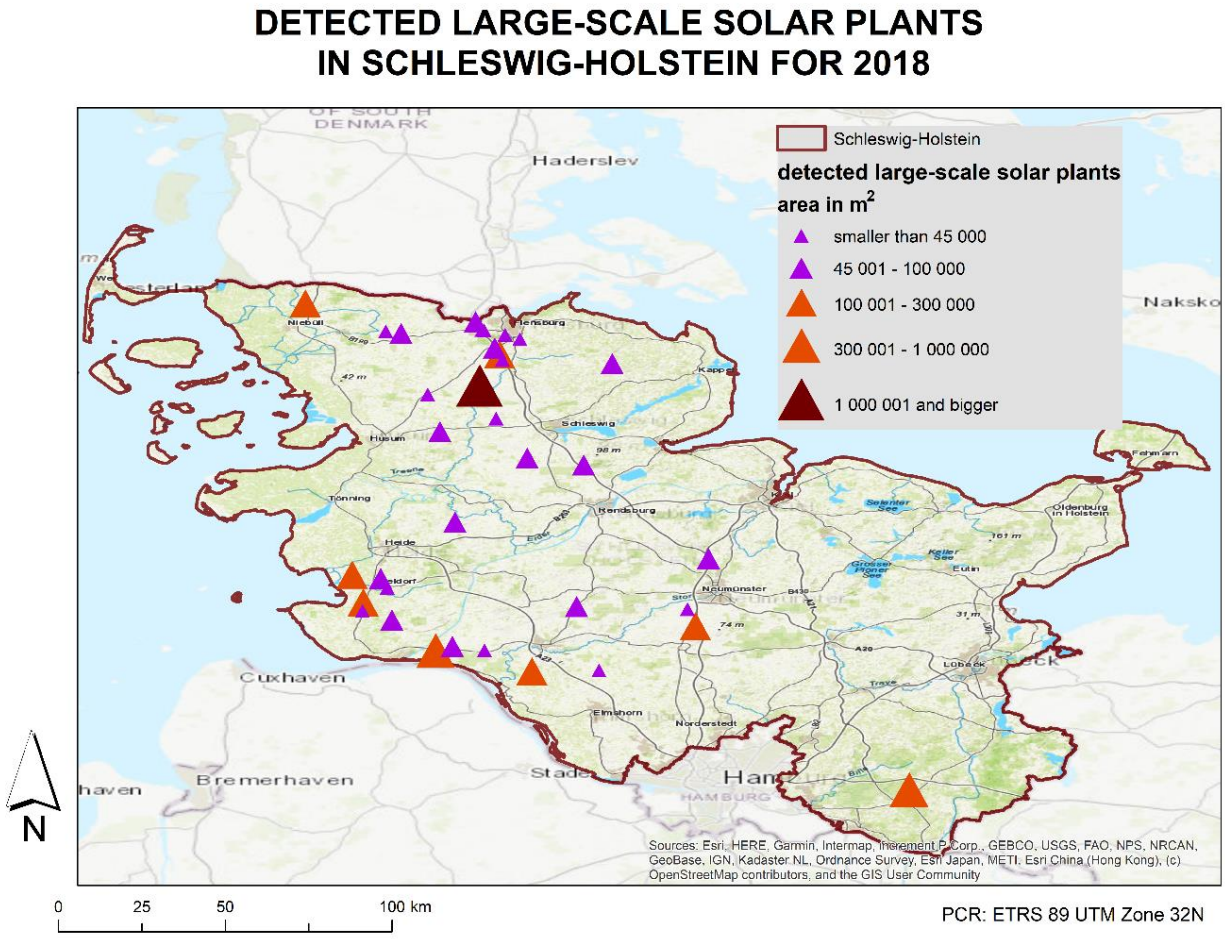


Figure 18. Map of detected large-scale solar plants in the region without validation data

Ultimately, the brief visual inspection was carried out to detect and delete false positive patches, and some of the polygon features were reshaped using the Reshape Feature Tool function. Based on these findings and post-processing, the map with the largest detected large-scale solar installation in Schleswig-Holstein was made. The map includes 35 detected solar plants, where the biggest by area size has the area of approximately 1 km^2 near the Eggebek in the northern part of the state. Similarly, like the largest solar installation in Denmark, it is built on the old unused airport (based on the satellite image observation). The smallest detected solar plant in this analysis was the installation near Haselund with an area size of 12 000 m^2. With this size, it is arguable whether this solar installation can be considered large-scale, but as mentioned before, there is no exact definition relating to size or installed capacity. When comparing the area size information with measuring tools of area size using several web GIS app, the results were almost identical with only a small and insignificant discrepancy. Nevertheless, the biggest tread and disadvantage of this

procedure is that even though all of the detected solar installations were checked using high spatial resolution imagery in the form of the base map in the desktop GIS software, there is still a chance that the process did not detect all the existing plants in the researched region. The verification of such a deficiency another way than by visual inspection (which is for a region of size as Schleswig-Holstein) would be time-consuming and inapplicable. In 2018, most of the large-scale solar plants consist of arrays located closely together as can be seen from for example in Figure 16, but in a few observed cases, the arrays are scattered with a distance between separate solar arrays. This example is presented in Figure 19 on a high spatial resolution image. These solar plants with a higher distance between arrays are not frequent but might occur, and with a spatial resolution of satellite imager used in this study, it is almost impossible to detect them. This is by the author of this study, the biggest weakness of the object of the thesis and object-based supervised classification in general.

Figure 19: The case of the specific spatial deployment which was not classified

5 CONCLUSION AND DISCUSSION

5.1 Conclusion of the project

Analysis of remote sensing imagery has already a broad spectrum of applications, but there are still new possible undiscovered capabilities or methods which can be improved. Although researchers in numerous studies have covered the detection of solar plants or solar panels, none of the existing studies endeavored on the scale of the whole country. With GEE getting a significant role in remote sensing over the last years, the analysis and calculations which were not possible several years ago are now accessible using this cloud platform. This thesis presented the approach of processing satellite data and conducting supervised classification in order to get information about one particular class – solar plants. Despite the fact that the study was conducted using medium spatial resolution imagery, the results are convenient and sufficient to a certain extent.

The study compared and chose the most optimal machine learning classifier for this project, but even though the accuracy assessment numerical results do not reach up to the results of the comparable supervised classification studies with more different classes being included. This is, however, caused by the fact that the goal is to detect only one specific class and thus the antagonistic class has to contain all different land areas which are not solar. Additionally, Denmark is a country with solar energy on rising, but not comparable to leading European countries like Spain or Germany, where the number of large-scale solar plants with a spacious area can be found. Also, the validation data for accuracy assessment contained even the lower boundary of solar plants which can be considered large-scale (installations with a capacity of 500 kW), and the results visually showed that the smaller the solar plant is, the harder it is to detect her using this specific satellite dataset combination. Visual inspection of results yet showed on the example of three different solar plants with diverse area size and installed capacity that the classification, especially using Random Forest with 20 trees classifier performed well. The classification of these solar plants had the distinctive shape which, to a large extent, copied the boundaries of the solar areas with high exactness. Especially these results can be a desirable

original set for transferring the raster classification results into the more usable vector geodata format to be used furthermore supplemented with more precise spatial location, polygon features, and area size. These data can be in most of the cases better and more accurate than the original datasets which are owned by governmental or commercial agencies and companies (in general) and are based on the registers of address and not on the precise location of the large-scale solar plant. Nowadays, when the world is over-filled with data, the precise data plays a significant role, and the accurate location information can lead to a better consequent spatial analysis. Ultimately, no matter the regions of study as the approach can be applied throughout the world, the information on the locations, area sizes and numbers of solar plants can be particularly valuable for mapping and surveying the development of this area of sustainable energy over the time.

The study presented an innovative approach with a few not frequently used processes and methods. One of the not regularly seen ideas implemented in the research was to combine also radar data from Sentinel-1 to (in the supervised classification of remote sensing imagery widely used) Sentinel-2 data. As compared in 4.4, the results using imagery from both satellite missions exceeded results working only with Sentinel-2 data. On top of that, indices (NDVI and NDWI) were calculated and with a function added to Sentinel-2 bands.

5.2 The future direction of the thesis research

There is a huge potential for more frequent utilization of object detection in remote sensing of different variety of objects, not only solar plants and solar panels. Unlike land cover / land use classifications, which dominated the research in the foregone years, object detection from satellite imagery is dependent on the a) spatial, temporal, radiometric and spectral resolution of the provided imagery and b) computational and processing capabilities of tools and software. Related to this, it is expected that GEE (as still at the time of writing this study a new cloud computing platform) will continue to develop simultaneously with cloud processing research. Similarly, new satellite missions with a better optical (not only) sensor will be launched. Thus, it is expected that the potential object detection will spread on the smaller objects even when computing on the large regions of interests.

Correspondingly, this will affect the research on the detection of solar plants and panels and presumably overcome the disadvantages mentioned in this study. Renewable forms of energy are currently one of the most discussed topics in the energy sector and having better, more precise and frequently updated data on solar plants even though as an additional source to existing databases will play its role. By the author of this study is believed that this will happen mostly in developed countries with less sufficient availability of energy data. Ultimately, as the satellite data is becoming more accessible and there are possibilities to observe the whole world, having data on solar energy for different countries can have an effect on the political collaboration and partnership cooperation of countries for better and sustainable future of energy.

The complete code is provided in Appendix 1 on the example of classifier Random Forest with 20 trees and can be reproduced for another research, however, some of the data could not be provided as open. The code is commented and can be used with small changes when own training and validating data are provided.

5.3 The utility of results and recommendations

Despite this thesis being conducted for academic purposes, the results and the processing methods and ideas can be potentially used furthermore. There are numerous companies and start-ups from the last few years which detect, observe or follow numerous objects on the planet like cargo ships movement, parking lots and the current number of cars or residential pools. The goal of these companies is not only to provide this data further but mainly to use them for mapping economic index, forecasting the trends in the development (mainly economic development) and transfer these trends for businesses, defense sector, government, and many others.

Among the usage for the governments and non-governmental organizations could be fighting energy fraud as the investments from funds for solar energy are indispensable. Another potential usage factor could be linking the area size with estimated capacity estimation. This information would stress the relevance of satellite imagery obtained data compared to address-based data with only longitude and latitude information. It was preliminary planned that this study would also implement these knowing and conduct additional analysis. It is known and affirmed by many studies that there is a strong relation between production power / installed capacity

and the solar plant size, but the consequential study research would be needed. However, the potential to link the results and findings of this study with capacity and power estimation is high and might be conducted in the future.

5.4 Capabilities and limitations of GEE in this study

This research demonstrated how GEE is a powerful tool when pre-processing satellite data and conducting classification on a large region of interest area. The processing approach using server along with constantly updated and extensive catalog of various satellite data are just the main ones of many advantages GEE have. However, there are certain limitations and inconveniences which were brought working on this thesis project. GEE is capable of working with spacious datasets and conducting analysis on a large area, but it is paid off by a considerable time needed to perform these big analyses. Especially the loading of the classification map lasted a couple of minutes (3 to 5) when performing classification for whole Denmark.

On top of that, this classification layer was not stable when zooming to a certain location or moving around. Additionally, for numerical results or matrices, the options are that the results can be printed out to the console or saved to several format files and exported to the Google Drive. However, on-demand mode (printing to the console of Code Editor) is restricted to the 5 minutes running time which was exceeded in most of the analyses. Moreover, when exporting results in another format outside GEE environment, the calculation and export lasted usually from 10 to 35 minutes when performing classification for all Denmark. Also, memory limitations exist, so the training data had to be reduced (specifically the number of non-solar training samples). Additionally, the export of the raster format results of classification had to be clipped with the smaller regions of interests, because there is a limitation of pixels when exporting rasters to drive.

The other demerits of the analysis and results were caused not by GEE but specifically by deficiencies and capability boundaries by the provided satellite imagery data.

6 BIBLIOGRAPHY

"API Tutorials | Google Earth Engine API | Google Developers." n.d. Accessed June 3, 2019. https://developers.google.com/earth-engine/playground.

Behar, Omar. 2018. "Solar Thermal Power Plants – A Review of Configurations and Performance Comparison." *Renewable and Sustainable Energy Reviews* 92 (May): 608–27. https://doi.org/10.1016/j.rser.2018.04.102.

Belgiu, Mariana, and Lucian Drăgu. 2016. "Random Forest in Remote Sensing: A Review of Applications and Future Directions." *ISPRS Journal of Photogrammetry and Remote Sensing* 114: 24–31. https://doi.org/10.1016/j.isprsjprs.2016.01.011.

Bittencourt, H.R., and R.T. Clarke. 2004. "Use of Classification and Regression Trees (CART) to Classify Remotely-Sensed Digital Images," no. August: 3751–53. https://doi.org/10.1109/igarss.2003.1295258.

Bomans, Bart, Gabriele Buttafuoco, Fabio Castaldi, Bas van Wesemael, Kathrin Ward, Maximilian Brell, Sabine Chabrillat, Andreas Hueni, and Kristin Vreys. 2018. "Evaluating the Capability of the Sentinel 2 Data for Soil Organic Carbon Prediction in Croplands." *ISPRS Journal of Photogrammetry and Remote Sensing* 147 (October 2018): 267–82. https://doi.org/10.1016/j.isprsjprs.2018.11.026.

Chen, Renxi, Xinhui Li, and Jonathan Li. 2018. "Object-Based Features for House Detection from RGB High-Resolution Images." *Remote Sensing* 10 (3): 1–24. https://doi.org/10.3390/rs10030451.

Cheng, Gong, and Junwei Han. 2016. "A Survey on Object Detection in Optical Remote Sensing Images." *ISPRS Journal of Photogrammetry and Remote Sensing* 117: 11–28. https://doi.org/10.1016/j.isprsjprs.2016.03.014.

ESA. 2019a. "Sentinel-1 - Overview - Sentinel Online." 2019. https://sentinel.esa.int/web/sentinel/missions/sentinel-1/overview.

———. 2019b. "Sentinel-2 - Overview - Sentinel Online." 2019. https://sentinel.esa.int/web/sentinel/missions/sentinel-2/overview.

EuropeanEnergy.dk. 2015. “Vandel — European Energy.” 2015. https://www.europeanenergy.dk/en/vandel.

Farda, N. M. 2017. “Multi-Temporal Land Use Mapping of Coastal Wetlands Area Using Machine Learning in Google Earth Engine.” In *IOP Conference Series: Earth and Environmental Science*. Vol. 98. https://doi.org/10.1088/1755-1315/98/1/012042.

Furbo, Simon, Janne Dragsted, Bengt Perers, Elsa Andersen, Federico Bava, and Kristian Pagh Nielsen. 2018. “Yearly Thermal Performances of Solar Heating Plants in Denmark – Measured and Calculated.” *Solar Energy* 159 (December 2017): 186–96. https://doi.org/10.1016/j.solener.2017.10.067.

Google Earth Engine API. 2017. “Sentinel-1 Algorithms | Google Earth Engine API | Google Developers.” 2017. https://developers.google.com/earth-engine/sentinel1.

Gorelick, Noel, Matt Hancher, Mike Dixon, Simon Ilyushchenko, David Thau, and Rebecca Moore. 2017. “Google Earth Engine: Planetary-Scale Geospatial Analysis for Everyone.” *Remote Sensing of Environment* 202: 18–27. https://doi.org/10.1016/j.rse.2017.06.031.

Hansen, Matthew C., and Thomas R. Loveland. 2012. “A Review of Large Area Monitoring of Land Cover Change Using Landsat Data.” *Remote Sensing of Environment* 122: 66–74. https://doi.org/10.1016/j.rse.2011.08.024.

Hasmadi, Mohd, Pakhriazad HZ, and Shahrin MF. 2005. “Evaluating Supervised and Unsupervised Techniques for Land Cover Mapping Using Remote Sensing Data.” *Geografia - Malaysian Journal of Society and Space* 5 (1): 1–10.

Healey, Sean, Warren Cohen, Zhiqiang Yang, Noel Gorelick, Justin Braaten, Robert Kennedy, and Lucas Cavalcante. 2018. “Implementation of the LandTrendr Algorithm on Google Earth Engine.” *Remote Sensing* 10 (5): 691. https://doi.org/10.3390/rs10050691.

Imamoglu, Nevrez, Motoki Kimura, Hiroki Miyamoto, Aito Fujita, and Ryosuke Nakamura. 2017. “Solar Power Plant Detection on Multi-Spectral Satellite Imagery Using Weakly-Supervised CNN with Feedback Features and m-PCNN Fusion,” 6–9. http://arxiv.org/abs/1704.06410.

Ishii, Tomohiro, Edgar Simo-Serra, Satoshi Iizuka, Yoshihiko Mochizuki, Akihiro

Sugimoto, Hiroshi Ishikawa, and Ryosuke Nakamura. 2017. “Detection by Classification of Buildings in Multispectral Satellite Imagery.” *Proceedings - International Conference on Pattern Recognition*, 3344–49. https://doi.org/10.1109/ICPR.2016.7900150.

Kranjčić, Nikola, Damir Medak, Robert Župan, and Milan Rezo. 2019. “Support Vector Machine Accuracy Assessment for Extracting Green Urban Areas in Towns.” *Remote Sensing* 11 (6): 655. https://doi.org/10.3390/rs11060655.

Lee, Janice Ser Huay, Serge Wich, Atiek Widayati, and Lian Pin Koh. 2016. “Detecting Industrial Oil Palm Plantations on Landsat Images with Google Earth Engine.” *Remote Sensing Applications: Society and Environment* 4 (November): 219–24. https://doi.org/10.1016/j.rsase.2016.11.003.

Li, Huan, Wei Wan, Yu Fang, Siyu Zhu, Xi Chen, Baojian Liu, and Yang Hong. 2019. “A Google Earth Engine-Enabled Software for Efficiently Generating High-Quality User-Ready Landsat Mosaic Images.” *Environmental Modelling and Software* 112 (November 2018): 16–22. https://doi.org/10.1016/j.envsoft.2018.11.004.

Liss, Brady, Matthew D. Howland, and Thomas E. Levy. 2017. “Testing Google Earth Engine for the Automatic Identification and Vectorization of Archaeological Features: A Case Study from Faynan, Jordan.” *Journal of Archaeological Science: Reports* 15 (September): 299–304. https://doi.org/10.1016/j.jasrep.2017.08.013.

Liu, Xiaoping, Guohua Hu, Yimin Chen, Xia Li, Xiaocong Xu, Shaoying Li, Fengsong Pei, and Shaojian Wang. 2018. “High-Resolution Multi-Temporal Mapping of Global Urban Land Using Landsat Images Based on the Google Earth Engine Platform.” *Remote Sensing of Environment* 209 (February): 227–39. https://doi.org/10.1016/j.rse.2018.02.055.

Mack, Benjamin, Ribana Roscher, and Björn Waske. 2014. “Can i Trust My One-Class Classification?” *Remote Sensing* 6 (9): 8779–8802. https://doi.org/10.3390/rs6098779.

Malof, Jordan M., Rui Hou, Leslie M. Collins, Kyle Bradbury, and Richard Newell. 2015. “Automatic Solar Photovoltaic Panel Detection in Satellite Imagery.” *2015 International Conference on Renewable Energy Research and Applications, ICRERA 2015*, no. November: 1428–31.

https://doi.org/10.1109/ICRERA.2015.7418643.

Maxwell, Aaron E., Timothy A. Warner, and Fang Fang. 2018. “Implementation of Machine-Learning Classification in Remote Sensing: An Applied Review.” *International Journal of Remote Sensing* 39 (9): 2784–2817. https://doi.org/10.1080/01431161.2018.1433343.

Mountrakis, Giorgos, Jungho Im, and Caesar Ogole. 2011. “Support Vector Machines in Remote Sensing: A Review.” *ISPRS Journal of Photogrammetry and Remote Sensing* 66 (3): 247–59. https://doi.org/10.1016/j.isprsjprs.2010.11.001.

Mutanga, Onisimo, and Lalit Kumar. 2019. “Google Earth Engine Applications.” *Remote Sensing* 11 (5): 591. https://doi.org/10.3390/rs11050591.

Navratilova, Pavlina. 2013. “Solar Energy in the Czech Republic History, Nowadays, Prospects.” Masaryk University.

Nyheim, Thomas, and Anders Roed Bruhn. 2017. “Next Generation Opportunities in Utility-Scale Solar.” *QVARTZ*, no. June: 1–6.

Padarian, José. 2018. “Weekend Project: Detecting Solar Panels from Satellite Imagery.” 2018. https://towardsdatascience.com/weekend-project-detecting-solar-panels-from-satellite-imagery-f6f5d5e0da40.

Pasco, Xavier, Gil Denis, Franck Ranera, Steven Hosford, Hélène de Boissezon, and Bruno Montfort. 2016. “The Evolution of Earth Observation Satellites in Europe and Its Impact on the Performance of Emergency Response Services.” *Acta Astronautica* 127: 619–33. https://doi.org/10.1016/j.actaastro.2016.06.012.

Phillips, Jason. 2013. “Determining the Sustainability of Large-Scale Photovoltaic Solar Power Plants.” *Renewable and Sustainable Energy Reviews* 27: 435–44. https://doi.org/10.1016/j.rser.2013.07.003.

Piyatadsananon, Pantip. 2016. “Spatial Factors Consideration in Site Selection of Ground-Mounted PV Power Plants.” *Energy Procedia* 100 (September): 78–85. https://doi.org/10.1016/j.egypro.2016.10.135.

Solares, Cristina, and Ana Maria Sanz. 2007. “Different Bayesian Network Models in the Classification of Remote Sensing Images.” *Intelligent Data Engineering and*

Automated Learning - IDEAL 2007, 10–16. https://doi.org/10.1007/978-3-540-77226-2_2.

SpaceKnow. 2016. “Mapping Renewable Energy : Rising Solar Plants Making a Global Difference The Solar Power Boom Is the Future of the Energy Sector.”

Taufik, Afirah, Sharifah Sakinah Syed Ahmad, and Nor Fatin Emiliza Khairuddin. 2017. “Classification of Landsat 8 Satellite Data Using Fuzzy C-Means.” *Proceedings of the 2017 International Conference on Machine Learning and Soft Computing - ICMLSC ’17*, no. August: 58–62. https://doi.org/10.1145/3036290.3036330.

Thamilselvana, P, and J. G. R. Sathiaseelan. 2015. “A Comparative Study of Data Mining Algorithms for Image Classification.” *International Journal of Education and Management Engineering* 5 (2): 1–9. https://doi.org/10.5815/ijeme.2015.02.01.

Topaloğlu, Raziye Hale, Elif Sertel, and Nebiye Musaoğlu. 2016. “Assessment of Classification Accuracies of Sentinel-2 and Landsat-8 Data for Land Cover/Use Mapping.” *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives* 41 (July): 1055–59. https://doi.org/10.5194/isprsarchives-XLI-B8-1055-2016.

Tsoutsos, Theocharis, Niki Frantzeskaki, and Vassilis Gekas. 2005. “Environmental Impacts from the Solar Energy Technologies.” *Energy Policy* 33 (3): 289–96. https://doi.org/10.1016/S0301-4215(03)00241-6.

Wasson, Matthew F., Yolandita Franklin, Andrew A. Pericak, David J. Campagna, Nicholas E. Clinton, David A. Kroodsma, Emily S. Bernhardt, Matthew R. V. Ross, Christian J. Thomas, and John F. Amos. 2018. “Mapping the Yearly Extent of Surface Coal Mining in Central Appalachia Using Landsat and Google Earth Engine.” *Plos One* 13 (7): e0197758. https://doi.org/10.1371/journal.pone.0197758.

Woodcock, CE, Richard Allen, Martha Anderson, Allan Belward, Robert Bindschandler, Warren B. Cohen, Feng Gao, et al. 2008. “Free Access to Landsat Imagery Teach by the Book Science Education :” *Science* 320 (May): 1011. https://doi.org/10.1126/science.320.5879.1011a.

Yang, Juan, Zhiwei Ye, Xu Zhang, Wei Liu, and Huazhong Jin. 2018. “Attribute Weighted Naive Bayes for Remote Sensing Image Classification Based on Cuckoo

Search Algorithm.” *2017 International Conference on Security, Pattern Analysis, and Cybernetics, SPAC 2017* 2018-Janua: 169–74. https://doi.org/10.1109/SPAC.2017.8304270.

7 APPENDIX

```
// regions of Denmark are using preloaded asset
var region1 = DNK1.filter(ee.Filter.eq('VARNAME_1', 'Zealand'));
var region2 = DNK1.filter(ee.Filter.eq('VARNAME_1', 'South Denmark'));
var region3 = DNK1.filter(ee.Filter.eq('VARNAME_1', 'Central Jutland'));
var region4 = DNK1.filter(ee.Filter.eq('VARNAME_1', 'North Jutland'));
// solarDenmark is dataset about training solar areas and is in FusionTable format
var solarDenmark = ee.FeatureCollection('ft:1yAPTE-IsxMPlmQhjNI3h-5vkaq7nlH40z0iQsqpA');
//merging of 4 Denmark regions (excluding Capital Region and additional region of Germany,
//where the training data are not provided)
var region = region1.merge(region2).merge(region3).merge(region4).merge(Shleswig_Holstein);
//merging of training data, both solar and nonsolars, non_solar_areas dataset is loaded in GEE as asset
var solar_non_solar = non_solar_areas.merge(solarDenmark);

//function that creates NDVI and NDWI indices (must be later used for image collection)
var addNDVIBands = function(image) {
  var NDVI = image.addBands(image.normalizedDifference(['B8', 'B4']));
  var NDWI = NDVI.addBands(NDVI.normalizedDifference(['B3', 'B8']));
  return NDWI.addBands(NDWI.metadata('system:time_start'));
};

// Loading the Sentinel-1 image collection
var sentinel1Collection = ee.ImageCollection('COPERNICUS/S1_GRD')
                  .filterDate('2018-01-01', '2018-06-30')
                  .filterBounds(region);

// Filtering by metadata properties.
var metaSentinel1 = sentinel1Collection
  .filter(ee.Filter.listContains('transmitterReceiverPolarisation', 'VV'))
  .filter(ee.Filter.listContains('transmitterReceiverPolarisation', 'VH'))
  .filter(ee.Filter.eq('instrumentMode', 'IW'));

// Filtering to get images from different look angles.
var vhAscending = metaSentinel1.filter(ee.Filter.eq('orbitProperties_pass', 'ASCENDING'));
var vhDescending = metaSentinel1.filter(ee.Filter.eq('orbitProperties_pass', 'DESCENDING'));

// Create a composite from means at different polarizations and look angles.
var composite = ee.Image.cat([
  vhAscending.select('VH').mean(),
  ee.ImageCollection(vhAscending.select('VV').merge(vhDescending.select('VV'))).mean(),
```

```
  vhDescending.select('VH').mean()
]).focal_median();

// Display as a composite of polarization and backscattering characteristics.
//Map.addLayer(composite, {min: [-25, -20, -
25], max: [0, 10, 0]}, 'composite');

// Loading Sentinel-2 TOA reflectance data.
// Map the function over half a year of data and take the median.
var sentinel2Collection = ee.ImageCollection('COPERNICUS/S2')
                  .filterDate('2018-01-01', '2018-06-30')
                  // Pre-filter to get less cloudy granules.
                  .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 20))
                  .filterBounds(region)
                  .map(addNDVIBands);

//function which renames NDVI index
function renameNDVI(img) {
  var ndvi = img.select(["nd"], ["NDVI"]);
  return img.addBands(ndvi);
}
var ndviColl = sentinel2Collection.map(renameNDVI);

//function which renames NDWI index
function renameNDWI(img) {
  var ndwi = img.select(["nd_1"], ["NDWI"]);
  return img.addBands(ndwi);
}
var ndwiColl = ndviColl.map(renameNDWI);

var sentinel2Composite = ndwiColl.median();
print(sentinel2Composite);

var visParamsTrue = {bands: ['B4', 'B3', 'B2'], min: 0, max: 3000, gamma: 1.4};

//Map.addLayer(sentinel2Composite.clip(region), visParamsTrue, 'Sentinel 2 True',
 false);
//Map.addLayer(sentinel2Composite.clip(region), visParamsTrue, 'Sentinel 2 True',
 false);

//This code is based on the RF classification, but can the classifier can be simp
ly changed
/***************************
Random Forest Classification
***************************/
// Loading preprocessed training data with a feature property solar = 0 or 1
var trainingData = solar_non_solar;

var color_solar = trainingData
  .filter(ee.Filter.neq('solar', null))
  .reduceToImage({
    properties: ['solar'],
    reducer: ee.Reducer.first()
  });
// Merging of Sentinel-2 and 1 preprocced data
var finalComposite = sentinel2Composite.addBands(composite);

// Training sample data
// Selecting of bands suitable for classification from Sentinel-2 and 1
var bands2 = ['B2', 'B3', 'B4', 'B5', 'B6', 'B7', 'B8', 'B11', 'B12', 'NDVI', 'ND
WI'];
var bands1 = ['VV', 'VH'];

var bands = bands2.concat(bands1); //Merging of bands from Sentinel 2 and Sentine
l 1
```

```
99.
100.  var input = finalComposite.select(bands);
101.
102.  var classifierTraining = input.select(bands)
103.        .sampleRegions({
104.          collection: trainingData,
105.          properties: ['solar'],
106.          scale: 20
107.        });
108.
109.  // Instantiating Random Forest (or different) classifier and training it:
110.  var classifier = ee.Classifier.randomForest(20).train({
111.    features: classifierTraining,
112.    classProperty: 'solar',
113.    inputProperties: bands
114.  });
115.
116.  //Classifying the image filtered by region
117.  var classified = input.select(bands).classify(classifier).clip(Shleswig_Holste
    in);
118.
119.  // Define a palette for the IGBP classification.
120.  var igbpPalette = [
121.    'A9A9A9', //non-solar
122.    'CCCC00' //solar
123.  ];
124.
125.  Map.addLayer(classified, {palette: igbpPalette, min: 0, max: 1}, 'classificati
    on visualization');
126.  Map.centerObject(region, 10);
127.  Map.addLayer(validation_PV_points_90, {}, 'validation_points');
128.
129.  // possible export of Schleswig_Holstein region solely as TIFF
130.  /*
131.  Export.image.toDrive({
132.    image: classified,
133.    description: 'Germany_solar_plants',
134.    region: Schleswig_Holstein.geometry().bounds(),
135.    scale: 20
136.  });
137.  */
138.
139.  /******************
140.  Accuracy assessment (2)
141.  ******************/
142.
143.  var trainingTesting2 = validation_PV_points;
144.
145.  var validation = classified.sampleRegions({
146.    collection: trainingTesting2,
147.    properties: ['solar'],
148.    scale: 20
149.  });
150.
151.  var testAccuracy = validation.errorMatrix('solar', 'classification');
152.
153.  var confMatrix = ee.Feature(null, {matrix: testAccuracy.array()});
154.  var overAccuracy = ee.Feature(null, {matrix: testAccuracy.accuracy()});
155.  var prodAccracy = ee.Feature(null, {matrix: testAccuracy.producersAccuracy()})
    ;
156.  var consAccuracy = ee.Feature(null, {matrix: testAccuracy.consumersAccuracy()}
    );
157.  var kappa2 = ee.Feature(null, {matrix: testAccuracy.kappa()});
158.  var order2 = ee.Feature(null, {matrix: testAccuracy.kappa()});
159.
```

```
// Printing the results to console in GEE (sample)
//print('Confusion Matrix:', confMatrix);
//print('Overal Accuracy:', overAccuracy);

// Exporting the results in the CSV format to Google Drive (sample)
Export.table.toDrive({
  collection: ee.FeatureCollection(overAccuracy),
  description: 'overalAccuracy_Denmark_PV110_RF20',
  fileFormat: 'CSV'
});

Export.table.toDrive({
  collection:  ee.FeatureCollection(prodAccracy),
  description: 'producersAccuracy_Denmark_PV110_RF20',
  fileFormat: 'CSV'
});

Export.table.toDrive({
  collection:  ee.FeatureCollection(consAccuracy),
  description: 'consumersAccuracy_Denmark_PV110_RF20',
  fileFormat: 'CSV'
});

Export.table.toDrive({
  collection:  ee.FeatureCollection(kappa2),
  description: 'kappa_Denmark_PV110_RF20',
  fileFormat: 'CSV'
});

/******************
Accuracy assessment (1)
*******************/

var testTraining = classifierTraining.randomColumn();
var trainedSet = testTraining
.filter(ee.Filter.lessThan('random', 0.7));
var testingSet = testTraining
.filter(ee.Filter.greaterThanOrEquals('random', 0.7));

// Training the classifier with the trainedSet:
var trained = ee.Classifier.randomForest(10).train({
  features: trainedSet,
  classProperty: 'solar',
  inputProperties: bands
});

// classifying the testingSet and get a confusion matrix
var confusionMatrix = ee.ConfusionMatrix(testingSet.classify(trained)
      .errorMatrix({
        actual: 'solar',
        predicted: 'classification'
      }));

var confMat = ee.Feature(null, {matrix: confusionMatrix.array()});
var overAccu = ee.Feature(null, {matrix: confusionMatrix.accuracy()});
var prodAccu = ee.Feature(null, {matrix: confusionMatrix.producersAccuracy()})
;
var consAccu = ee.Feature(null, {matrix: confusionMatrix.consumersAccuracy()})
;
var kappa = ee.Feature(null, {matrix: confusionMatrix.kappa()});
var order = ee.Feature(null, {matrix: testAccuracy.order()});

// Printing the results to console in GEE (sample)
print('Confusion Matrix:', confMat);
print('Overal Accuracy:', overAccu);
```

```
223.
224.  // Exporting the results in the CSV format to Google Drive (sample)
225.  Export.table.toDrive({
226.    collection: ee.FeatureCollection(confMat),
227.    description: 'confMat_Zealand',
228.    fileFormat: 'CSV'
229.  });
230.
231.  //Print the confusion matrix and expand the object to inspect the matrix.
232.  print('Confusion matrix:', confusionMatrix);
233.  print('Overal Accuracy:', confusionMatrix.accuracy());
234.  print('Producers Accuracy:', confusionMatrix.producersAccuracy());
235.  print('Consumers Accuracy:', confusionMatrix.consumersAccuracy());
236.
237.
238.  Map.addLayer(color_solar, {palette: ['ff0000', '00ff00']}, 'solar non solar');
239.
240.  /******************
241.  Adding the legend for GEE
242.  ********************/
243.
244.  // set position of panel
245.  var legend = ui.Panel({
246.    style: {
247.      position: 'bottom-left',
248.      padding: '8px 15px'
249.    }
250.  });
251.
252.  // Create legend title
253.  var legendTitle = ui.Label({
254.    value: '',
255.    style: {
256.      fontWeight: 'bold',
257.      fontSize: '18px',
258.      margin: '0 0 4px 0',
259.      padding: '0'
260.      }
261.  });
262.
263.  // Add the title to the panel
264.  legend.add(legendTitle);
265.
266.  // Creates and styles 1 row of the legend.
267.  var makeRow = function(color, name) {
268.
269.        // Create the label that is actually the colored box.
270.        var colorBox = ui.Label({
271.          style: {
272.            backgroundColor: '#' + color,
273.            // Use padding to give the box height and width.
274.            padding: '8px',
275.            margin: '0 0 4px 0'
276.          }
277.        });
278.
279.        // Create the label filled with the description text.
280.        var description = ui.Label({
281.          value: name,
282.          style: {margin: '0 0 4px 6px'}
283.        });
284.
285.        // return the panel
286.        return ui.Panel({
```

```
          widgets: [colorBox, description],
          layout: ui.Panel.Layout.Flow('horizontal')
        });
};

//  Palette with the colors
var palette =['FF0000', '00ff00', 'CCCC00'];

// name of the legend
var names = ['non solar training','solar training','results solar area'];

// Add color and and names
for (var i = 0; i < 3; i++) {
  legend.add(makeRow(palette[i], names[i]));
  }
// add legend to map
Map.add(legend);
```

Appendix 1. The complete JavaScript code for the study using RS classifier