Master's Programme in Data Science

Data Science project: Land use change detection

Group Landuse: Onni Aarne, Jaakko Kuurne, Roope Niemi, Leonardo Pratesi, Haoyu Wei.

May 2021

University of Helsinki
Faculty of Science

HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta — Fakultet — Faculty	Koulutusohjelma — Utbildningsprogram — Degree programme
Faculty of Science	Master's Programme in Data Science

Tekijä — Författare — Author

Group Landuse: Onni Aarne, Jaakko Kuurne, Roope Niemi, Leonardo Pratesi, Haoyu Wei.

Työn nimi — Arbetets titel — Title

Data Science project: Land use change detection

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidantal — Number of pages
Report	May 2021	21

Tiivistelmä — Referat — Abstract

ACM Computing Classification System (CCS):

Avainsanat — Nyckelord — Keywords

layout, summary, list of references

Säilytyspaikka — Förvaringsställe — Where deposited

Muita tietoja — Övriga uppgifter — Additional information

Contents

1 Introduction **2**

2 The technical approach **3**

2.1 The dataset . . . 3

2.2 The CCDC algorithm . . . 4

2.3 Classification . . . 5

2.4 Discussion . . . 6

3 Results **7**

3.1 Google Earth Engine scripts . . . 7

3.2 Classification results . . . 7

3.2.1 Paraguay . . . 8

3.2.2 Finland and other countries . . . 9

3.3 Evaluation scripts . . . 9

3.3.1 Evaluation notebook . . . 9

3.3.2 Visualizer app . . . 10

3.3.3 Discussion . . . 10

4 Conclusion **11**

References **12**

Appendix A Introduction to remote sensing **13**

A.1 Remote Sensing . . . 13

Appendix B Details of technical approach **15**

B.1 Change detection . . . 15

B.2 Classification . . . 16

Appendix C More results from Finland **17**

Appendix D Visualizer application **19**

1. Introduction

Climate change is undoubtedly one of the greatest challenges facing humanity this century. The basic cause of climate change is excessive emission of greenhouse gases, such as CO_2, compared to their reabsorption, leading to an increase in the overall concentration of these gases in the atmosphere. Thus, at this time more than ever, we need the ability of our forests to absorb CO_2 out of the atmosphere. Sadly, at the same time, we have seen accelerating deforestation [5].

To be able to combat this problem, we need to know where the deforestation is happening, preferably as quickly as possible. The Forest and Agriculture Organization of the United Nations (FAO) contributes to solving this epistemic problem through their Global Forest Resources Assessments (FRA). A core component of the FRA is a remote sensing survey [9]. The survey is conducted in collaboration with local domain experts in different countries, who estimate land use at sample sites by visually interpreting Landsat-based satellite imagery of the sites at different times.

This human visual interpretation approach taken by the FRA is highly labor intensive, and results in there being a substantial delay in the publication of their reports. However, the experts at FAO recognized that the process could be expedited substantially if a computerized classification system was used to do some of the work of detecting the most obvious instances of areas where clearly there had been no change. The human experts could then focus on the areas where there has likely been change. The Finnish non-profit Avoin ry then helped coordinate this collaboration between FAO and the University of Helsinki to develop a solution to this problem.

The solution we developed was a set of scripts in the Google Earth Engine system [6], that can be used to train land cover classifiers. Additionally, we developed tools for analyzing and visualizing the results of these models, to facilitate their development and adoption. FAO and Avoin now plan to further develop this system, so that it can be taken into use to speed up future FRA remote sensing surveys.

In Chapter 2 we will explain the technical approach we took to solving this problem. This is followed in Chapter 3 by an overview of the system we developed. In Chapter 4 we then summarize all of this and add some thoughts on future work. More technical details of our work is covered in the appendices.

2. The technical approach

Our primary approach to solving the problem was the Continuous Change Detection and Classification (CCDC) algorithm developed by Zhu and Woodcock [11]. After an overview of our data in Section 2.1, the details of the change detection part of this approach are discussed in Section 2.2, followed by elaboration of the classification step of the system in Section 2.3. Finally in Section 2.4 we discuss some other approaches that were considered during the project. Some details of our approach that couldn't be covered here due to space limitation are covered in Appendix B.

This approach was implemented in Google Earth Engine [6], which is a free-to-use cloud based geospatial analysis system. Google Earth Engine provided us with the computing capacity we needed, and existing implementations of many of the key algorithms. It also provided us with no shortage of its own technical difficulties, which are further discussed in Section 2.4.

2.1 The dataset

As our primary dataset, we used visual interpretation data for seven countries, collected as part of a previous forest resources assessment survey. We received this data from our client at the FAO, in the form of shapefiles and .xlsx files describing the locations and properties of sample hexagons in seven different countries: Bolivia, Ethiopia, Finland, Indonesia, Nigeria, Paraguay and Tanzania. The number of these hexagons varied by country, ranging from fewer than 800 in Finland, to over 17 000 in Indonesia. The .xlsx files [†] contained visual interpretation results from those hexagons.

By visual interpretation results we mean the estimates of various aspects of land use in the hexagon in the years 2000, 2010, and 2018. These were produced by local domain experts by looking at Landsat-based satellite imagery. The most important information here was information about what percent of the hexagon was devoted to different categories of land use in the different years, and the information about forest loss and forest gain implied therein.

[†] In many different languages, all structured differently.

2.2 The CCDC algorithm

The basic idea of the CCDC algorithm [11] is to first fit mechanistically plausible time series models to each pixel in the Landsat data. These time series models account for expected change, such as seasonal variation and secular trends. Then changes in land cover are detected by looking for sudden changes not predicted by the time series model. This gives us a segmentation of the analyzed time window into periods with different land cover. However, we do not yet know which type of land cover each segment corresponds to. Thus the segments then need to be classified by land cover type. This is discussed in the next section.

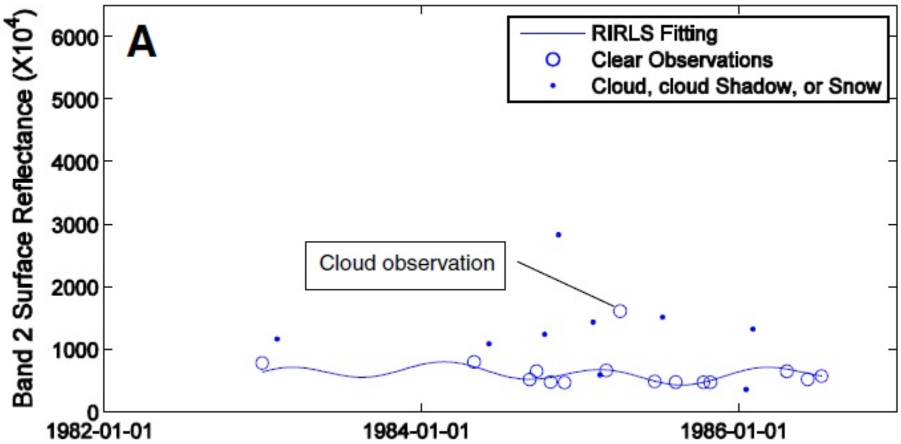


Figure 2.1: CCDC function is being fitted to a collection of band 2 surface reflectance observations. The cloudy observations are ignored and do not affect how the function is fitted. The figure is from the paper by Zhu and Woodcock [11]

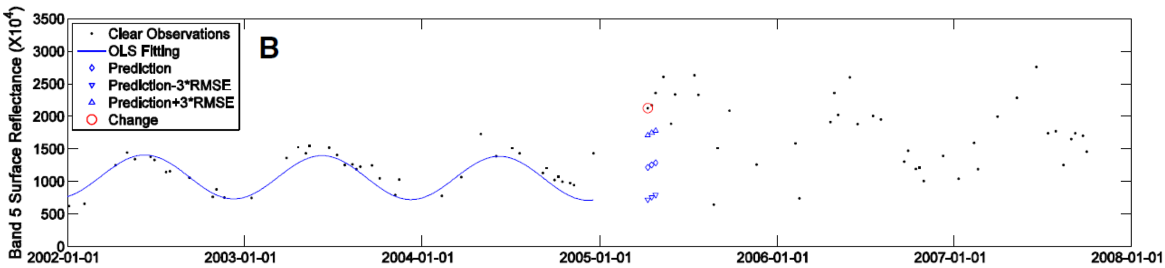


Figure 2.2: The observation marked with the red circle has a band 5 surface reflectance value of more than prediction + 3 x RMSE illustrated by the triangle points. As it is the first observation exceeding the threshold in a series of other observations exceeding the threshold, the observation indicates a change in the land cover. The figure is from the paper by Zhu and Woodcock [11].

2.3 Classification

In the classification step, the segments produced by the change detection model are classified by using the coefficients (learned parameters) of the time series model for each segment as the features. This transforms the task from a time series domain to a more conventional classification task, and gives us a relatively low-dimensional and mechanistically motivated feature space [11]. We used a random forest classifier for this step, because.

However, here we meet our first major challenge: A classifier needs labels. We did receive data about past land use from the client as described, but did not use this data for training our model because it was ill-suited for that task in many ways. A major issue is that the visual interpretation data only describes the amount of each land use category in each sample hexagon, but does not tell us which parts of the hexagon belong to which category. Additionally, the information only covered land use, rather than land cover. Land cover refers to simply to what is physically there on the land: is it buildings or trees or bare soil or something else. Land use refers to what the land is being used for. This is a much more abstract and subtle concept. For example, in Finland it is common to cut down large patches of commercial forest at once and then let them regrow. After such a cutting, the are no longer has trees and thus the land cover has changed. But if the owners of the land intend to replant the forest, the land use of the area is still forest, even when there are no trees. Similarly, an urban park densely covered with trees is not a forest from the land use perspective, because it is considered to be part of a human settlement. Because of this, land use requires human contextual understanding to determine and cannot be simply inferred from satellite readings alone. For this reason the client instructed us to develop a land cover classifier, rather than a land use classifier. Due to these and other difficulties with the visual interpretation data, we primarily used it for validation, and sought other sources of land cover labels. We converged on two possible solutions: Making the labels ourselves, or using pre-made land-cover maps.

Making the labels ourselves meant looking at images in Google Earth Engine, and creating the labels using the GUI. We will refer to this method as *hand labeled* approach later in the report. *Forest*-labels were inserted in areas were there was forest in a particular year, otherwise *other* labels were used. The labels used this class information as well as a date information, which was information about when the particular pixel labeled was forest/other.

As an alternative to hand-labeling Another way to get the target labels was to use two maps: A forest/non-forest map from JAXA [10], and a tree canopy cover map from NASA [1]. We will refer to this method as *auto labeled* approach later in the report. To get a forest/non-forest map from the tree canopy cover data, A threshold was set: anything with canopy cover at or above that was considered as forest, and everything below as non-forest. Our labels were then sampled from areas where these two maps agreed. The labels were sometimes clearly wrong, so we had to be quite careful with this approach. The canopy cover threshold that produced the most accurate maps also varied by region. Choosing a proper tree canopy cover threshold was important. Even then, the resulting map was clearly wrong in some areas so it was important to carefully choose the area to sample the labels from.

2.4 Discussion

The idea to apply CCDC in Google Earth Engine came from our client. The primary rationale behind this was that Google Earth Engine and CCDC were previously familiar to the client, and that CCDC was the only viable model time series analysis algorithm available in Google Earth Engine at the time. Google Earth Engine doesn't afford much opportunity for implementing one's own algorithms, because the algorithms would have to be able to run well on their servers, the workings of which are highly opaque to users. It is also a proven algorithm for the problem. We believe that the use of such standard and familiar tooling will make it easier for the client to further tweak the system, and increase the probability that our system will actually be taken into use.

An alternative approach to producing the labels would have been to try to infer them from the land use data anyway. For example, we could have only used hexagons which were 100% forest or non-forest to avoid the issue of not knowing which pixels within a hexagon had which land use. Additionally, forestry practices in many countries are such that forest (land use) almost perfectly coincides with tree cover (land cover). The notable exceptions to this are boreal countries such as Finland, where clear-cutting is a common method of forest management, and palm oil producing countries where palm plantations can be difficult to distinguish from forests. However, we did have more detailed information for some times and places, and in those cases it could have been possible to reliably infer land cover from land use. We did not have time to implement this, but it could be a fruitful direction of further development for the client to pursue.

In addition to the method detailed above we explored some other approaches. The idea of training a regression model on a single pixel aggregating all the information of the pixel values of one hexagon was deemed an interesting approach. Due to time constraint the idea was partially followed through but was not completed due to software problems while interfacing with the Google Earth Engine platform.

3. Results

The primary deliverable results of the project are Google Earth Engine scripts used to train and use the CCDC-based land cover classifiers described in Chapter 2, as well as Python notebooks and application for analyzing and visualizing the results produced by the classifiers. These were accompanied by a detailed, illustrated usage guide, and descriptions and discussion of the performance of our classifiers in different countries.

3.1 Google Earth Engine scripts

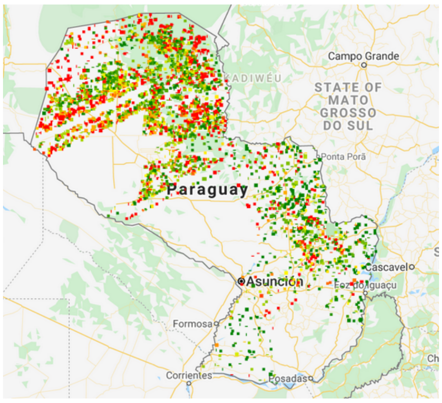
We delivered our client a collection of Google Earth Engine scripts that the client can either utilize to reproduce our results or to classify new areas. These scripts can also act as a basis for future development. The aforementioned tasks can be achieved by the following scripts: *AutoLabelsClassification*, *AutoLabelsTrainClassifier*, *AutoLabelsPre-TrainedClassifier*, *OwnLabelsClassification* and *ExampleCreateLabels*.

3.2 Classification results

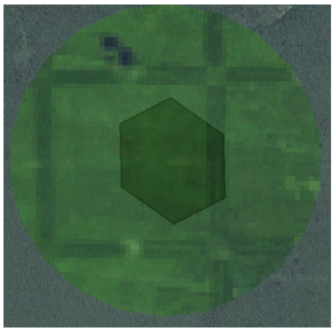
We evaluated the classification results using multiple metrics. Because the primary intended function of the system was to expedite future surveys by filtering out areas with no change, an important aspect of our evaluation was to view the task as one of deforestation detection. The model was considered to have detected deforestation in a hex if more than 5% of the hexagon was predicted to have lost forest. The threshold was motivated by rounding the predicted forest loss percentage to the nearest 10%, as in the visual interpretation data. We then used the precision and recall of this detector as the main metrics of the quality of our models. To unify these metrics also computed $F - \beta$ scores for the results with $\beta = 5$, and included functionality in our evaluation notebook for the user to supply their own β. The $F - \beta$ score provides the harmonic mean between precision and recall, with recall considered β times more important than precision. In this case, we believed that recall is significantly more important than precision for our client, because false positives can easily be corrected by the human experts, whereas false

Period	MAE	ρ	% Precision	% Recall	F-5.0	Avg. actual	Avg. predicted
2000-2010	24.481	0.481	76.579	61.446	0.619	32.761	13.664
2010-2018	20.761	0.449	57.937	69.896	0.693	24.653	16.727

Table 3.1: Perfomance metrics of our classifier for Paraguay. ρ means Pearson's correlation.



(a) Model performance in Paraguay. Red illustrates notable difference, yellow indicates some difference and green indicates that the prediction was aligned with the visual expert evaluation.



(b) Example hexagon illustrating the difference between land cover and land use.

negatives can not be corrected as easily.

3.2.1 Paraguay

We were able to get some results for all of the countries (Finland and Paraguay were nearly fully classified). The best results were from Paraguay. For Paraguay, we did not have all the human evaluation data that we compare our models performance against. However we had more than half of it, so some conclusions could be made about the results.

Figure 3.1a illustrates the difference between predicted vs human evaluation for forest % in a hexagon. Red stands for 30% or more difference.

It should be noted that a part of the difference between our results and the visual interpretation data would remain even if our classifier performed perfectly, due to the land cover / land use distinction discussed in Section 2.3. An example of this can be seen in Figure 3.1b, where darker green areas are trees. The hexagon contains some percentage of trees, and our classifier classifies them correctly. However a human had interpreted this hexagon as 0% forest, as the darker area is not a forest, but just an area with trees.

Year	MAE	ρ	% Precision	% Recall	F-5.0	Avg. actual	Avg. predicted
2000-2010	1.060	0.095	28.571	12.500	0.128	0.828	0.298
2010-2018	0.912	0.083	5.882	14.286	0.135	0.373	0.601

Table 3.2: Performance metrics for Finland for the auto labeled model. As there were only a few hexagons that had experienced deforestation according the ground truth data, the exact values or recall and precision are not to be trusted.

These issues make it clear that just staring at the differences between predicted values and human evaluation will not be enough. The model could still be improved by fine-tuning of the parameters but it already provides results that benefit our client.

3.2.2 Finland and other countries

Shapefile for Finland contained 776 Hexagons, scattered around the country. According the ground truth data, the majority of Hexagons in Finland had not experienced deforestation in the period from 2000 to 2018. For instance during 2010-2018, there were only 13 hexagons in which deforestation had occurred.

Neither auto labeled or hand labeled approach were successful in the classification task when measured against the ground truth data. See Table 3.2 for various performance metrics.

The results for areas in the countries such as Ethiopia, Tanzania, Bolivia and Indonesia were similar; the performance was not in a level of being useful.

3.3 Evaluation scripts

We developed Python scripts, operated through notebooks, for evaluating and analyzing the output of the classifiers trained in Google Earth Engine. These are available in a public Github repository.

The scripts are accompanied by scripts for cleaning the data we were provided by the client, and separating out a hold-out test set of 20%, obtained through reproducible stratified sampling.

3.3.1 Evaluation notebook

A Jupyter notebook was developed to combine the output of the classifiers with the visual interpretation data, and compute performance metrics. It was used for example to produce the tables in this report.

3.3.2 Visualizer app

Having observed not so good classification performance apart from Paraguay we decided to build an application that would allow the client to more easily inspect individual hexagons in terms of both our prediction, the ground truth data and other tools for analysing land use and land cover of each pixel. The application provides the following main functionalities:

- An overview of certain area in terms of forest loss measured against the ground truth data.
- An overview of certain area in terms of amount of forest measured against the ground truth data.
- Tools for inspecting an individual hexagon in terms of true color satellite image, false color satellite image with appropriate bands and JAXA classification.

The app is online and be accessed via browser. More details and screenshots from can be found in Appendix D.

3.3.3 Discussion

Despite our classifiers failing in other countries than Paraguay our efforts provided us and our client important insights. Firstly, the very different results confirm that countries are indeed different in terms of classification difficulty. According the FAO's remote sensing expert, Paraguay is easier than other countries due to very low cloud coverage and other factors.

Secondly, by visualizing individual hexagons with the visualizer app we implemented as a one deliverable for this project, we recognized that the visual interpretation data might not be as accurate as has previously been assumed. We found several hexagons where the annotated amount of forest seemed to contradict with both true color and false color images, which can be utilized to evaluate the amount of forest in a given area. See appendix D for an example of one such hexagon. However, more detailed investigation with forest remote sensing expert knowledge would be needed to confirm whether the conclusion is justified.

Based on our experience we suggest an expert in the forest remote sensing to be involved hands-on in the selection for training area and adjusting the tree canopy level threshold for *auto labeled* approach or in the labeling process in the *hand labeled* approach.

More importantly we propose that in the future, the visual interpreters would describe the area in terms of *land cover* as as well as *land use*, and did so at the pixel level. This would result in a dataset that would be much more useful for machine learning applications.

4. Conclusion

The aim of our project was to develop ways of reducing the workload of FAO visual interpreters in their effort to assess areas where changes in land cover takes place. To evaluate the performance of the to be developed machine learning model we received ground truth data from areas in seven different countries.

Due to potential problems with the ground truth data in terms of its suitability for model training we acquired the training data in two alternative ways: by labeling points by hand and by utilizing a combination of existing land cover datasets. We developed Google Earth Engine scripts to train a random forest classifier for labeling a single pixel in landsat7 image collection with label *forest* or *other*. By comparing the amount of forest in specific years our system was then able to output the areas where deforestation occurs, to the precision of single pixels. To mitigate challenges caused by clouds and seasonal changes the classifier used CCDC coefficients as input features.

We trained separate models for each of the seven countries. Paraguay was the only country in which the performance was at the level that enables the classifier to be utilized as such. Results from the other countries provided important insights in terms of how to develop better machine learning models in the future. We suggest that forest remote sensing experts should be involved if *hand labeled* approach is to be used. Similarly, expert supervision is recommended in parameter tuning and training area selection, if pre-existing land cover maps are to be used. We also suggest experimenting with alternative approaches where the current ground truth data would actually be used to train the model, despite the potential problems.

More importantly, we propose that in the future, the data items collected from visual interpreters would be selected with the aim of using the data items to train a pixel-level machine learning model for classifying *land cover* in a more straight-forward manner.

In addition to the GEE scripts that allow the client to continue the work for automated land cover detection, we have provided easy-to-use evaluation scripts and a visualizer app to support both model development and visualization purposes.

References

[1]

[2] Near infrared imaging. https://publiclab.org/wiki/near-infrared-imaging. Accessed: 2013-05-18.

[3] D. K. Bolton, J. M. Gray, E. K. Melaas, M. Moon, L. Eklundh, and M. A. Friedl. Continental-scale land surface phenology from harmonized landsat 8 and sentinel-2 imagery. *Remote Sensing of Environment*, 240:111685, 2020.

[4] T. N. Carlson and D. A. Ripley. On the relation between ndvi, fractional vegetation cover, and leaf area index. *Remote sensing of Environment*, 62(3):241–252, 1997.

[5] Food and A. O. of the United Nations (FAO). Global forest resources assessment 2020: Main report, 2020.

[6] N. Gorelick, M. Hancher, M. Dixon, S. Ilyushchenko, D. Thau, and R. Moore. Google earth engine: Planetary-scale geospatial analysis for everyone. *Remote sensing of Environment*, 202:18–27, 2017.

[7] J. R. Irons, J. L. Dwyer, and J. A. Barsi. The next landsat satellite: The landsat data continuity mission. *Remote Sensing of Environment*, 122:11–21, 2012.

[8] E. Kairu. An introduction to remote sensing. *GeoJournal*, 6(3):251–260, 1982.

[9] F. a. A. O. of the United Nations. Methodology | Global Forest Resources Assessments. http://www.fao.org/forest-resources-assessment/remote-sensing/fra-2020-remote-sensing-survey/methodology/en/. Retrieved 2021-05-18.

[10] M. Shimada, T. Itoh, T. Motooka, M. Watanabe, T. Shiraishi, R. Thapa, and R. Lucas. New global forest/non-forest maps from ALOS PALSAR data (2007–2010). *Remote Sensing of Environment*, 155:13–31, 2014.

[11] Z. Zhu and C. E. Woodcock. Continuous change detection and classification of land cover using all available landsat data. *Remote sensing of Environment*, 144:152–171, 2014.

Appendix A. Introduction to remote sensing

A.1 Remote Sensing

This project is possible thanks to the information gathered with Remote Sensing Techniques.

Remote sensing is the process of detecting and monitoring the physical characteristics of an area by measuring its reflected and emitted radiation at a distance (typically from satellite or aircraft) [8]. Special cameras collect remotely sensed images, which help researchers "sense" things about the Earth.

Satellites measure the amount of electromagnetic radiation (EMR) that is reflected or emitted from the Earthâs surface. These sensors, known as multi-spectral sensors, simultaneously measure data in multiple regions of the electromagnetic spectrum. The range of wavelengths measured by a sensor is known as a band and are commonly described by the name (Red or Near-IR for example) and the wavelength of the energy being recorded [8].

The near infrared part of the spectrum is especially important for our task because healthy plants reflect it as the water in their leaves scatters the wavelengths back into the sky. By comparing it with other bands, we get indexes like NDVI, which let us measure plant health more precisely than if we only looked at visible greenness [4].

In our particular case we leveraged Landsat Earth surface images. Landsat is a joint program of the USGS and NASA that has been observing the Earth continuously from 1972 through the present day [7]. Today the Landsat satellites image the entire Earth's surface at a 30-meter resolution about once every two weeks, including multi-spectral and thermal data [3].

By analyzing the image time series we expect to gain information on the past and current vegetation.

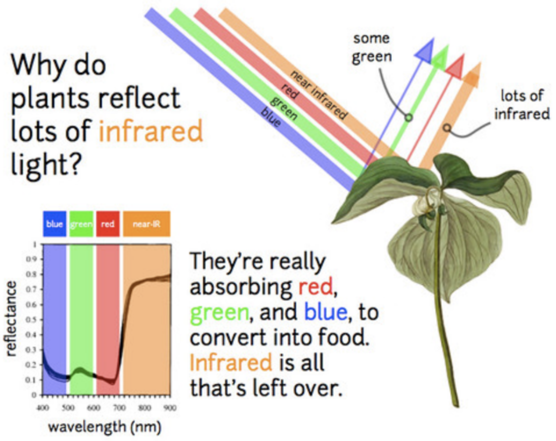


Figure A.1: Different wavelengths are absorbed to different degrees by plants. Image from [2].

Appendix B. Details of technical approach

B.1 Change detection

The CCDC algorithm first masks the cloudy pixels of each image, by looking at the pixel values of the short-wave infrared and green bands (bands 7 and 2). After this, it creates a time-series model for each pixel of each band, and finds the breakpoints (points where the land cover class has changed). B.1 Is the equation that the model tries to fit, the x represent the temporal variable, T is a constant equal to 365 in our case (year length in days). The model takes into consideration the seasonal changes as well as the general trend. The initial model is built by using the first pixel values (in the paper, 12 was used).

Ordinary least squares is to fit this initial model. After creating the initial model, each new observation of a pixel value is compared to a prediction made by the model. If the observed pixel value is within a given threshold, there is no change in land cover. If the observed pixel value is outside the threshold, a possible breakpoint has been found. The threshold used is $\pm 3 \times RMSE$, where RMSE is the Root Mean Squared Error of the time series model.

$$\rho(x) = a_0 + a_1 cos(\frac{2\pi}{T}x) + b_1 sin(\frac{2\pi}{T}x) + c_1 x \tag{B.1}$$

If new observations appear outside the threshold for n consecutive observations, a breakpoint has been found, meaning the land cover class has changed in that particular pixel. In the paper the algorithm was introduced, 3 consecutive observations had to be outside the threshold for a land cover change to be accepted. Once all the images have been processed, the algorithm returns a an image which contains the coefficients of the time series models of each pixel of each band. In Google Earth Engine it is possible to define which bands of an image to use to detect breakpoints. We decided to use the NDVI-band of an image. NDVI can be calculated from the formula $NDVI = \frac{NIR-Red}{NIR+Red}$ and added to an image as a separate band. In the formula NIR stands for the Near Infrared band. We also had to include the short-wave infrared and green bands for breakpoint detection, because we used those to mask clouds, and the algorithm expects that the cloud detection bands are included in the breakpoint detection bands.

It is unclear how much the algorithm has changed after its introduction. For example in Google Earth Engine a new observation has to be outside the threshold 6 consecutive times by default (was 3 in the paper). The algorithm also returns more coefficients than what was mentioned in the paper.

B.2 Classification

For every classification area, we selected all the images of that area from year 1985 to year 2018. We filtered out the most cloudy images, and used these images as input to the Continuous Change Detection and Classification algorithm (CCDC) [11].

The coefficients of the models are used as input to a random forest classifier.

For the classification, we used a JavaScript class shared with us by the client to store the CCDC results. The class has a lot of helper functions to visualize, classify, and recreate images for any given time with the coefficients.

The classifier is trained at an area where the labels are from. It is then used to classify a subset of hexagons. The classification is done for each segment of a pixel's time series model. Done this way, we get land cover classification for every year. This is helpful later when we calculate the deforestation and reforestation of the areas. Theoretically every hexagon could have been classified, however Google Earth Engine had memory issues when classifying too large an area. This is why we classified only subsets. When selecting the training and classification area, we had to keep in mind that the vegetation should be pretty similar in both areas for the classifier to work well enough. The training area had to have diverse enough vegetation for the classifier to work well enough in the much larger classification area. If the classification area had places with water, farmland, forests and roads, an ideal training area would contain these types of vegetation as well.

The result is a csv that contains information about the hexagons classified: deforestation percentages for time periods 2000-2010 and 2010-2018, as well as information about forest percentages per hexagon.

Appendix C. More results from Finland

The distribution of forest loss percentage is depicted in Figure C.1

We classified 775 of the hexagons using both hand labeled approach and auto labeled approach. One hexagon was left unclassified.

The hand labeled approach was performed by training a random forest classifier with 53 positive and negative hand-labeled samples, in total 106 samples. We experimented with multiple different sets of labels. The auto labeled approach used tree canopy cover of 40 %.

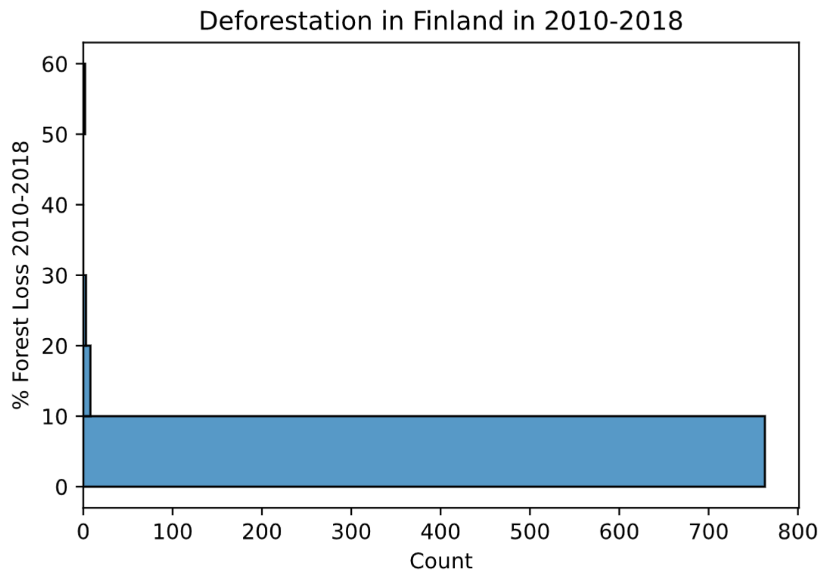


Figure C.1: Most hexagons in Finland in 2010-2018 did not experience deforestation according the ground truth data.

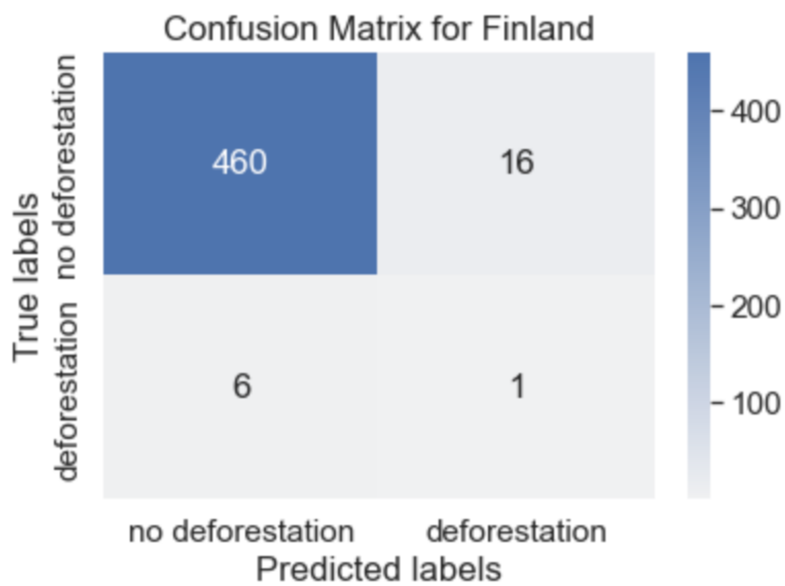


Figure C.2: Confusion matrix illustrating how the auto labeled model compared against ground truth data in 2010-2018.

Appendix D. Visualizer application

The application was build with Python, Jupyter notebooks and Voila. It is available at https://hexa-visualizer.herokuapp.com/. Be noted though, that as it is deployed with free plan in a service called Heroku, it will take time to load fully. It is only fully loaded when it has displayed interactive maps. It's also worth to mention that the deployed app only contains data from Finland. The recommended way of using the app is to use a version that is installed locally as it can then be easily configured to use data from any of the countries explored in this project.

The visualizer app is designed to with the following workflow. First one can use interactive maps to select the area of interest. One can use either a map displaying the hexagons and color coded result against the ground truth data in terms of either forest loss or forest vs. non-forest.

By clicking a button, user is then able to see a list of all the hexagons in the bounds of the current view on the map. For each hexagon all the essential data is displayed, such as the plot id identifying the hexagon, forest loss predictions, forest loss ground truth data, forest percentage predictions, forest percentage ground truth and the forest subcategory indicating whether the hexagon centroid is unstocked forest.

User can then drill down to individual hexagon and display various layers on top of the interactive map. These layers enables the user to for instance inspect the false color image of the area, constructed via bands B4, B5 and B3 which were recommended to be used by a remote sensing domain expert. In this layer, the dark red indicates a high probability for forest while other shades of red and green indicates high probability for non-forest.

An example can been seen In Figure D.1 and Figure D.2, which display an image of the same hexagon. In our ground truth data, this area is classified as being 100% forest. However, both the satellite image and the false color image suggest that this might not be the case. Although we don't know the dates of the images used to produce the satellite image, the date of the false color image matches the date of the ground truth data. We also know that the land use of the hexagon centroid is not unstocked forest. However, as we don't know the land use for the whole hexagon we cannot draw conclusions of whether the ground truth data could be actually somewhat incorrect. Our model in turn predicted this hexagon to be only 30% forest.

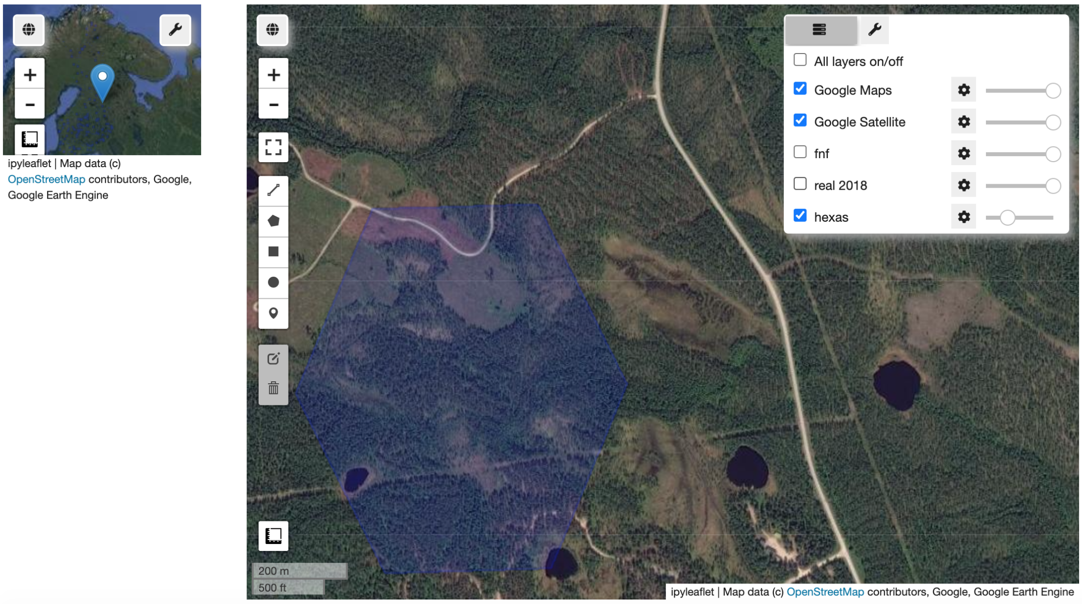


Figure D.1: Inspecting individual hexagon with the Visualizer app. On top left corner, a small map displays the hexagon location with low zoom value. On the right, a large map displaying satellite image of the hexagon and the surrounding area. A display is open for interactively selecting which layers are visible. The hexagon is displayed with a faint blue overview. Details of the hexagons are available in the app but did not fit in the image.

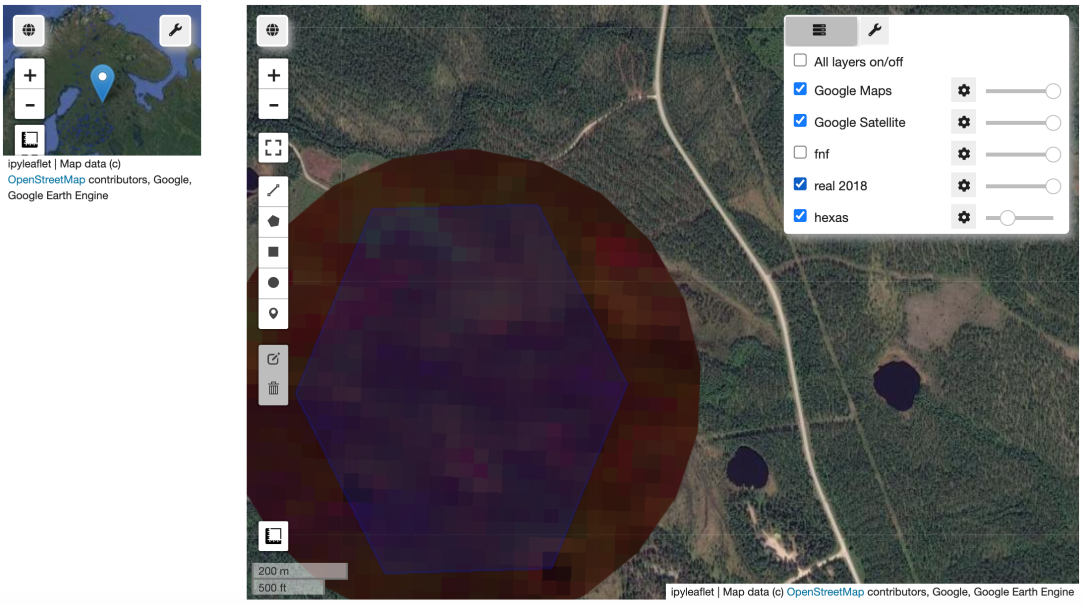


Figure D.2: Inspecting individual hexagon with the Visualizer app. The false color layer is visible, displaying forest as dark red and non forest with lighter shades of red or green. User is able to deactivate hexagon, to see more clearly the different shades of color in the image.

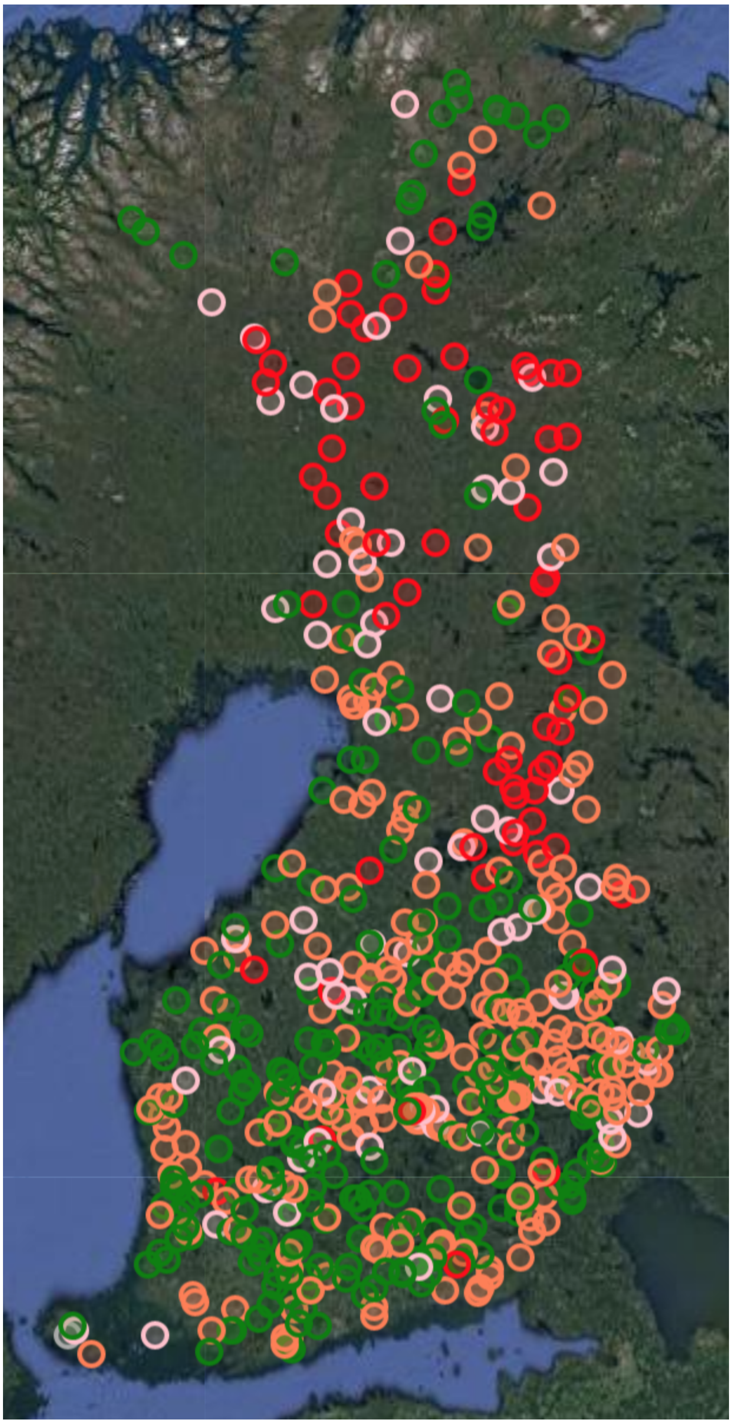


Figure D.3: Forest vs non-forest in Finland in 2018, as illustrated by the Visualizer app. Each circle represents a single hexagon. The green colors indicates that the difference between the predicted percentage of forest and ground truth data is between 0 and 10, the orange color indicates the difference is between 10 and 30, the pink indicates that the difference is between 30-50 and the red indicates the the difference is more than 50.