

T.C.F. Sassen

# Comparing WWII aerial photographs to Sentinel-2 data





# Comparing WWII aerial photographs to Sentinel-2 data

By

**T.C.F. Sassen**

in partial fulfilment of the requirements for the degree of

**Master of Science**  
in Applied Geology

at the Delft University of Technology,  
to be defended publicly on 26<sup>th</sup> of April, 2018 at 2 PM.

Supervisor:	Dr. R.C. Lindenbergh	TU Delft
Thesis committee:	Drs. B. Kreuger	Kreuger in Cultuur
	Dr. L. Lannini	TU Delft



# Preface

You are about to read the additional thesis: comparing WWII aerial photographs to Sentinel-2 data. I did this thesis as a part of my Master Degree Geoscience and Remote Sensing at the Delft University of Technology. I chose this subject because I am interested in satellite imagery and the classification and interpretation of this data. This combined with the remote location and historical importance made this a very interesting project and I enjoyed working on it.

*T.C.F. Sassen  
Delft, February 2018*



# Contents

Preface .....	3
Abstract.....	8
Acknowledgements.....	8
1 Introduction .....	9
2 Region of interest.....	10
2.1 Region of interest .....	10
2.2 History of West Papua .....	11
2.3 Papua during WWII .....	12
2.4 West Papuan Geography .....	12
3 Datasets & Software .....	13
3.1 Datasets .....	13
3.1.1 Photographs .....	13
3.1.2 Sentinel-2 .....	14
3.1.3 Landsat.....	15
3.2 Software .....	16
3.2.1 Python .....	16
3.2.2 QGIS .....	16
3.2.3 Google Earth Engine .....	16
3.2.4 Panorama Maker .....	17
3.2.5 Insufficient Software .....	17
4 Method.....	18
4.1 Classification methods.....	18
4.2.1 Stitching .....	24
4.2.2 Georeferencing .....	25
4.2.3 Classification.....	28
4.3 Landsat and Sentinel.....	34
4.3.1 Cloud filtering .....	34
4.3.2 Classification.....	37
5 Results.....	41
5.1 Panorama.....	41
5.2 Landsat and Sentinel.....	44
6 Method of validation .....	46
6.1 Panorama.....	46
6.2 Landsat and Sentinel.....	49
7 Conclusion .....	51
Bibliography .....	52

## Abstract

During World War II West Papua was the scene of war between the Axis and the Allies. In this case, it was the Japanese on one side Australian and US forces on the other. After the war a lot of equipment was left in the jungle and because of the low population density on the island some material can still be found to this day. Planes and vehicles that were abandoned far enough from villages were able to withstand the test of time and are still hidden under the trees.

In November 2018 an expedition force is heading to the island to find some of these planes. This assignment was meant as a preparation for that expedition and it was my job to find places that could be interesting to investigate closer in the field. The research question that had to be answered was as follows: is it possible to classify and link historical data to modern data.

To do this I looked at old aerial photographs of the airports, and at modern Sentinel-2 and Landsat multispectral images of the area. The aerial photographs are stitched into a panorama and geolocated so that the coordinates were known. Next features like the average and gradient of surrounding pixels are computed so the image could be classified. Classifications were then performed on the panoramas and on the Landsat 8 dataset in order to create a map of the region of interest and possibly point out potential war equipment locations. The Landsat data was also used to create a heatmap to show the warmest spots in the region. These spots could potentially point to material that is warmer than its surroundings, e.g. a metal car in a grass field that has been standing in the sun for some time.

The classification of the panoramas proved to be a difficult task and the final result leaves much to be desired. The classification of the Landsat 8 dataset went better and the results show a useable map. However the most interesting class in this case was the war equipment class, which cannot be validated. The heatmap could mostly be used to highlight the roads, but there are certain spots that light up and cannot be explained without field data. These spots might be interesting to check out during the expedition.

## Acknowledgements

I would like to thank Roderik for supervising the research and guiding me through the process. Furthermore I would like to thank Bas and Lorenzo for being on the committee and proofreading the thesis. Lastly I would like to thank Shannan for reading my thesis many times and improving my English where possible.



# 1 Introduction

In 1940 the Germans invaded the Netherlands and the country was occupied. Two years later the Japanese followed suit with the Netherlands East Indies and New Guinea on their way to Australia. From that point onward, Netherlands New Guinea, now West Papua became a battleground between the Japanese and Australian and US forces. To stop the Japanese, the Americans launched exploration and bombing flights, during which they would photograph the area. These photographs were used to study the airports and plan future attacks and offensives. After the war, the photographs were stored in archives where they are still available to this day.

In addition to these pictures, some of the equipment survived the past seventy years. Airport equipment and crashed planes can still be found to this day in the exact location they were left. These machines survived because the island is very sparsely populated. If a plane crashed near a village, the residents would strip it and recycle the material, however, since there are no real villages in our area of interest, most of what crashed or is left behind can still be found in a state as can be seen in Figure 1.

As part of an Additional Thesis I will be looking at this area and trying to create a map showing a classification of the area and the possible location of military equipment. The question I will try to answer is if it is possible to classify and link historical data to modern data. The results of this thesis will then be used in an expedition that will take place in the November 2018: with the use of a restored Bell helicopter the team will fly over the area to look for plane wrecks. Although the team will do their best, it would make the search easier if they knew what areas are more interesting before they started.

The datasets that are available for this project are the aerial photographs, and the multispectral satellite datasets Landsat and Sentinel-2. Each of these datasets has its own set of benefits and problems, but a recurring problem is the clouds: West Papua is located near the equator and is therefore always at least partially covered in clouds. Although some datasets manage to deal with these clouds better than others, but in the end the problem will still need to be addressed.



**Figure 1** A second world war military truck on site

# 2 Region of interest

## 2.1 Region of interest

North of Australia lies the island New Guinea, which is split into two parts. On the east part the independent country Papua New Guinea is found while the western part belongs to the Republic of Indonesia. The western part is again split into two separate provinces: Papua and West Papua, as can be seen in Figure 2.

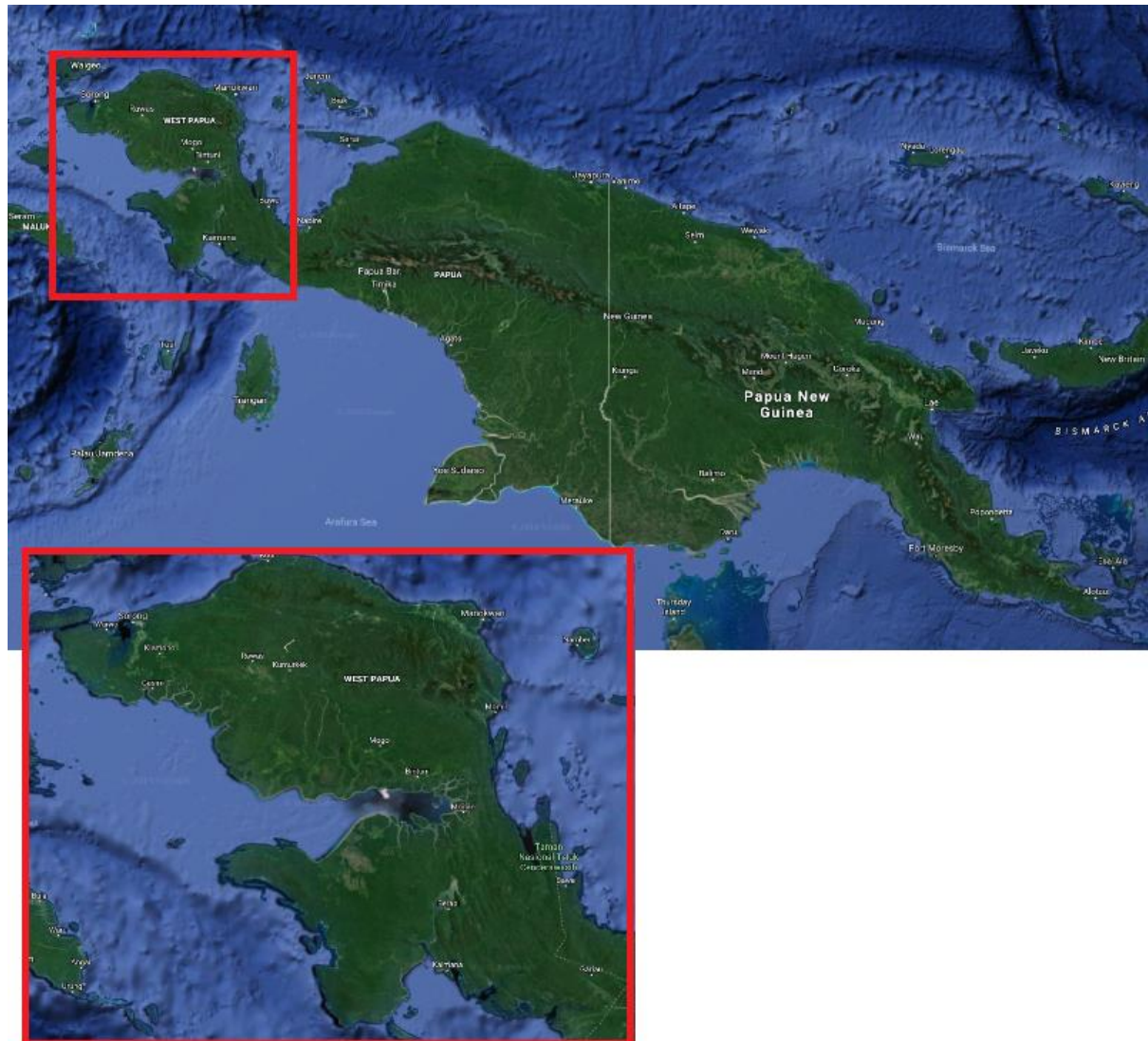
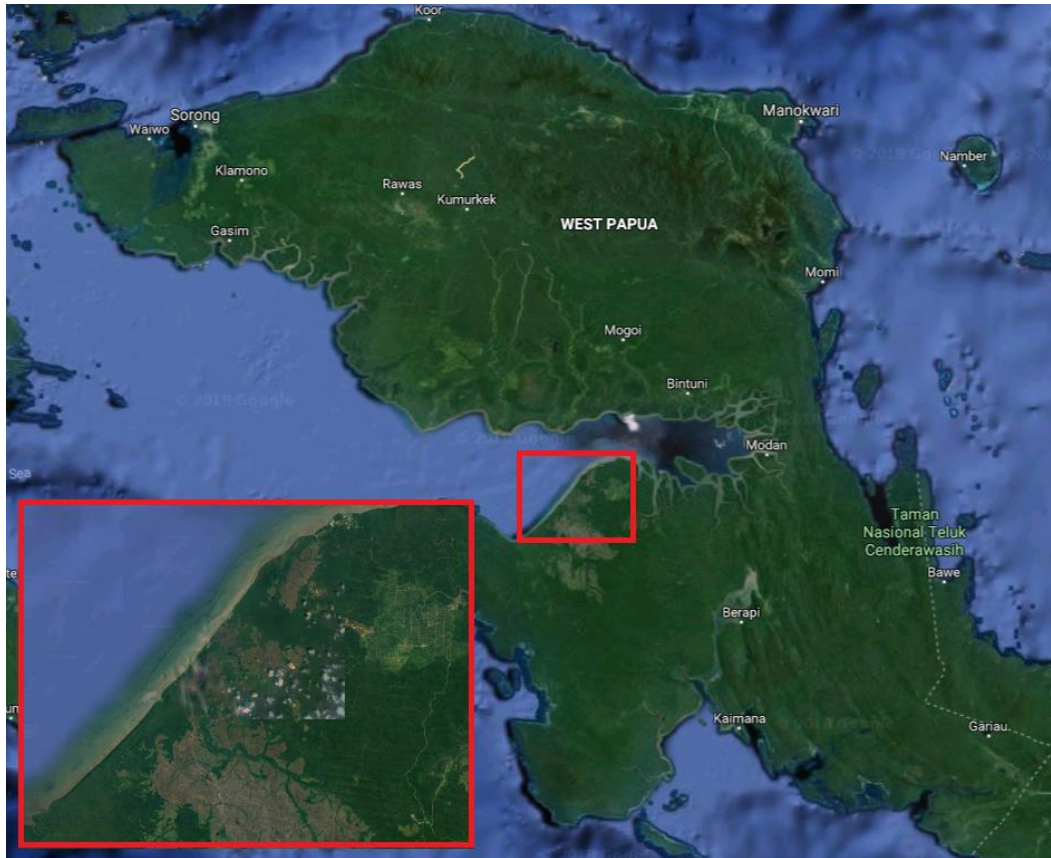


Figure 2 The Island New Guinea and the province West Papua. Source: Google Maps

In Figure 3 West Papua is shown. The province has a northern and a southern part. The region of interest lies on the coast in the north of the southern part as shown in Figure 3. The area is very sparsely populated, but there is a large palm oil plantation. The area contains four airports.



**Figure 3 Region of interest**

## 2.2 History of West Papua

The island was originally populated by Melanesians tens of thousands of years ago. These people had little to no contact with the western world up until their colonization together with what we now call Indonesia by the Dutch in 1898. In 1949 the Republic of Indonesia became independent, but Papua did not become part of Indonesia, remaining a Dutch colony. In 1950 the Dutch started to prepare Papua for their own independence and in 1961 a Congress was held at which the people declared independence and raised their new flag.

After a few months the Indonesian army invaded Papua in order to take control over all of the former Dutch colonies and a conflict broke out between the Dutch, Indonesians and the Papuans. Indonesia turned to the Soviet Union for help. The US government feared an increase in the spread of communism in Asia and wrote to the Dutch Prime Minister asking them to cede Papua to Indonesia. The US government organized a meeting between The Netherlands and Indonesia, and in 1962 control over Papua was given to the United Nations, and a year later to Indonesia. The people of Papua had no say in this, but they were promised the right to self-determination.

In 1969 resistance to Indonesian rule grew, because the Indonesian army had killed thousands of Papuans in the past seven years. They turned to the United Nations and a ballot for independence was organized. The Indonesian military declared the Papuans as too primitive to cope with democracy and therefore handpicked 1.026 representatives out of a population of around one million. These representatives were bribed and threatened and in the end they all voted for Indonesian rule despite widespread opposition. The United Nations did not interfere in the rigged ballot and Papua is still under Indonesian rule to this day. (Free West Papua Campaign, n.d.) (Cordell, 2013)

### 2.3 Papua during WWII

From November 1941 until April 1942 Japan invades Dutch New Guinea and moves towards Australia. Due to the lack of battalions of the Royal Dutch Army in the East Indies, the Japanese are able to take over the country in record time. They plan to go from the north of Dutch New Guinea on to Australia, but never got further than Frederik Hendrik Island. Because the very low population density in the south, the Japanese leave this area unoccupied and the Dutch flag continues to fly there.

Both the Japanese and Australian army then build their basecamp on the east side of the island divided by the steep Owen Stanley mountain range. In January 1943 the American Army lands on the island and launches a major attack. The whole operation to recapture New Guinea takes well into October 1944 and the Allied Forces need to fight off the Japanese on 2.000 kilometers of fortified coast line. In the end 13.000 Japanese, 2.100 Australian and 2.000 American soldiers did not survive the battles. (Stichting Papua Erfgoed, n.d.)

### 2.4 West Papuan Geography

West Papua is located close to the equator and therefore has a tropical climate. This means that a large part of the area is covered in forest. On the coast different types of palm trees and swamps dominate the area, while in the wet lowlands the Barringtonia of the Brazil nut family grows better. Since the late 20<sup>th</sup> century, deforestation is claiming a large part of the trees, as they are cut for commercial logging or to make space for palm oil plantations. As can be seen in Figure 4, West Papua has mountains that are well over 2.000 meters high. The main chain, however, does not cross over to the peninsula we are interested in. In our region of interest only one small hill of around two hundred meters in elevation is found in the south west.

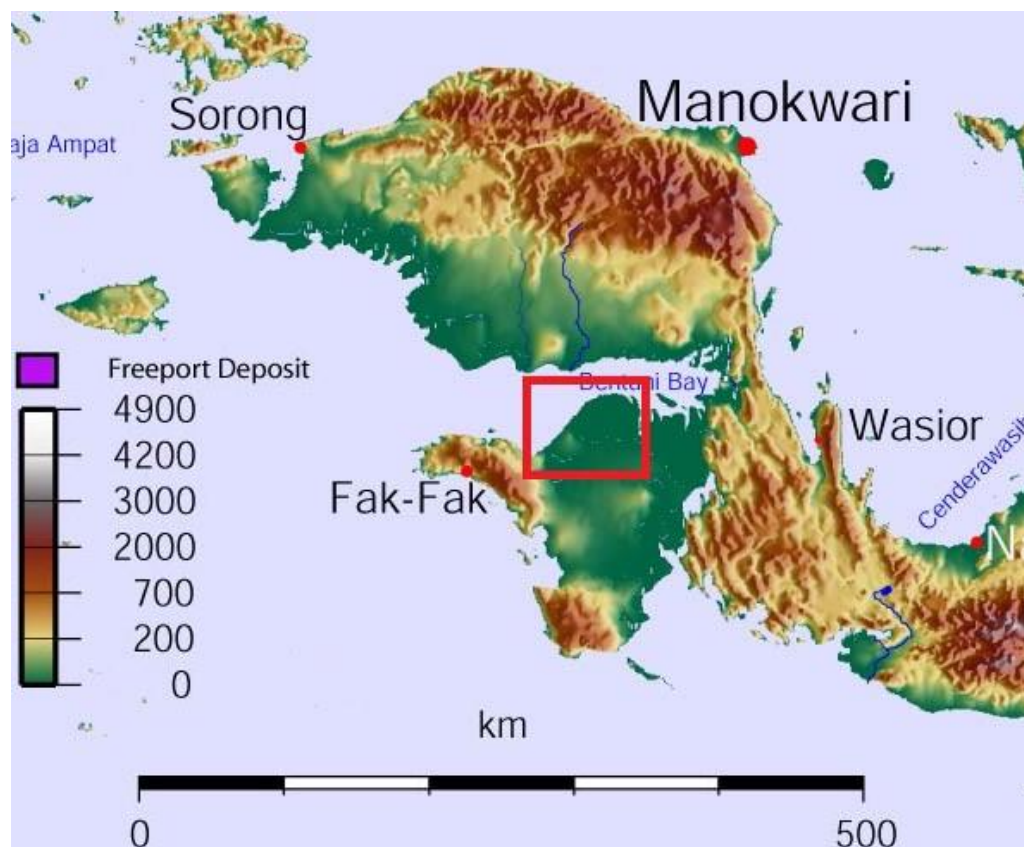


Figure 4 Elevation map West Papua(province) showing the region of interest (Faymer, 2007)

# 3 Datasets & Software

## 3.1 Datasets

For this project three different datasets were available:

- Aerial photographs from World War II
- Sentinel-2 images
- Landsat images

All three datasets have different bands and timings.

### 3.1.1 Photographs

The photographs used are aerial photographs taken by the American Army while flying over the area to either bomb the area or do reconnaissance flights, as can be seen in Figure 5. Typical cameras used to capture these pictures are the Fairchild K20 and the Kodak K24. There are four strips of photographs available for this project, but there exist archives that cover a larger part of the island. The number of pictures in a strip varied and can be seen in Table 1. The pictures are highly detailed compared to other datasets, and assumed to be orthophotos that are not georeferenced. This would mean that the scale is uniform and there is no distortion in the image, but the exact location of pixels is unknown. The original photographs are digitized to a 3600 by 3600 pixel image. This roughly corresponds to a resolution of around seventy centimeters in the field, but depends on the altitude at which the pictures are taken and therefore differs. The resolution is found by simply measuring a pixel in the georeferenced map in QGIS. A big problem with the dataset is clouds, which can be seen in Figure 5. This is something that cannot be solved since only one dataset per area is available. We will therefore have to accept that some areas will not be covered by this dataset. The pictures are in grey and contain no other information than the light intensity. To classify the images extra features are created from this intensity.

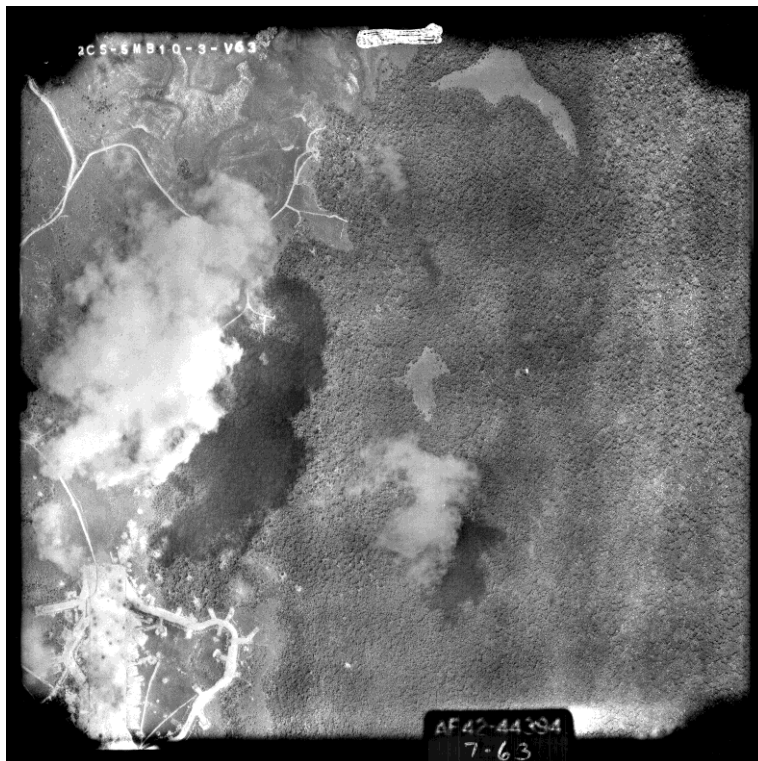


Figure 5 Original aerial photograph used in this project

Strip	Number of pictures
1	5
2	9
3	5
4	4

**Table 1** Number of available pictures per strip

### 3.1.2 Sentinel-2

Sentinel-2 is a satellite mission from the European Space Agency. It delivers multispectral images that cover the entire earth, with a revisit time of ten days with one satellite and five days with the two satellites. The dataset is publicly available online, but in this project the edited version from Google Earth Engine is used. This version does not contain all the bands from the original dataset, but is easily available, can be processed online, and can be processed and clipped before downloading. This saves a lot of time and does not require the end user to have a powerful computer, since processing is done in the cloud. The bands that are available in the Google Earth Engine are shown in Table 2. The available data is from the twenty-third of January 2015 to a few days before processing and contains 232 separate datasets. These datasets are then merged and processed so that only one large set is left. (Google, n.d.) (ESA, n.d.)

Band	Description	Resolution [m]	Wavelength [nm]
B1	Aerosols	60	443
B2	Blue	10	490
B3	Green	10	560
B4	Red	10	665
B5	Red Edge 1	20	705
B6	Red Edge 2	20	740
B7	Red Edge 3	20	783
B8	NIR	10	842
B8A	Red Edge 4	20	865
B9	Water Vapor	60	950
B10	Cirrus	60	1375
B11	SWIR 1	20	1610
B12	SWIR 2	20	2190
QA10	Empty	10	-
QA20	Empty	20	-
QA60	Cloud mask	60	-

**Table 2** Sentinel data description

### 3.1.3 Landsat

Landsat 8 is a NASA project and just like Sentinel-2 it is a multispectral dataset. Since the eleventh of February 2013 this project delivers a global coverage with revisiting time of sixteen days. In total 259 Landsat 8 datasets are used in the project, which are merged and processed into one large set just like is done for the Sentinel-2 data. As with Sentinel-2, not all bands are available in the Google Earth Engine. The available bands are shown in Table 3. (Google, n.d.) (NASA, n.d.)

Band	Description	Resolution in meters	Wavelength [nm]
B1	Ultra-Blue	30	435-451
B2	Blue	30	452-512
B3	Green	30	533-590
B4	Red	30	636-673
B5	Near Infrared	30	851-879
B6	Shortwave Infrared 1	30	1566-1651
B7	Shortwave Infrared 2	30	2107-2294
B10	Brightness temperature in °K	100→30	1060-1119
B11	Brightness temperature in °K	100→30	1150-1251
Sr_aerosol	Aerosol attributes	30	-
Pixel_qa	Pixel quality attributes	30	-
Randsat_qa	Radiometric saturation	30	-

**Table 3 Landsat data description**

## 3.2 Software

For this project different programs were used in the process.

### 3.2.1 Python

Python is an easy-to-use open source programming language. The learning curve for Python is not as steep as for some more advanced programming languages. Python is easy to use and, therefore, the actual writing of code will be faster. The running of code will generally take a bit longer compared to for example C++, but since the code for this project is not meant to run for hours and is only intended for this project, the time saved during writing is more than the time lost in running. In this project, Python is used for creating features from the greyscale panoramas, applying K-means to the new feature set and exporting the new images back to the Geotiff format so it can again be used by other programs. (<https://www.python.org/>)

### 3.2.2 QGIS

QGIS is a free and open source geographic information system. In this case, programs like ArcGIS would have worked equally well, but QGIS was chosen because it is available for free. In this project it is used for creating maps that display the area and the different bands for both the panorama pictures and Landsat and Sentinel-2 datasets. It is also used to georeference and classify the panoramas.

QGIS has an extended plugin library of which the Semi-Automatic Classification Plugin and the OpenLayers Plugin were used in this case. The Semi-Automatic Classification Plugin is used to classify the panoramas based on manually selected training data fields. The OpenLayers Plugin is used to apply a Google Maps background to some of the maps to see how successful the georeferencing was, and where the panoramas are located in the area. The OpenLayers Plugin has known issues with printing and is not recommended to use in the actual making of maps, but with some manipulation it was possible to create some maps out of the data. (<https://www.qgis.org/>)

### 3.2.3 Google Earth Engine

Google Earth Engine is a computing platform that allows users to run geospatial analysis on Google's infrastructure. The platform has the tools to download and process multiple geoinformation datasets. Since the tool is online and has all datasets available in the cloud, it is much faster than downloading everything and processing it on a personal computer. The Google Earth Engine made it possible to process years of Landsat and Sentinel-2 data in seconds and only do the visualization and map making part on the computer. The platform uses JavaScript. Personally I have never worked with JavaScript before, but a lot of examples were available in either the code editor itself or in the forum and blog, which made it very easy to get started with the engine. The only thing that could be improved on the platform is the use of different cloud suppliers or the ability to download without the use of a cloud platform. Currently, the Google Earth Engine only supports exporting to Google Drive, or storing it on the Google Earth Engine platform itself. (<https://earthengine.google.com/>)



### 3.2.4 Panorama Maker

ArcSoft's Panorama Maker 6 is a tool that can be used to stitch photos together to create a panorama. I have tried five different programs to achieve this for the old aerial photographs, but this is the only one that produced a good result and was available for free or with a free trial period. The interface is simple and there are not a lot of settings you can change, but the result is sufficient and it does not look like there is a lot of distortion on the overlapping zones. The software also allows the user to manually pick matching points to improve stitching if necessary. Since it is closed source and no proper documentation can be found it is unclear what algorithms are used in the various steps of the program. The resulting panoramas seem appropriate for the project, but it is unclear if corrections are applied for for example lens distortion or errors in manual matching of the images. This means the results should always be examined carefully before proceeding to the next step. The software had a free trial, but is currently not on ArcSofts site anymore. (<http://www.arcsoft.com/panorama-maker/>)

### 3.2.5 Insufficient Software

During the project, I attempted to use several different programs to accomplish the same task. In the end only one program works best for the given task. The following software was tried but found insufficient for the project.

Agisoft Photoscan is a program that performs photogrammetric processing of digital images to create point clouds. The problem with this software was that it did not find matches between all photos in a strip, but generally just three or four pictures. It is very complicated and time consuming to manually match the extra pictures. The output of the process is a point cloud, that on the highest detail is still not as good as the original photographs, which means data loss. (<http://www.agisoft.com/>)

Photostitcher is a program specifically designed for the stitching of photos. The problems that I found with this program were that there were no settings to improve the stitching and, due to memory issues, the program would not complete a full strip of stitching. (<https://www.photostitcher.com/>)

Panoweaver is a photo stitching program made by Easypano. It is easy to use and had a lot of settings to improve the image. The resulting output image was looking good, but the free version added multiple watermarks over the panorama. To remove the watermarks the full version had to be bought. I did not use the program here, but I do think it is a proper solution for the stitching of the remaining strips. (<http://www.easypano.com/download-panorama-software.html>)

Pix4d is much like Agisoft a program that can be used to create a 3D point cloud out of overlapping images. The output was better than from Agisoft, but the same problems were there. Manually stitching the remaining photographs was very difficult and the resulting 3D point cloud was not as sharp as the original pictures. (<https://pix4d.com/>)

# 4 Method

## 4.1 Classification methods

In this project multiple supervised and unsupervised classification algorithms are used. For supervised classification methods the user has to manually input trainings data to train the algorithm, while unsupervised algorithms work without any input.

### K-means

K-means is an unsupervised classification method. It will fit a given number of clusters through a dataset. This is done by first initializing K centroids and computing the distances of all points to these centroids. A point will belong to the closest centroid and so clusters are formed. Next the mean of these clusters is computed and the distances to these new means are calculated. Based on these distances new clusters are formed and the centroids will be moved again. This will be repeated a few times until the clusters are stable.

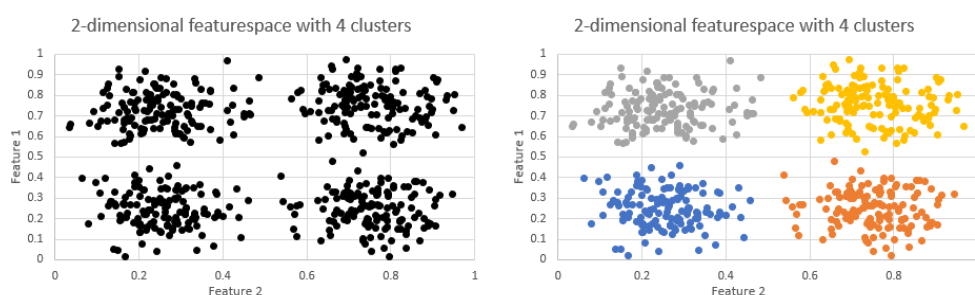


Figure 6 K-means explained

In Figure 6 an example 2D dataset is given. If in this dataset four clusters are used the K-means result will be as shown on the right. If however three or five cluster are fitted the algorithm will still give an output, but the clusters are not as we want them to be. It is therefore important that for K-means an appropriate number of clusters is chosen based on the dataset.

### Minimum distance

Minimum distance unlike the K-means is a supervised classification method. In the minimum distance method, an n-dimensional feature space is created where n is the number of features we created. The training data pixels are points in this space and the distance to these pixels per pixel of the panorama is computed. The distance in an n-dimensional feature space is calculated using the following formula:

$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_i - q_i)^2 + \dots + (p_n - q_n)^2}$$

Where p is the pixel of interest, q is a training data pixel, d is the difference, i is an iteration number and n is the number of features.

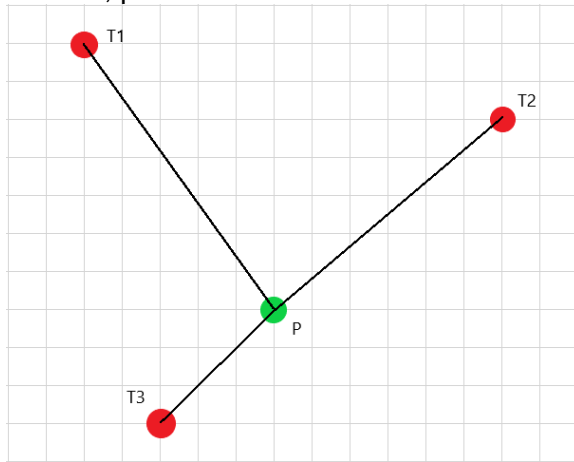
This method is visualized in Figure 7. Here a two-dimensional feature space is shown. The feature space contains three different training pixels (T1, T2 and T3) and one pixel that has to be classified (P). If the distance formula is applied, the following three calculations are used:

$$d(P, T1) = \sqrt{(P_1 - T1_1)^2 + (P_2 - T1_2)^2} = \sqrt{(5)^2 + (-7)^2} = 8.6$$

$$d(P, T2) = \sqrt{(P_1 - T2_1)^2 + (P_2 - T2_2)^2} = \sqrt{(-6)^2 + (-5)^2} = 7.8$$

$$d(P, T3) = \sqrt{(P_1 - T3_1)^2 + (P_2 - 3)^2} = \sqrt{(3)^2 + (3)^2} = 4.2$$

This means the distance between P and T3 is the shortest. Using the minimum distance method, pixel P would be classified as the same class as T3.



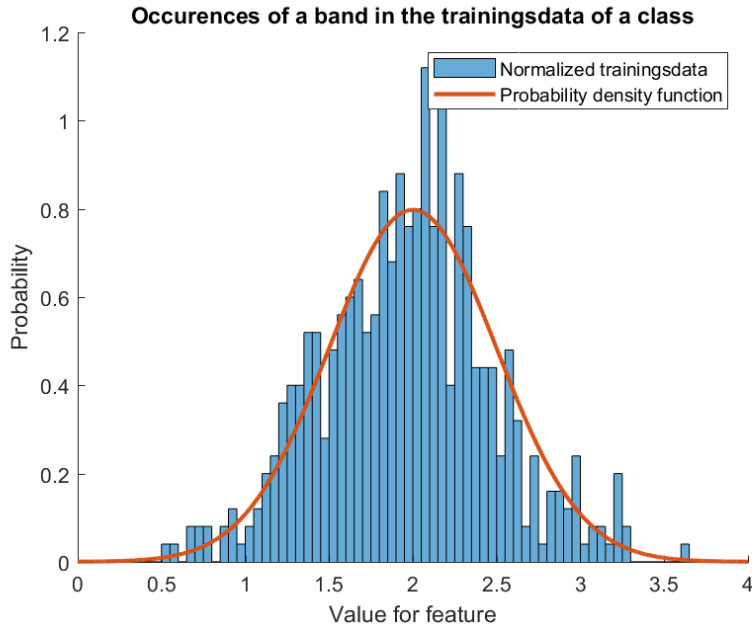
**Figure 7 Minimal distance visualization**

### Maximum likelihood

The maximum likelihood method is again a supervised method. For the Maximum likelihood algorithm the likelihood that a pixel belongs to a certain class is computed. To do this, first the probability distribution of a class needs to be computed. The probability density function is a function that gives the probability that a class would contain a certain value. This function is based on the values that are given in the training data. In Figure 8, a theoretical histogram of one band of the training data for a class is shown. From this data a mean and a standard deviation can be computed. Using the following formula, the probability density function can be constructed out of the mean and standard deviation:

$$f(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)}$$

Here  $x$  is the value of the feature,  $\mu$  is the mean of the feature in the training data and  $\sigma$  is the standard deviation of the training data.



**Figure 8 Normalized training data band and probability density function**

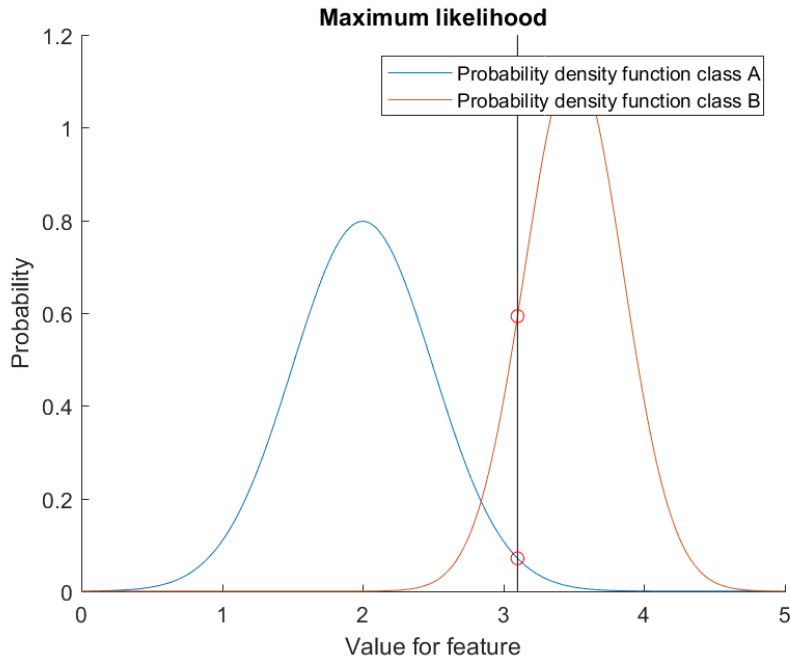
This probability density function is computed for every feature in the training data and for every training dataset.

Now assume there is a dataset with one feature and training data for two classes: class A and class B, as shown in Figure 9. Class A has a mean of 2 and a standard deviation of 0.5, while class B has a mean of 3.5 and a standard deviation of 0.35. When the pixel that has to be classified has a value of 3.1, the calculations are as follows:

$$f(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} = \frac{1}{\sqrt{2\pi}0.5} e^{-\frac{(3.1-2)^2}{2*0.5^2}} = 0.071$$

$$f(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} = \frac{1}{\sqrt{2\pi}0.35} e^{-\frac{(3.1-3.5)^2}{2*0.35^2}} = 0.593$$

This pixel would be classified as a member of class B. The same is done for the panorama, but then with a multidimensional feature space.



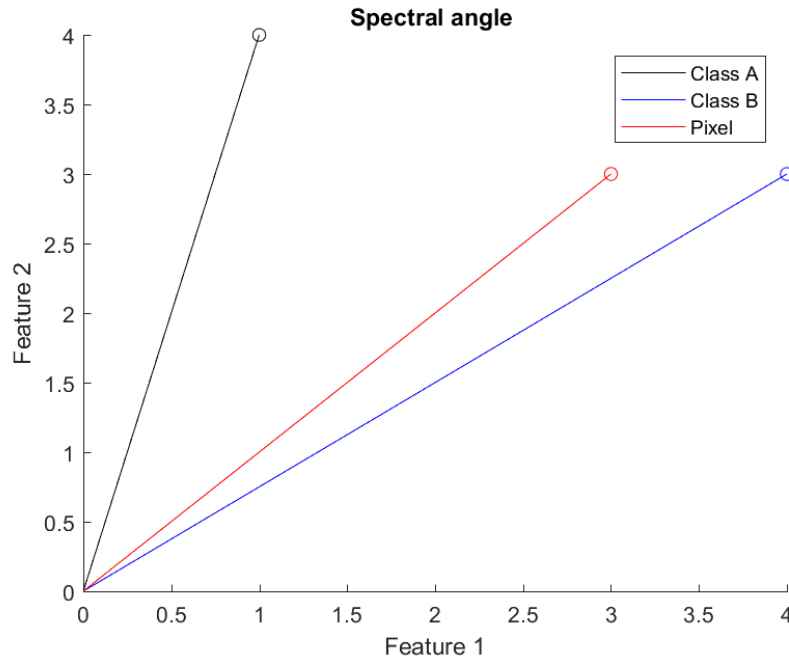
**Figure 9 Probability density functions of 2 classes**

### Spectral Angle

Spectral Angle Mapping is a supervised classification method where the angle is computed between the spectral signature of the training data of a class and the pixel that has to be classified. This is visualized in Figure 10. Here a two-dimensional feature space is shown with two classes and a pixel. The angle between two vectors can be computed with the following formula:

$$\theta(x, y) = \cos^{-1} \left( \frac{\sum_{i=1}^n x_i y_i}{\left( \sum_{i=1}^n x_i^2 \right)^{\frac{1}{2}} * \left( \sum_{i=1}^n y_i^2 \right)^{\frac{1}{2}}} \right)$$

Here  $\theta$  is the angle between the two vectors, x is the spectral signature for the pixel and y is the spectral signature for the training data.



**Figure 10 Spectral angle algorithm**

The coordinates of class A, class B and the pixel are respectively: [1 4], [4 3], [3 3], as shown in Figure 10. The angles for both classes can be calculated as follows:

Class A:

$$\theta(x, y) = \cos^{-1} \left( \frac{x_1 * y_1 + x_2 * y_2}{(x_1^2 + x_2^2)^{\frac{1}{2}} * (y_1^2 + y_2^2)^{\frac{1}{2}}} \right) = \cos^{-1} \left( \frac{1 * 3 + 4 * 3}{(1^2 + 4^2)^{\frac{1}{2}} * (3^2 + 3^2)^{\frac{1}{2}}} \right) = 30.96$$

Class B:

$$\theta(x, y) = \cos^{-1} \left( \frac{x_1 * y_1 + x_2 * y_2}{(x_1^2 + x_2^2)^{\frac{1}{2}} * (y_1^2 + y_2^2)^{\frac{1}{2}}} \right) = \cos^{-1} \left( \frac{4 * 3 + 3 * 3}{(4^2 + 3^2)^{\frac{1}{2}} * (3 + 3^2)^{\frac{1}{2}}} \right) = 8.13$$

It is shown that the angle between the pixel and class B is small than between the pixel and class A. Therefore, the pixel will be classified as class B.

### **CART**

CART or Classification And Regression Trees is a supervised classification. It builds a decision tree out of trainings data using a complex algorithm using machine learning. (Brownlee, 2016) A decision tree classifies data points based on a series of if statements. At every statement the tree splits up until finally a classification is reached. These statements are called nodes and the final solution is a leaf. When this system is drawn out, it looks like a tree with branches and leaves, which is where it got its name. An example dataset is given in Figure 11. This dataset contains two features: x1 and x2 and four classes. The four classes are clearly separated into four different clusters. A decision tree to classify this data is given in Figure 12. The first split is given by the formula  $x_2 < 1.64746$ . This splits the dataset in two parts, where one part contains the blue and red points and one part contains the yellow and purple points. Both parts are then split up by  $x_1 < 1.74994$  and  $x_1 < 1.75168$  respectively.

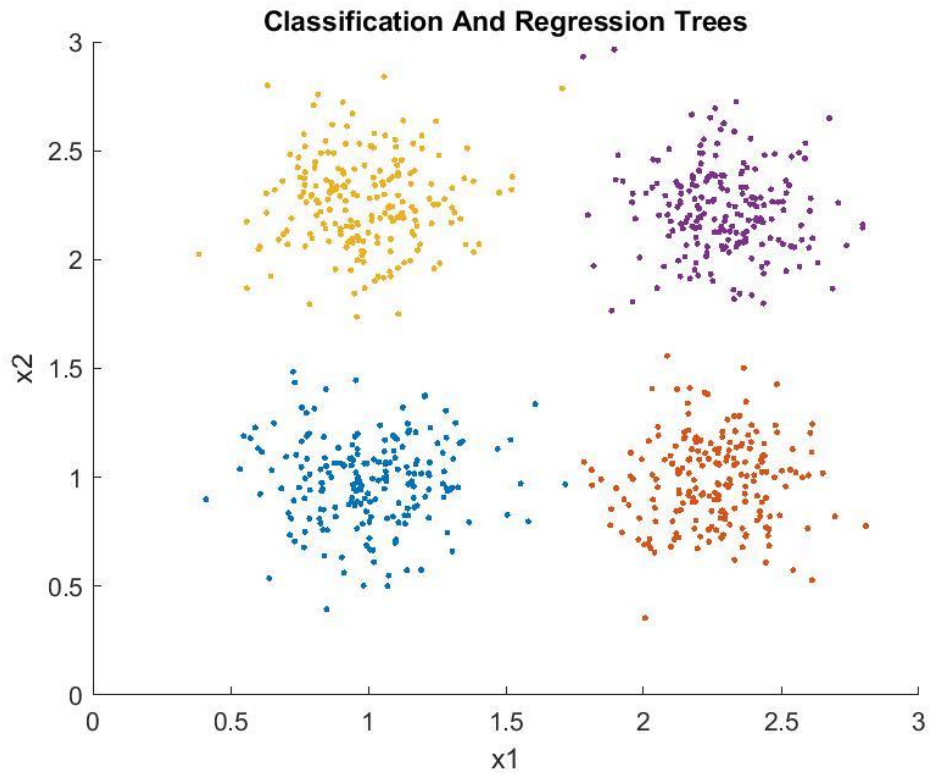


Figure 11 Artificial data showing 2 features and 4 classes

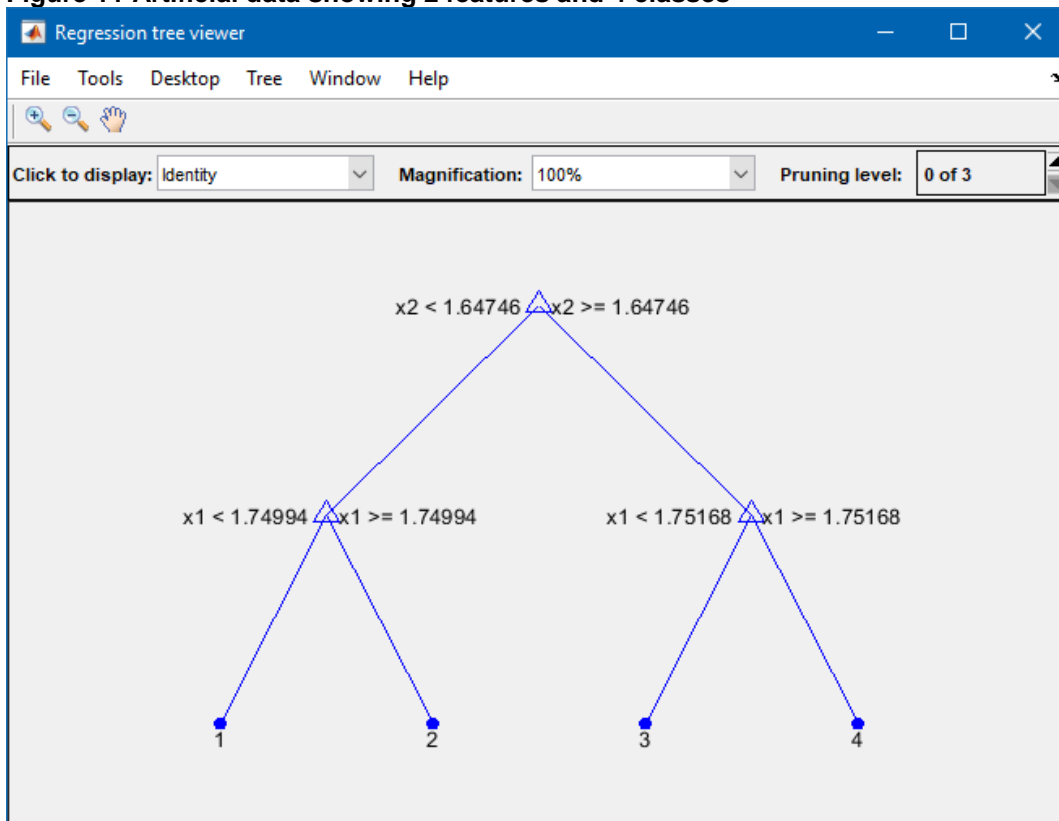


Figure 12 Regression tree created in Matlab

## 4.2 Panorama

### 4.2.1 Stitching

The first step in classifying the aerial photographs is to turn the separate pictures into panoramas. The group of pictures is split up in four strips that were each shot in one overpass of the plane. This is done using ArcSoft Panorama Maker 6. Panorama Maker is a tool that can semi-automatically stitch a batch of pictures together. Simply put, in image stitching multiple photographs are combined into a larger panoramic image by first finding matching points between the photographs. These points are then used to compute the camera location, orientation and distortion parameters. Using these parameters the original images can then be oriented so that a panoramic image is created. (Brown & Lowe, 2006)

The exact processes of image stitching is very complex and will therefore not be discussed in this thesis. First a single strip of photographs and the correct stitching mode are selected in Panorama Maker 6. In this case only the vertical mode works because of the orientation of the pictures. If the pictures are in the right order this will create a basic stitching as can be seen in Figure 13 on the left. The program detects overlapping areas and highlights these. It is possible to improve the matched area between two photos by clicking on the marked area. This will give you a screen with both images and two sets of three markers. These markers must be placed on the same location in both pictures.

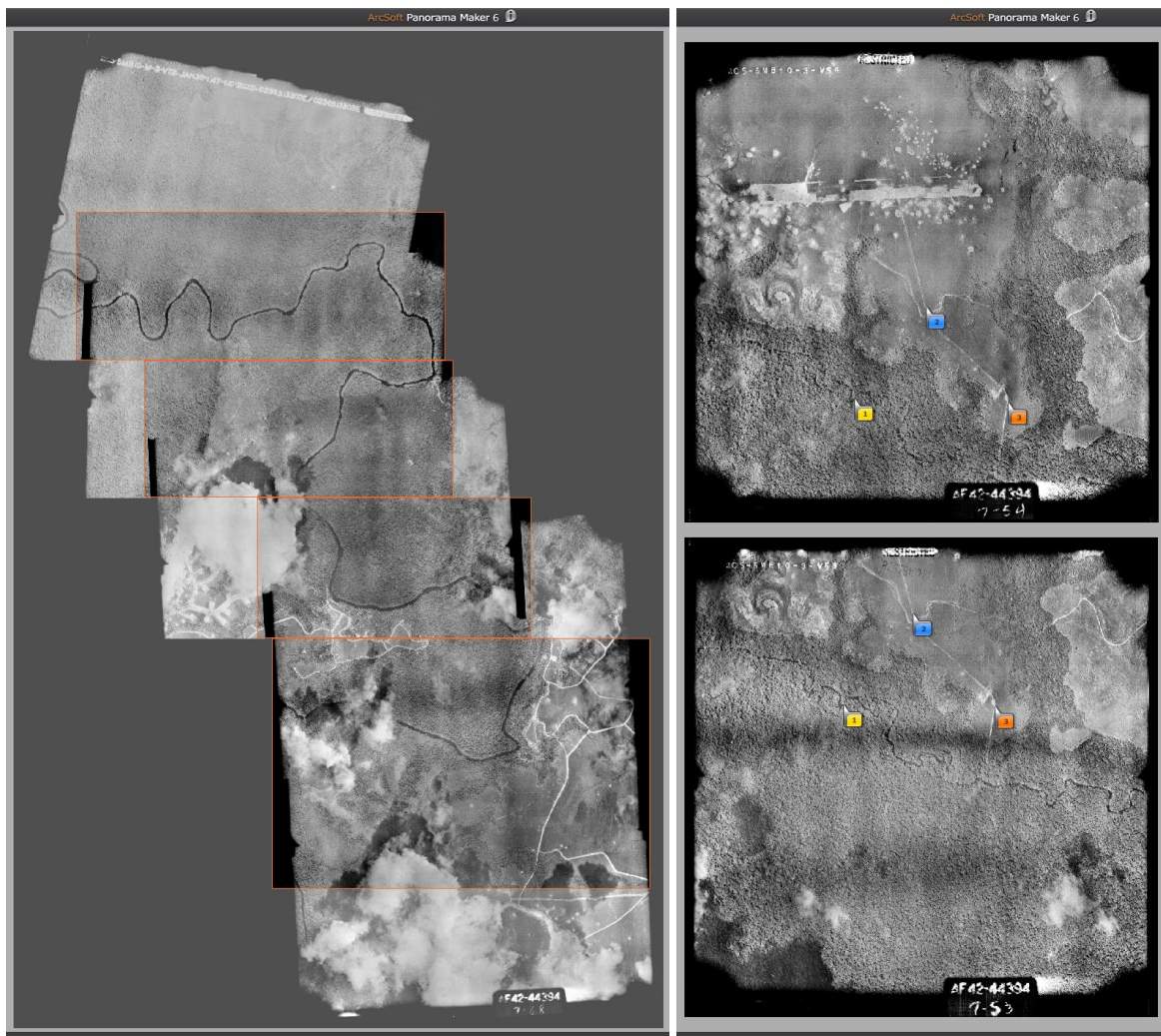
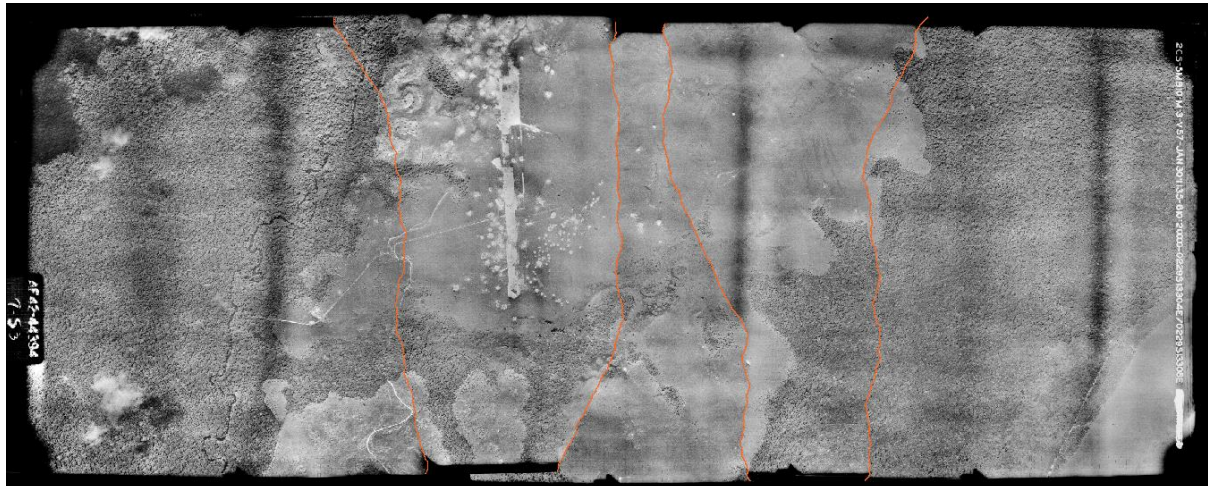


Figure 13 Left: first panorama output with simple stitching, right: picking matching points in pictures to improve stitching



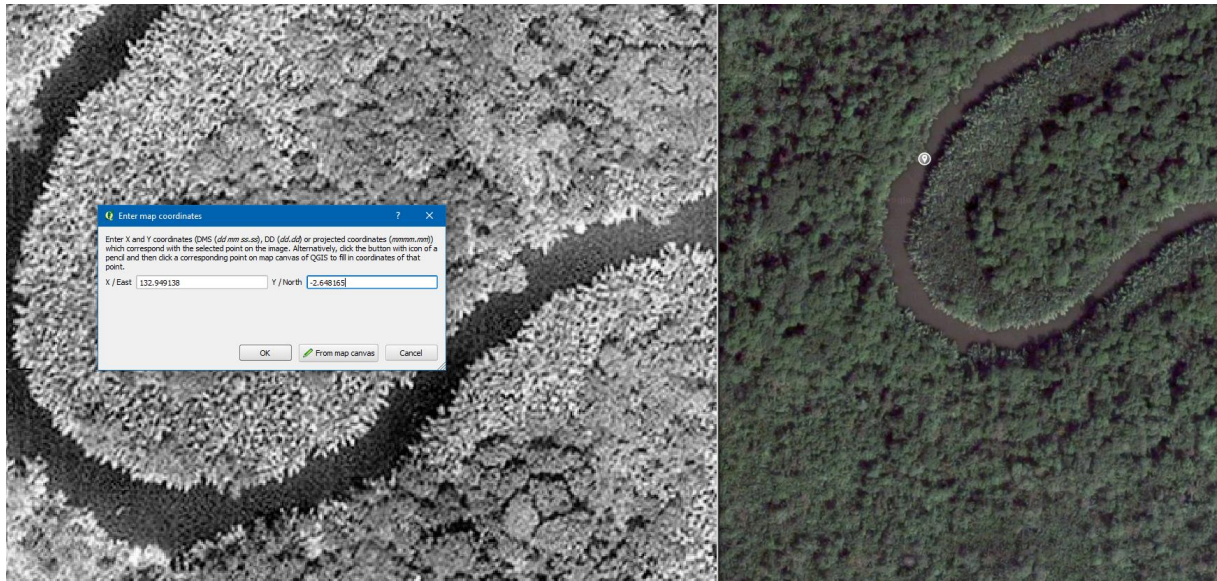
When the markers are placed at the correct position, a button can be pressed to update the stitching. The software will try to match the images based on these markers. The software does not seem to apply a very thorough coregistration algorithm, so it is important to place the markers at the exact same location in both pictures. Once the stitching is sufficient, the next step is to decide how to display the border between the two stitched photographs. In this case this option gave the opportunity to avoid the clouds as much as possible. Since all pictures in a strip were taken in one flight, there was not much gain in the photographed area, but if more datasets were available it would be possible to match different strips and remove all clouds in the entire area. The decision is made by dragging the orange boundary line to the desired location, as can be seen in Figure 14. After this step the panorama can be published.



**Figure 14 Placing the boundaries between photographs**

#### 4.2.2 Georeferencing

The publishing creates a TIFF image. This image needs to be geolocated, so it can be used to make maps of the area. Geolocating is adding information to the picture about pixel size and the location of the pixels in the real world. The geolocating is done in QGIS with the Georeferencer plugin. (Baiocchhi, Lelo, Vittoria, & Mormile, 2013) First, the coordinates of a certain point are found using the Google maps website. Next, the same point is found in the panorama. Clicking on this point will create a marker and allow us to add in the coordinates of the point as can be seen in Figure 15.



**Figure 15** The Georeferencer plugin during the process of adding the coordinates of a new marker.

It is necessary to spread the markers as evenly as possible across the panorama, but this is not always possible as the Google maps images do not provide the required detail to place markers on every location as can be seen on Figure 16. Another challenge here is the fact that there is approximately 70 years between the two photographs and not all landmarks are at the same location anymore. Rivers meander, paths change, and trees grow, so the landscapes are not the same and markers will therefore not be placed on their exact location. This will result in an offset of the panorama, but it will still be acceptable for our project.



**Figure 16** Difference in resolution in the same dataset. The top half has a higher resolution and even shows craters, while the bottom half has a low resolution where roads and rivers are the smallest recognisable objects.

In Figure 17, the resulting georeferenced panorama is shown in QGIS with Google Maps as a background. We can see that roads, rivers, forest boundaries and the airport match quite well, although an offset is visible at some points. The red circles show an offset on the river and on the road.



Figure 17 Georeferenced panorama showing small offsets.

#### 4.2.3 Classification

Next, the georeferenced panorama needs to be classified so we know which parts are grass, forest, and airport for example. Our image only delivers a gray band and more features are needed to classify the image. Python can be used to create new features out of the existing gray image. In the gray image, the intensity of every pixel is defined with a location and a value between 0 and 255. Close to 0 means a dark pixel and close to 255 means a light pixel. First the surrounding pixels for every pixel are found as can be seen in Figure 18. The first time this is done the surrounding band is only one pixel wide, but to get a better result later the thickness is increased to ten pixels.

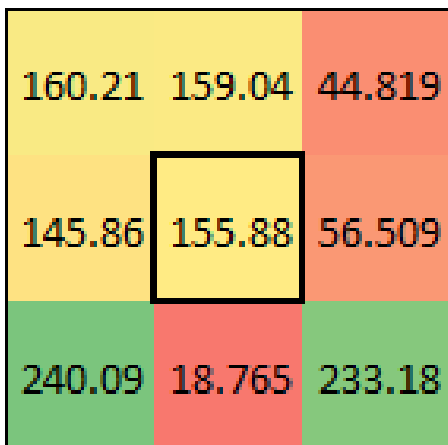


Figure 18 Representation of a pixel surrounded by pixels showing their intensity value.

Now we can calculate different features such as the mean, gradient, difference to the surrounding and the standard deviation. The mean is found by applying the following formula:

$$\mu = \frac{\sum_{i=1}^N x_i}{N}$$

Here  $\mu$  is the mean,  $N$  is the number of pixels used in the calculation (9 in the example case of Figure 18) and  $x_i$  is the intensity at the  $i^{\text{th}}$  pixel.

For the gradient we look in three directions: x, y and the total gradient of the area. For the x gradient, the pixels left and right of the pixel of interest at the same y are taken and for the y gradient the pixels above and below the pixel of interest with the same x as shown in Figure 19. In these values a linear gradient is fitted as shown in Figure 20 with the following formula:

$$\nabla = \frac{f(x_i + h) - f(x_i - h)}{2h} = \frac{f(x_{i+1}) - f(x_{i-1})}{2}$$

Here  $\nabla$  is the gradient,  $f(x)$  is the intensity at location  $x$ ,  $x_i$  is the location of the  $i^{\text{th}}$  pixel and  $h$  is the distance between pixels.

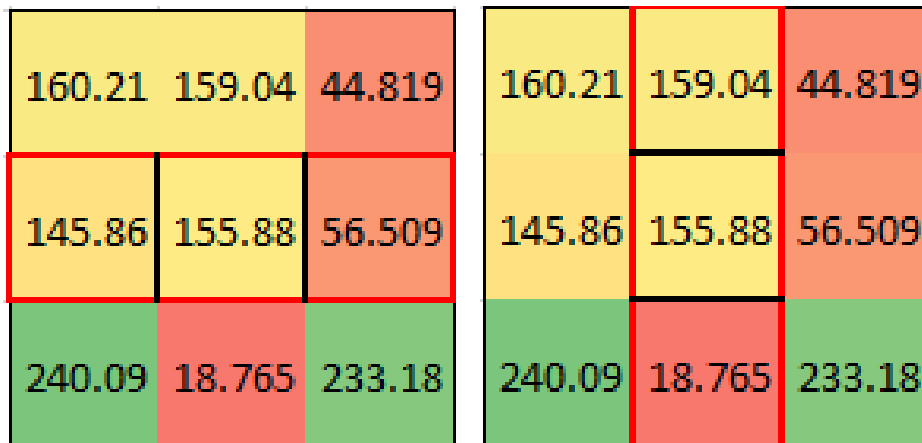


Figure 19 Pixels used for determining the gradient. On the left the three pixels used for the x gradient are outlined in red. On the right the three pixels used for the y gradient are outlined in red.

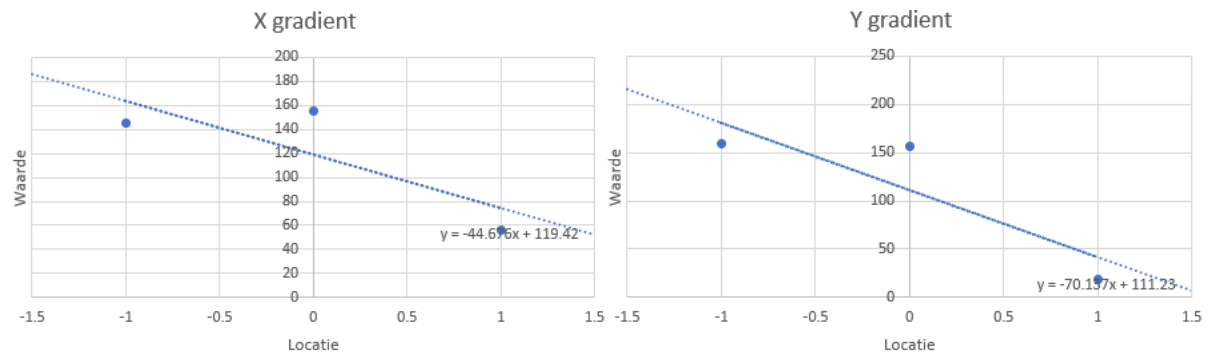


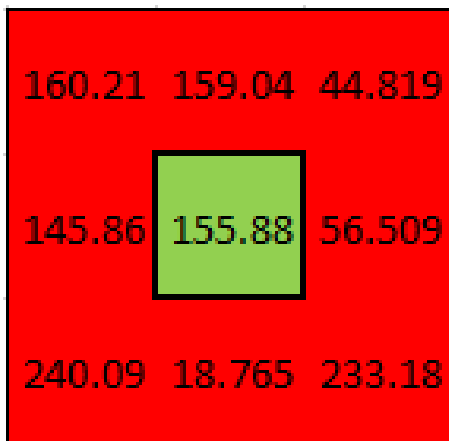
Figure 20 Linear fitting of x and y gradient

The total gradient is computed with the following formula:

$$\nabla = \sqrt{\nabla_x^2 + \nabla_y^2}$$

A higher gradient is expected over forest and a lower value is expected on the runway, because the runway is a single color while the forest has many different colors. This results in a different intensity for every pixel and therefore a larger gradient.

For the difference-feature, the mean of the surrounding pixels (in Figure 21 shown as the red pixels) is computed after which the value of the green pixel is subtracted. The resulting value is the difference between the pixel and its surroundings. Similar to the gradient again, we expect to see higher values over forest and lower values over the runway, since the difference in intensity is smaller over the runway than over the forest.



**Figure 21** Difference between value of interesting (green) pixel and mean of surrounding (red) pixels

The last feature to compute is the standard deviation. The standard deviation is a measure of variation within a data set. This means that we look at a few pixels close to each other. Because the pixels are close we expect the values to be very similar. With the standard deviation we compute the spread in those values. A small standard deviation means a small spread, which means the values are very similar. We would expect a low standard deviation on the runway for example and a high standard deviation in the forest. Just like the gradient the standard deviation is computed in x, y and total direction. To compute the standard deviation, the following formula is used:

$$\sigma = \sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}$$

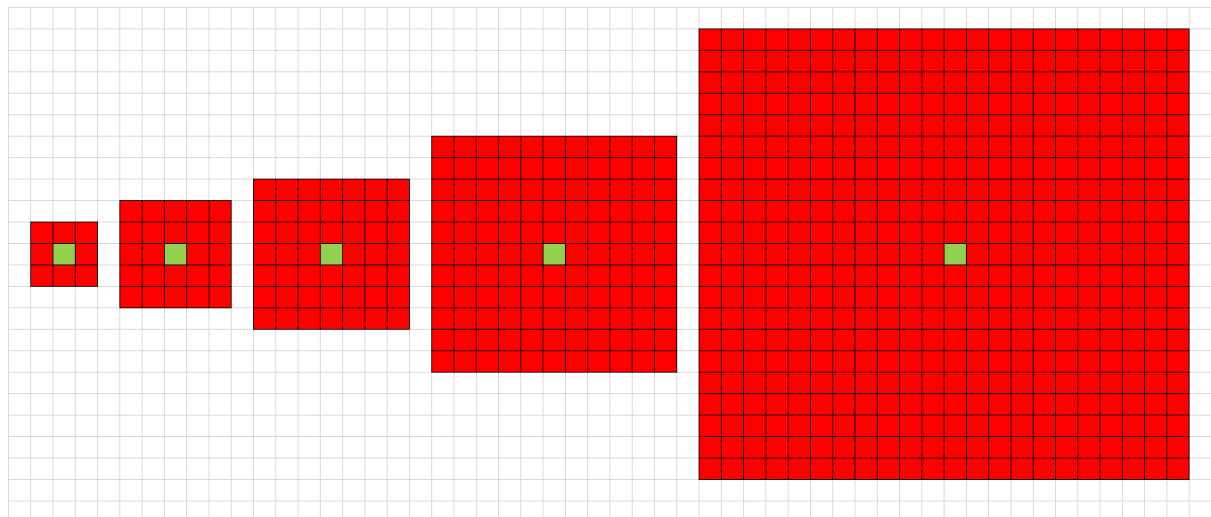
Here  $\sigma$  is the standard deviation, N is the number of pixels used in the calculation,  $x_i$  is the intensity at the  $i^{\text{th}}$  pixel and  $\mu$  is the mean over the pixels.

When all these features are computed we have a dataset with 9 values for every pixel, as can be seen in Table 4.

Band	Description
1	Original image
2	Mean of surrounding pixels
3	Gradient X
4	Gradient Y
5	Gradient total
6	Difference compared to surrounding
7	Standard deviation X
8	Standard deviation Y
9	Standard deviation total

**Table 4 Artificial features for panorama**

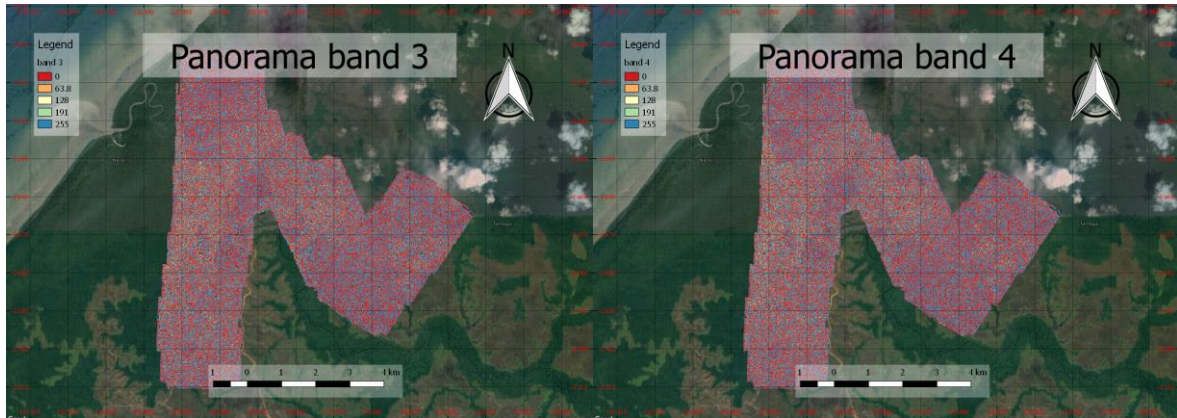
To view the impact of pixels involved in the calculation and to see if the features could be made more defining for the class, the features are created for one, two, three, five and ten rows around the pixel of interest, as can be seen in Figure 22. This resulted in 5 datasets containing 9 bands per image.



**Figure 22 Showing 1, 2, 3, 5, or 10 rows around a pixel to compute features.**

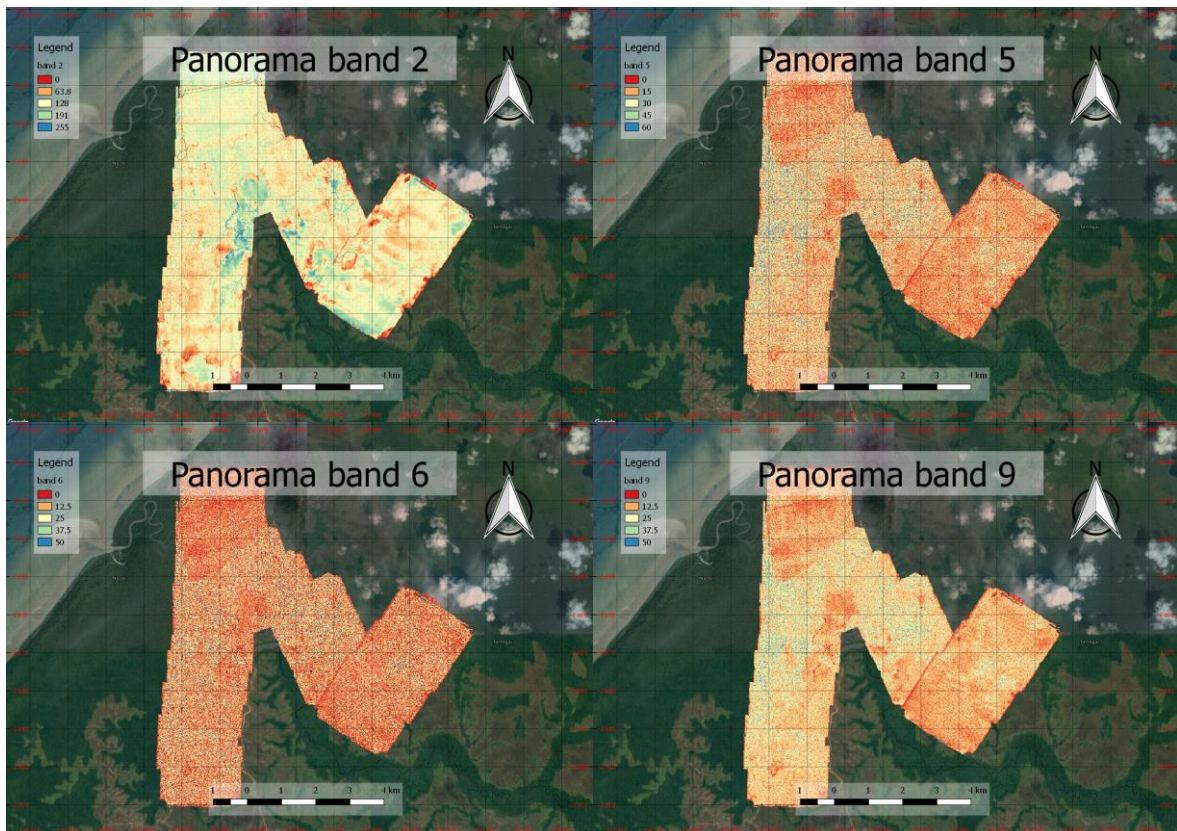
More rows of pixels should mean smoother and better results for features like mean, gradient and standard deviation, but it also means a lower resolution and a mixed zone between two clearly different objects that lie close to each other like forest and grass.

Some of these features are more useful in classifying the panorama than others. In Figure 23, two less useful features are shown. These features do not show a clear difference between for example the forest and the grass or runway. These features will therefore not be used in the classification.



**Figure 23** Less useful features, left: the gradient in x-direction, right: the gradient in y-direction

Classification is done in two different ways: by using K-means in Python and by using the maximum likelihood, nearest neighbor and spectral angle methods in the Semi-Automatic Classification plugin. First the most expressive features were chosen manually. The four chosen features are shown in Figure 24.



**Figure 24** Four most interesting features. Top left: Mean of surrounding pixels, top right: total gradient, bottom left: difference to surrounding pixels and bottom right: standard deviation.

These features are then split from the image in Python and normalized. This means that for every feature the values are scaled so that the minimum value is 0 and the maximum value is 1. This prevents that one feature is more influential than another feature. The result is a  $m$  by  $n$  by four matrix, where  $m$  and  $n$  are the number of rows and columns of the raster with all values between zero and one. Next band two and nine are multiplied with a factor five. The multiplication will give these features a larger weight in the clustering. These features show the clearest classes and therefore the clusters should be mainly based on these two



features and then finetuned on the other two features. The factor five is arbitrarily chosen as it was the lowest factor that showed an improved result over the original clustering.

The first classification algorithm to be applied to the dataset is K-means. Visualizing the full four-dimensional feature space is not possible. Therefore in Figure 25 the mean of the surrounding pixels and the total gradient are scattered, since these two features show the most characteristics and are expected to show the clearest clusters. No clusters show up clearly in this combination and neither does it for any other combination of features. This means the K-means method might not work very well on this dataset.

Using the K-means algorithm the solutions for five, ten and fifteen clusters are computed. The resulting images are then loaded into QGIS in order to create the final maps.

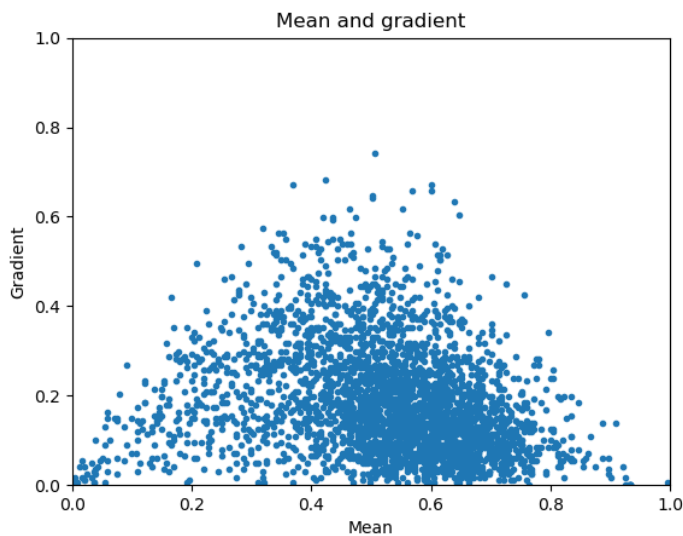
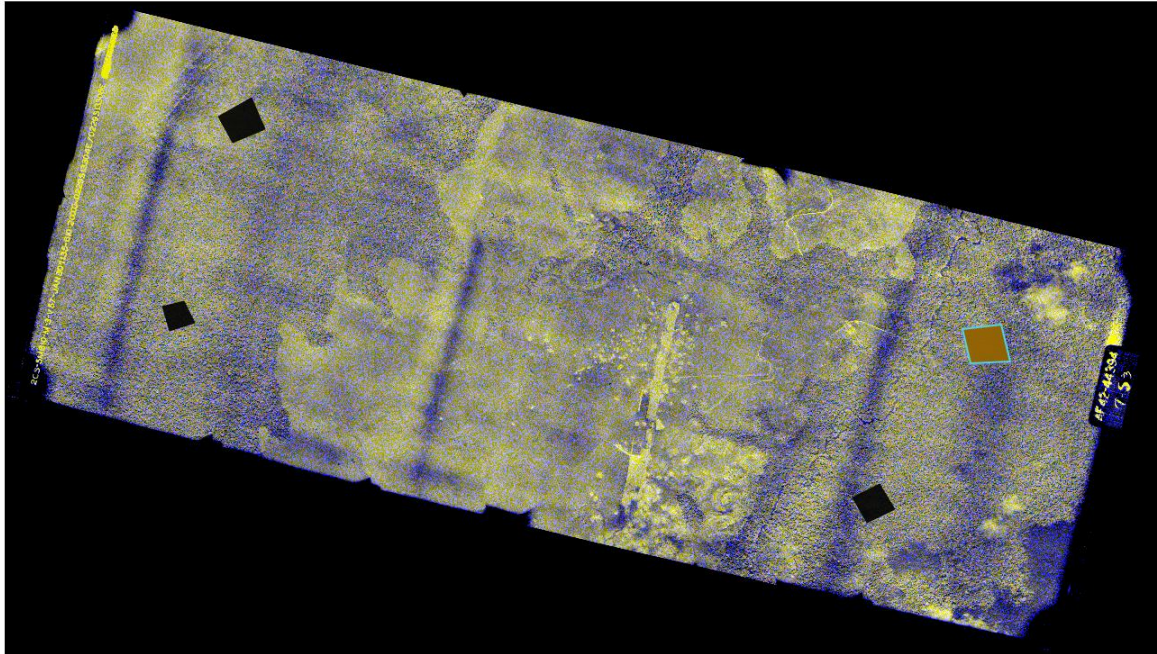


Figure 25 Scatter of the mean and the standard deviation.

The other classification methods are done in QGIS itself. Here the same four features that were used in the K-means method are loaded into QGIS. Using the Semi-Automatic Classification Plugin, the image is then classified. The plugin first needs training data. This means that areas containing one type of class are selected, so the program can learn what the class looks like in the feature space, as can be seen in figure 26. Here four shapes are shown to learn the Plugin to recognize the forest class. It is important to spread out the training data and make sure all different types of forest pixels are included. This training is done for all different classes found in the panorama.



**Figure 26 Selecting training data for the Semi-Automatic Classification Plugin.**

The Plugin has three different algorithms to compute a classification:

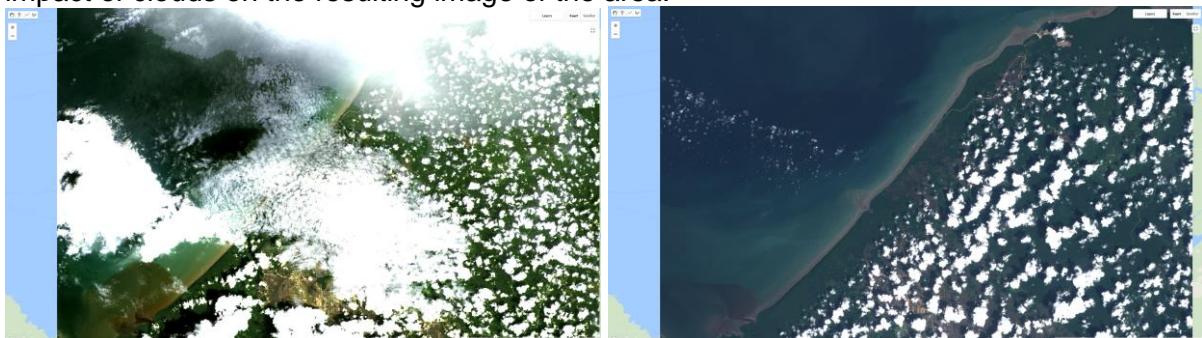
- Minimum distance
- Maximum likelihood
- Spectral angle mapping

The theory behind these algorithms is explained in 6.1. After the algorithms are run over the dataset the resulting classifications are then converted into high detail maps using the print composer.

## 4.3 Landsat and Sentinel

### 4.3.1 Cloud filtering

In the Google Earth Engine the Landsat 8 and Sentinel-2 is processed as well, to have an accurate map of the current situation. The main problem with satellite data in this area is the fact that there are always clouds covering a large portion of the area. Figure 27 shows the impact of clouds on the resulting image of the area.



**Figure 27 Cloud cover over the area of interest as seen in Landsat 8(left) and Sentinel-2(right)**

Both for the Landsat and Sentinel-2 missions there are algorithms available to filter the clouds automatically. For Landsat 8 data, this means it is first atmospherically corrected

using LaSRC, which stands for Landsat 8 Surface Reflectance Code product. This product removes the atmospheric influences, so the surface can be studied more accurately. Next, cloud, shadow, water and snow masks are created using CFMASK and the pixel saturation per band is computed. (Google, n.d.) CFMask is a decision trees based multi-pass algorithm that detects clouds, shadows water and snow. Like with other cloud masks, the algorithm struggles over bright objects and with thin clouds. (USGS, n.d.)

The effectiveness of the Cloud filter for Landsat 8 is mainly due to the adding of band nine, called the Cirrus band. This band covers a small part of the spectrum at 1370nm ±10nm. This part of the spectrum is mostly absorbed by the atmosphere, which means that everything that shows up on this band must be either highly reflective or be very high up in the atmosphere. This results in the band showing high cirrus clouds, that are normally very hard to spot for satellite imaging because of their soft edges. (NASA, n.d.)

The last three bands available in the Landsat dataset contain the results for these processes:

- Aerosol attributes
- Pixel quality attributes
- Radiometric saturation quality attributes

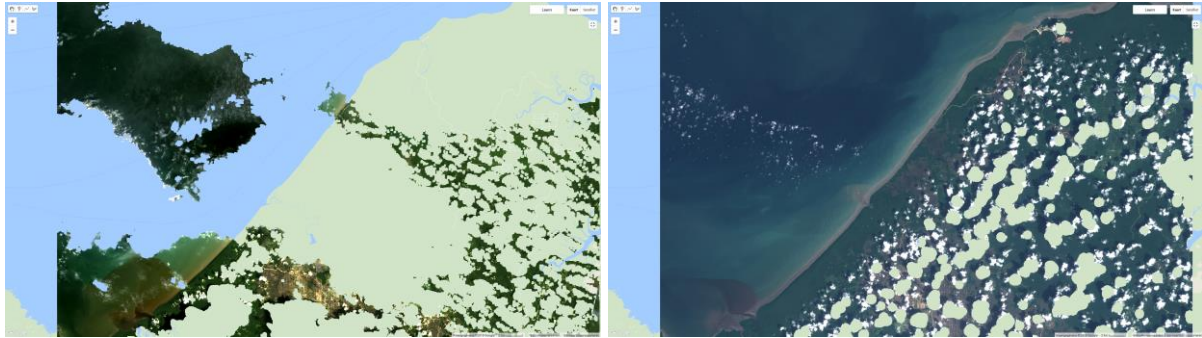
The results for this CFMASK are put in the pixel quality attributes band. This is a 16-bit band that shows the computed quality attributes as can be seen in Table 5.

Bit	Attribute
0	Fill
1	Clear
2	Water
3	Cloud Shadow
4	Snow
5	Cloud
6-7	Cloud Confidence (00 = None, 01 = Low, 10 = Medium, 11 = High)
8-9	Cirrus Confidence (00 = None, 01 = Low, 10 = Medium, 11 = High)
10	Terrain Occlusion

**Table 5 Pixel quality attribute bit flags**

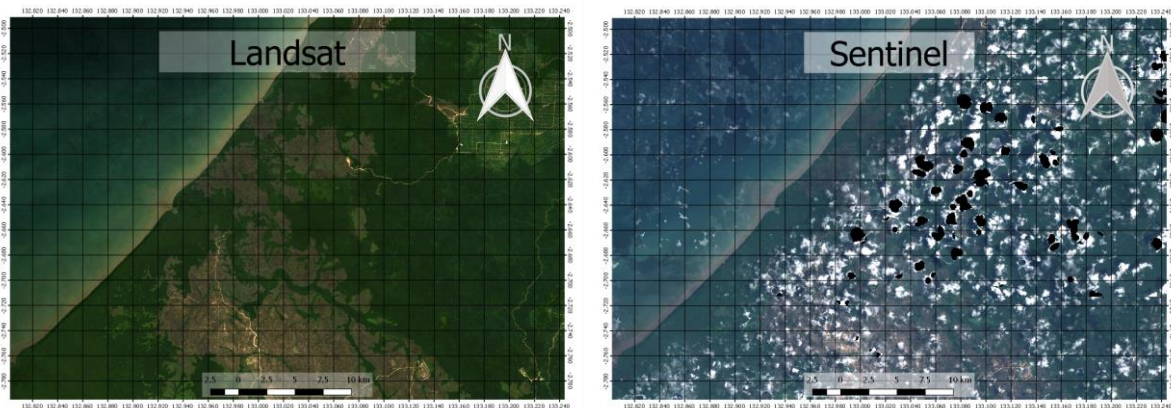
The simple method used in this assignment is to set the cloud mask to reject pixels that are classified as cloud or cloud shadow by the algorithm. This results in the image seen in Figure 28 on the left.

For Sentinel-2 a simple cloud filter is available as well, but it is not as effective as the one available for Landsat 8. This is mainly due to the lack of the band that can detect cirrus clouds. Sentinel-2 bases its cloud detection on a reflectance threshold in B2, which shows blue. Besides that, it also uses B11 and B12, which are the Short Wave Infrared band, to avoid false detection, mainly due to snow/cloud confusion. Snow and clouds both have a high reflectance in blue, but clouds have a high SWIR reflectance whereas snow does not reflect SWIR very strongly. (ESA, n.d.) The resulting filtered image can be seen in Figure 28 on the right.



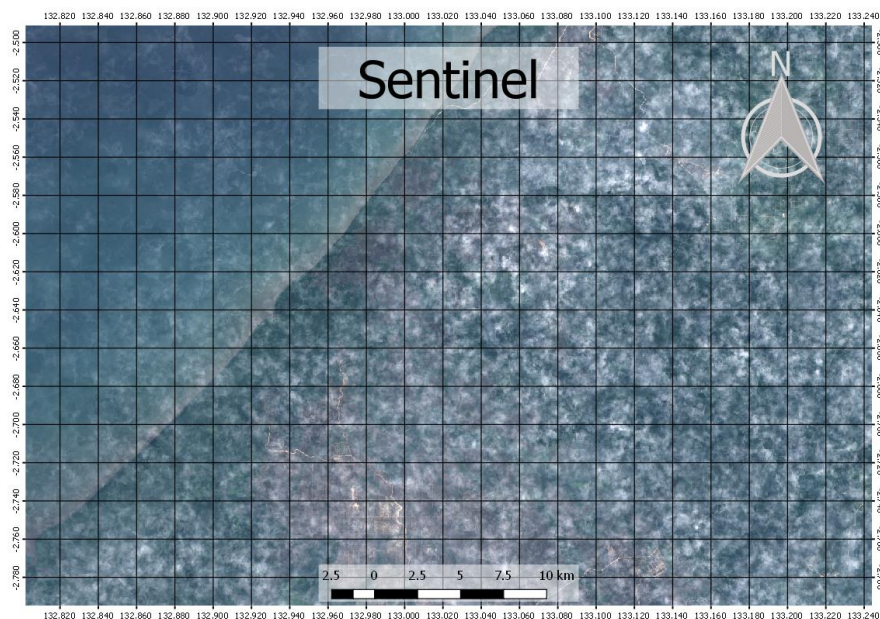
**Figure 28 Effect of cloud filter in Landsat images(left) and Sentinel-2 images(right)**

The dataset contains large holes when the cloud is filtered. This is why multiple images over the full acquisition periods of Landsat 8 and Sentinel-2 are taken where all clouds are removed and then the median of the remaining values is taken. If for Sentinel-2 all available data is used there are still holes and clouds visible. This is shown in Figure 29.



**Figure 29 Final images after removing clouds and stacking. Left: Landsat. Right: Sentinel-2**

Another way to address the clouds could be to take the average value over all available Sentinel-2 data, without masking the clouds. Assuming the chance of a cloud is the same for every location in the region of interest, the average should leave a uniform cloud that has the same impact on every pixel. Although this does not allow us to look at the true surface reflectance, it would be possible to look at differences between pixels and find pixels that stand out. The resulting true color of this mean cloud method can be seen in figure 30.



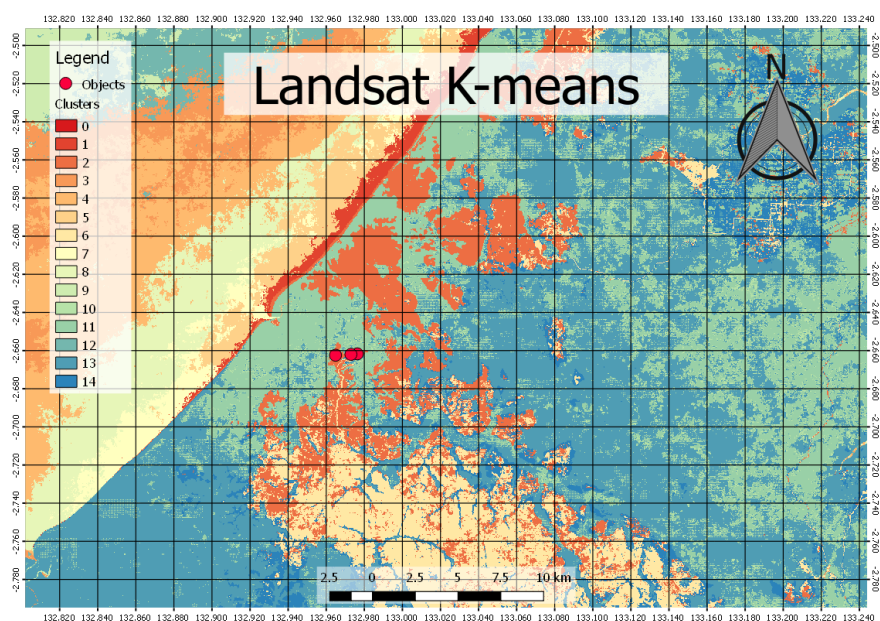
**Figure 30 Average of unfiltered Sentinel-2 data**

This leaves three datasets:

- Landsat 8 with clouds masked
- Sentinel-2 with clouds partially masked and holes in the data
- Sentinel-2 without cloud mask averaged over time

### 4.3.2 Classification

These datasets can then be classified using different classification methods. The first method is K-means as explained earlier in the chapter. The Google Earth Engine platform delivers a script that can be adjusted to run K-means on the cloud. This means it will run a lot faster and it saves on programming time since the script is already given. The classified images can then be uploaded to Google Drive and from there downloaded to the local computer. In QGIS the image can then be displayed as can be seen in Figure 31.

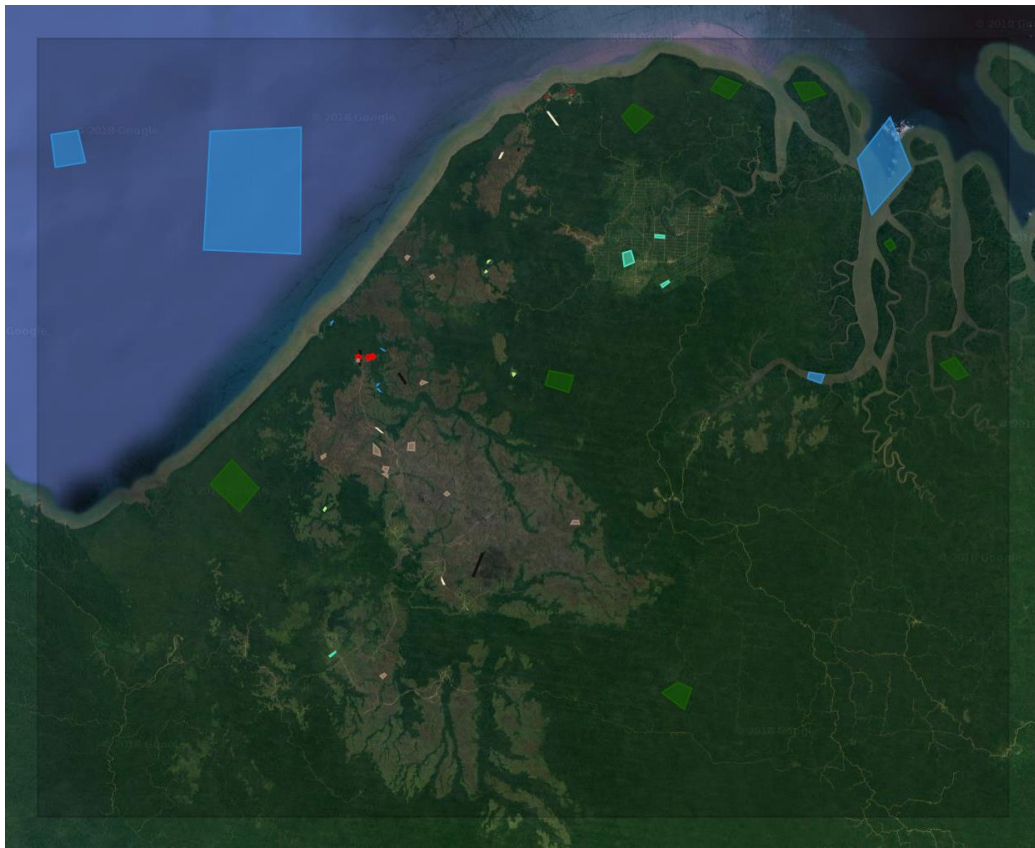


**Figure 31 A 15 cluster classification of the Landsat dataset**

Besides the unsupervised method K-means it is also possible to do supervised classification on the Google Earth Engine platform. Just like the Semi-Automatic Classification Plugin polygons need to be drawn by hand in order to train the classifier. The following classes are distinguished:

- Water
- Buildings
- Airports
- Roads
- Grass
- Plantation
- Forest
- Open area

These classes all contain at least four polygons containing at least ten pixels per polygon, as seen in Figure 32. All pixels in a polygon are used for the training process, unless a band is missing on a certain pixel due to the way the program is written. Some classes are more complex than others or could also be seen as two classes. Water for example could also be seen as a sea class and a river class and forest consists of different types of trees. These classes therefore need more training data to work properly. The polygons are drawn based on the underlying Google Maps image and are not based on data from the field. In addition to these classes, an extra class is sometimes used: equipment. This class contains a few known points in the field that contain equipment. Since only a few points are available, the class is not very trustworthy and cannot be checked. The classification process is therefore run with and without the equipment class, so that the classification process can be checked and possible equipment locations can still be computed.



**Figure 32** Shapefiles used for classification process

The classification algorithm used in this case is CART or Classification And Regression Trees, which is explained in chapter 4.1.

For Landsat the following bands are used:

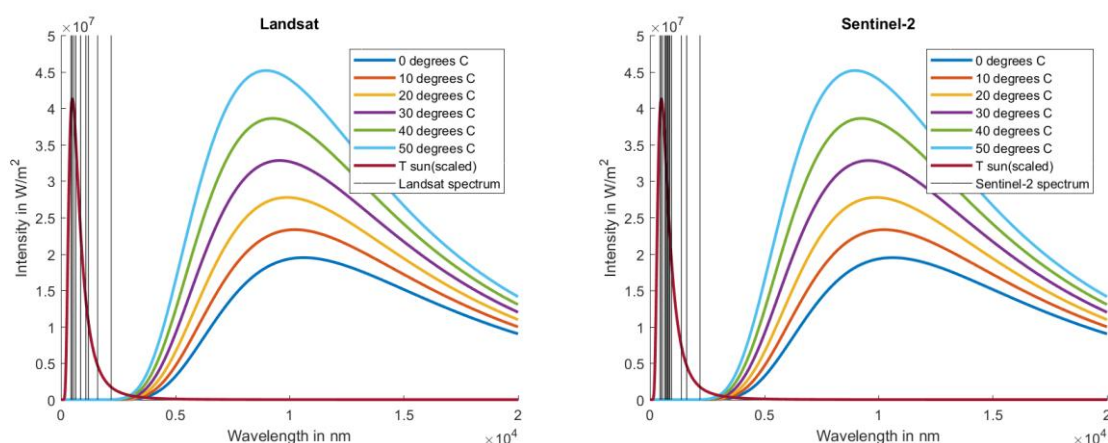
- B2
- B3
- B4
- B5
- B6
- B7
- B10
- B11

For Sentinel a few of the bands are blocked by clouds and are therefore not used. In the end the following Sentinel-2 bands were used for the classification:

- B4
- B5
- B6
- B7
- B8
- B8A
- B11
- B12

The resulting classified images are downloaded and converted into maps using QGIS.

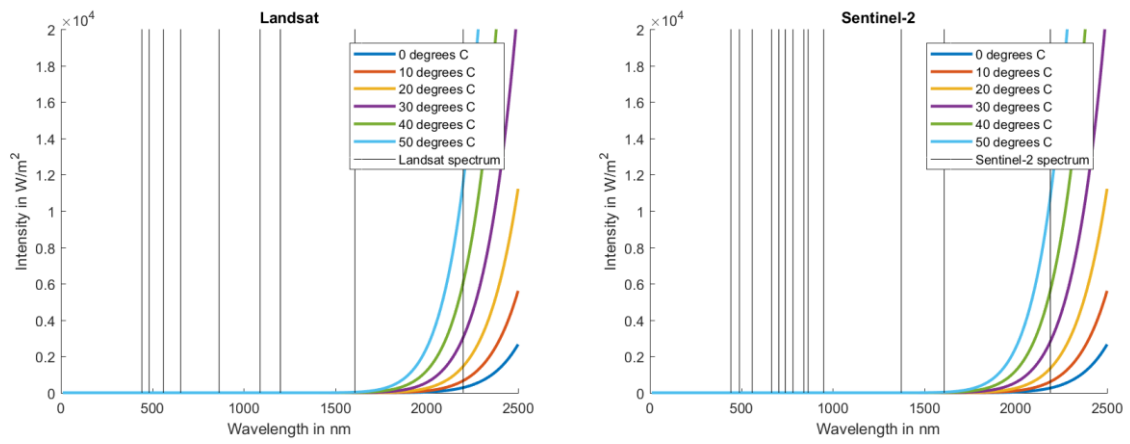
Besides classification it can also be interesting to look at specific bands. Every object radiates electromagnetic waves. The frequency spectrum based of these emitted waves depends on the temperature. If we assume that the temperature of a metal plane left in the sun becomes higher than the temperature of its surroundings, it is possible to look for changes in the radiation spectrum. In Figure 33 the radiation spectrum of the objects of 0, 10, 20, 30, 40 and 50 degrees Celsius are shown and a Scaled down spectrum of the sun to compare. The Black lines in this figure show the bands used in Landsat and Sentinel.



**Figure 33 Landsat 8(left) and Sentinel-2(right) bands compared to radiation spectra**

From this figure it becomes clear that most objects found on earth do not emit waves that are picked up by either Landsat or Sentinel-2 and most of the received energy is reflected solar energy. If we zoom in on the bands, as can be seen in Figure 34, we see that in Landsat band 7 and in Sentinel-2 band 12 might still be interesting since they do show a change in radiation for different temperatures. The differences are small compared to the

reflection of the sun, but since the incoming solar radiation for the area of interest is almost constant, this does not have to be a problem. To look for interesting spots, the top 0.2% of Landsat 8 and the top 1% of Sentinel-2 are shown in QGIS and overlaid on the map.



**Figure 34 Landsat 8(left) and Sentinel-2(right) bands compared to radiation spectra zoomed in**

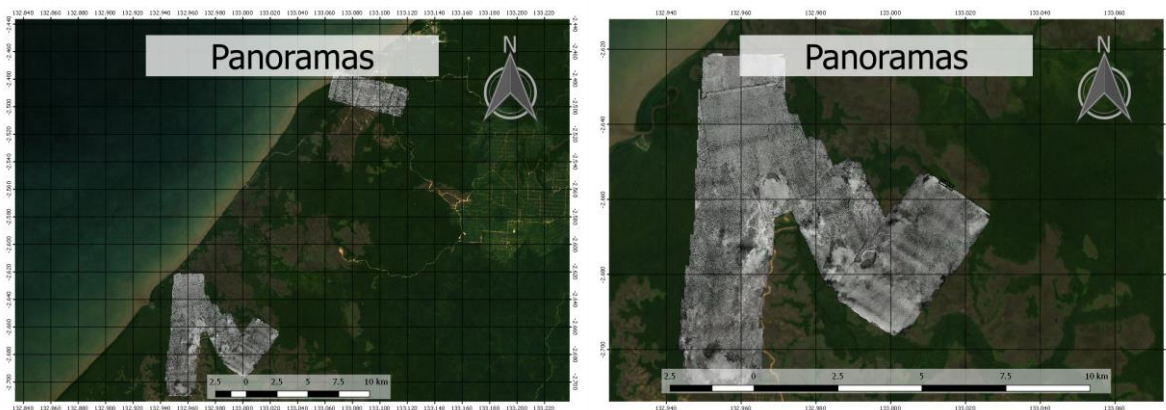


# 5 Results

As with the method, the results are also split into a part for the panoramas and a part for the satellite imagery. All the maps shown in results can be found in a higher resolution in the appendix, together with maps that were not interesting enough for the main report.

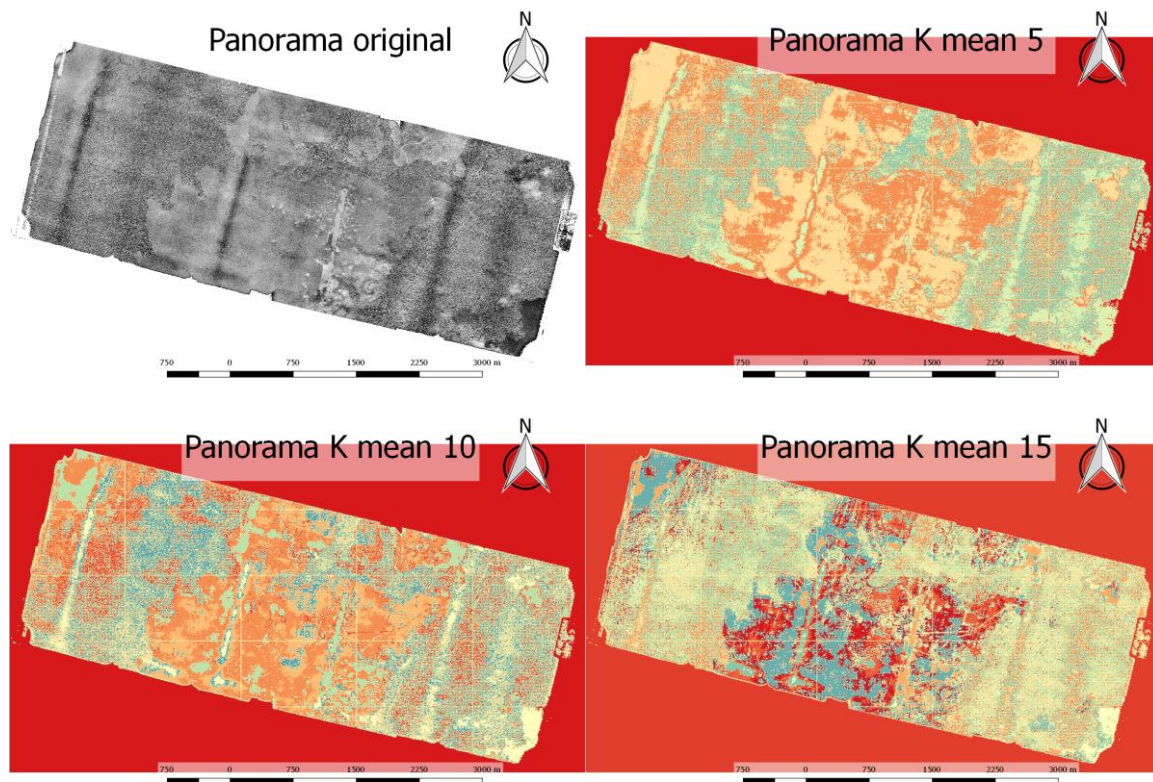
## 5.1 Panorama

In Figure 35 all four georeferenced panoramas are shown with a true color visualization of the Landsat bands as background. This figure shows the location of the panoramas in the region of interest. The left image shows all panoramas, and on the right a zoomed version of panorama two, three and four is shown.



**Figure 35 Georeferenced panorama locations in region of interest**

In figure 36 the K-means results for one of the panoramas is shown. Top left shows the original, top right shows five clusters, bottom left shows ten clusters and bottom right shows fifteen clusters. On the original panorama multiple dark and light bands are shown. These bands are present in the original pictures and it is unknown what caused them. The clusters are shown as different colors, and the colored background is classified as one of the clusters. A clear division between forest and other features is visible. The other classes that are present such as grass, the airport and water do not appear to have significant enough features to be separated using this method. Another noticeable feature is that the bands from the original panorama are still visible in the K-means classified image.

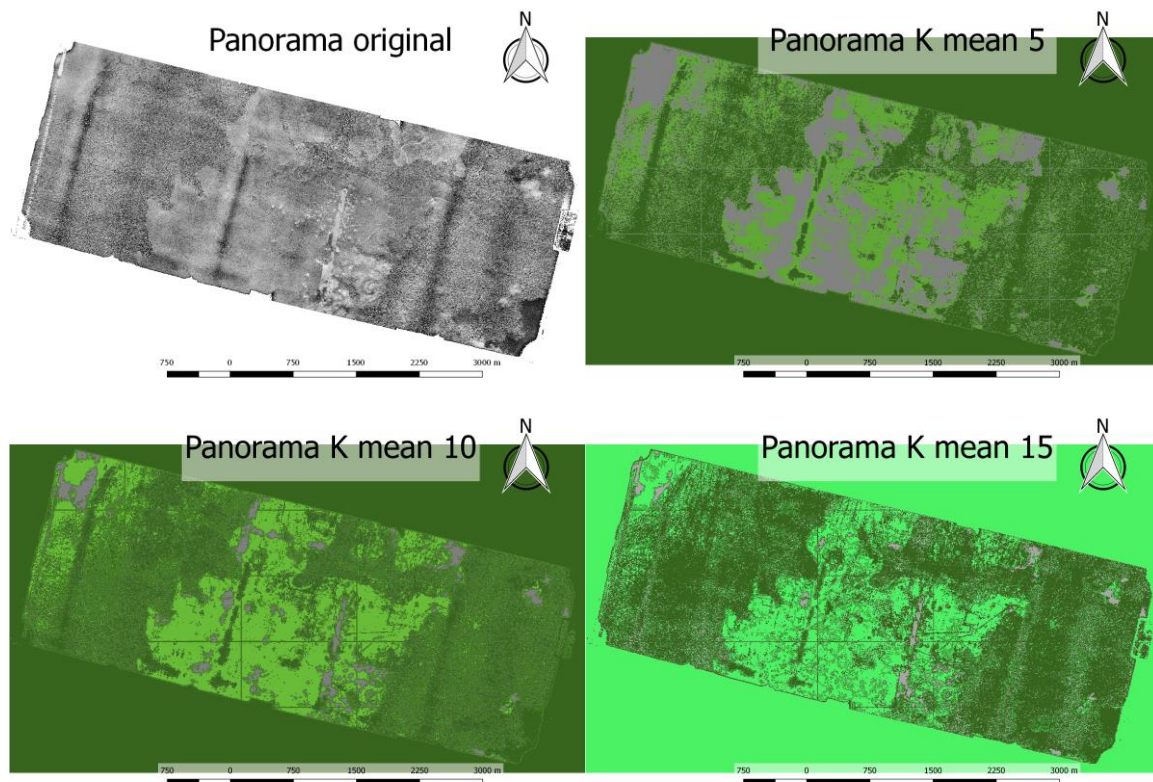


**Figure 36** The original panorama and resulting maps after applying K-means using 5, 10 and 15 clusters.

Figure 37 shows the same clusters, but now the clusters are manually classified as one of the following three groups:

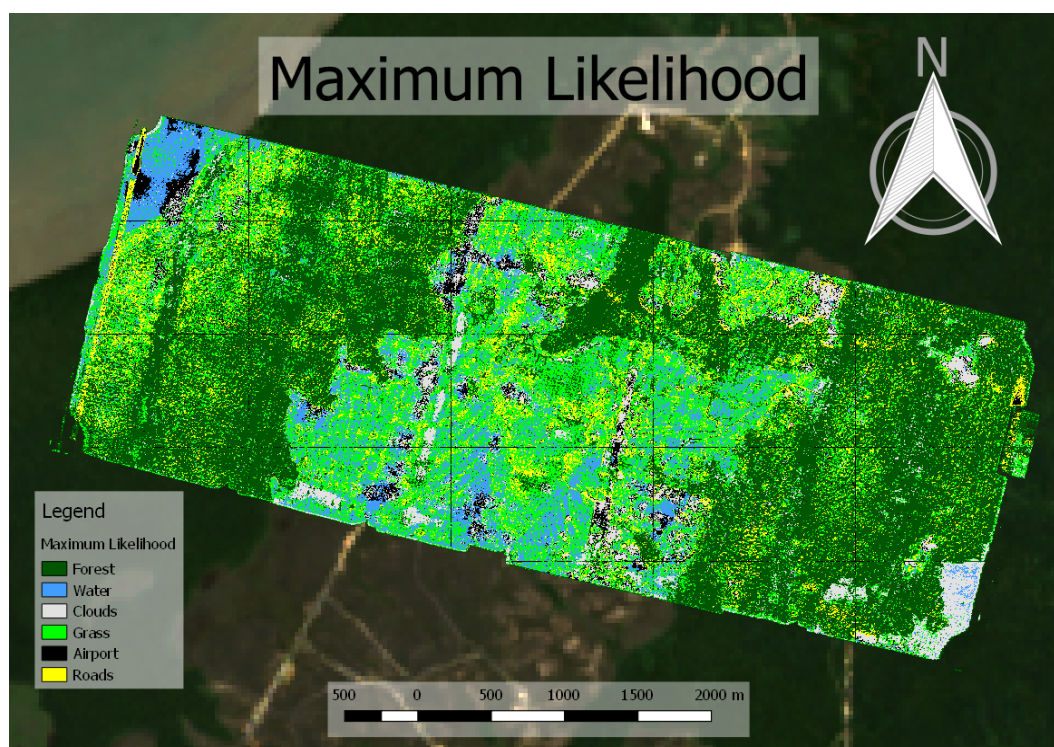
- Forest: Dark green
- Grass: Light green
- Airport: Grey

K-means is not meant as a full classification method, but it should show what classes should be easily classified. In Figure 37 we see that when ten or fifteen classes are used the result matches the truth closer than when only five clusters are used. This means that forest, for example, has different areas that have different features that should all be classified as forest. In order to get a correct classification enough samples should be taken so all features are included.



**Figure 37 The original panorama and manually classified K-means clusters**

The four-feature image is also classified using the Semi-Automatic Classification Plugin. This plugin had three different classification algorithms. In Figure 38, the results of the maximum likelihood algorithm are shown. The results from minimum distance and spectral angle are less accurate and are therefore not shown here. All results can be found in the Appendix.

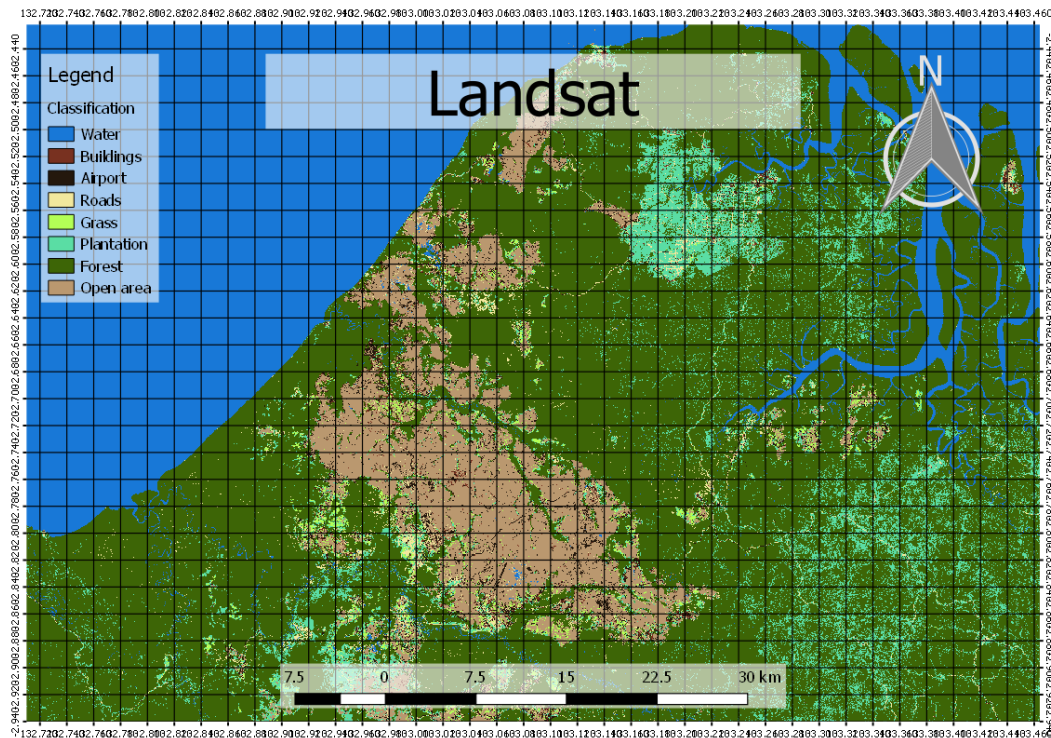


**Figure 38 Maximum likelihood classification of panorama**

## 5.2 Landsat and Sentinel

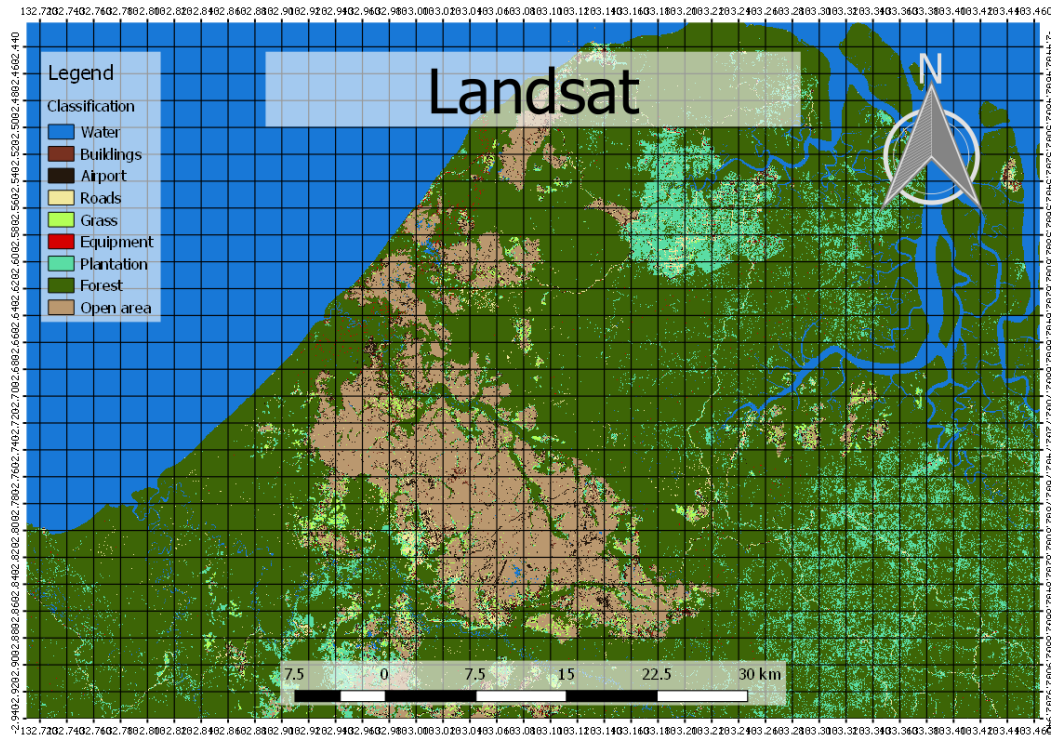
The satellite imagery datasets are classified twice: once without the equipment class and another time when it was included. This is done because the equipment class has not enough known locations to create training- and verification data. Therefore, one version of the classification can be accurately validated, while the other one is the one we are actually interested in, since the main goal is to find hidden planes and equipment.

Figure 39 shows classifications of Landsat 8. The classification is applied on Landsat 8, averaged Sentinel-2 and Sentinel-2 with an applied cloud filter. Since Landsat 8 has the best results these are shown here. The full results can be found in the Appendix.



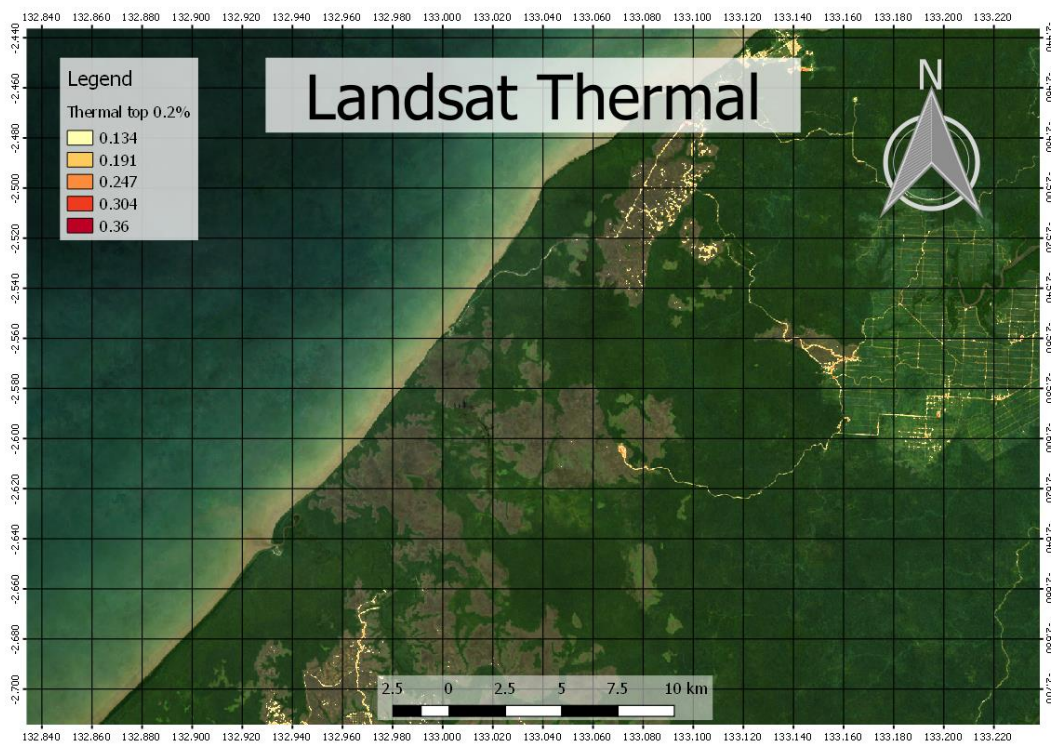
**Figure 39 Classification of Landsat 8 dataset**

In Figure 40 again Landsat 8 is classified but now the equipment class is also included. The red spots that show up in the classification are areas that could be interesting to visit and investigate a little closer. The Sentinel-2 classifications can be found in the Appendix.



**Figure 40 Classification of the Landsat 8 dataset including the equipment class**

Figure 41 shows the highest temperatures in the region of interest for Landsat 8. The top 0.2% highest surface temperatures are shown. The results for Sentinel-2 are shown in the Appendix.



**Figure 41 Thermal top 0.2% in Landsat 8 dataset**

# 6 Method of validation

## 6.1 Panorama

The first potential error source for the panoramas is the georeferencing. In Figure 42 panorama one is displayed on top of a true color visualization of Landsat data and in Figure 43 panoramas two, three and four are displayed on the same dataset. The red ellipses show the locations of distinct features on the panorama that can be checked to see the accuracy of the geolocating.

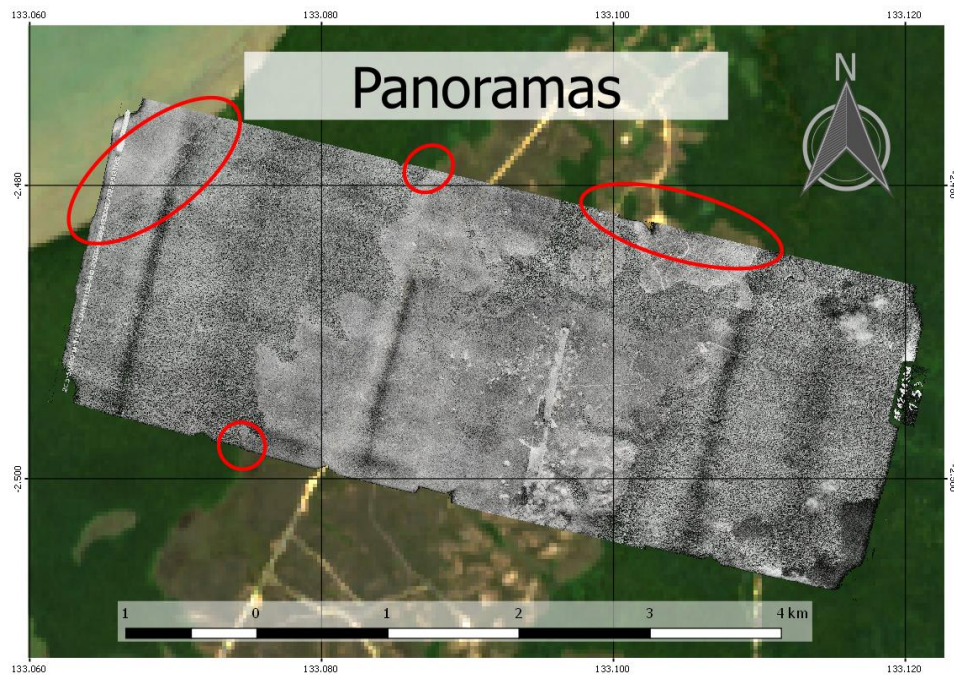


Figure 42 Errors in georeferencing of panorama 1

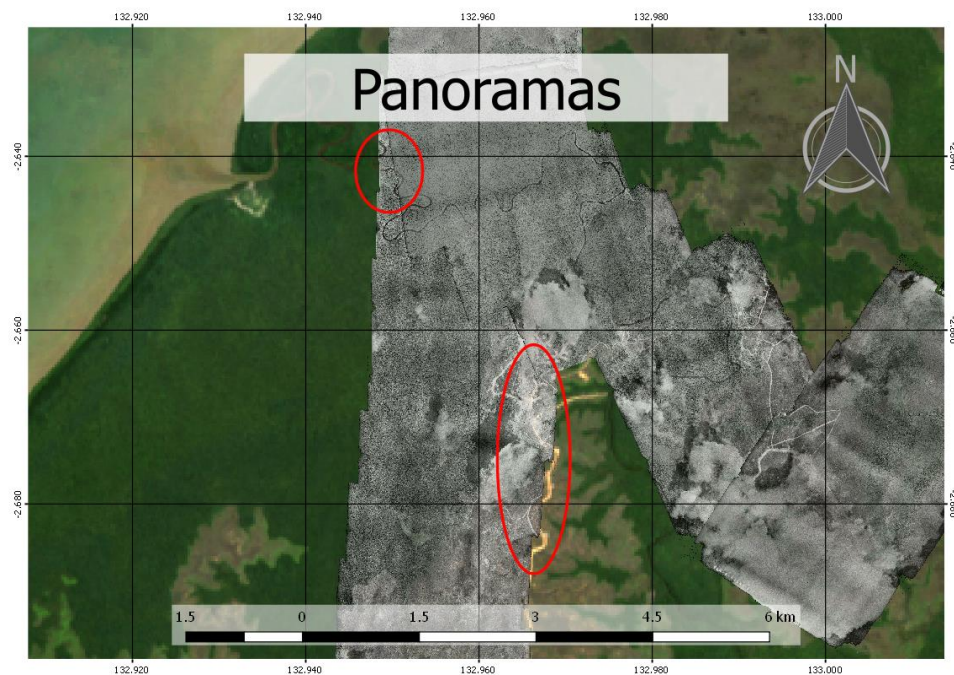


Figure 43 Errors in georeferencing of panoramas 2, 3 and 4

The errors here are between one and one hundred meters. This is quite a big error and range in error, but this can be explained by looking at the features we are comparing. To validate the panoramas, we look at the locations of edges of forest, coast lines, rivers and roads, which are all known to change over time. This means that there are no stable points to compare and the true accuracy does not have to be the same. It is also visible that the overlapping panoramas do not fully match. This can be caused by the panorama software not creating the perfect panorama or by the fact that again for the georeferencing of both panoramas different unstable points are used.

To validate the classifications, first shapefiles are made in QGIS. These shapefiles are created so that they only contain pixels from a certain class. Multiple shapefiles per class are created so that plenty of reference pixels are available and they are spread out over the total area. The pixels in these shapefiles are then clipped and exported so that the data can be used in Matlab. In Matlab the number of pixels that are classified as a certain class per class are computed. These values give the confusion matrix, which is then used to compute accuracy of the classification. In Table 6 the confusion matrix of the Maximum likelihood classification is given.

		Classified						
		Forest	Water	Clouds	Grass	Airport	Roads	Total
True	Forest	903	2	36	160	16	83	1200
	Water	170	598	109	247	75	1	1200
	Clouds	0	0	1077	5	108	10	1200
	Grass	36	258	103	622	63	118	1200
	Airport	30	35	419	150	474	92	1200
	Roads	420	0	80	86	124	490	1200
	Total	1559	893	1824	1270	860	794	7200

**Table 6 Confusion matrix of Maximum likelihood algorithm**

From these values the accuracy of a certain class is then computed with the following formula (Markham, 2014):

$$Accuracy = \frac{TrueClass}{Total}$$

Where *TrueClass* are all pixels correctly classified and *Total* is all pixels used in the validation.

For the Forest class this would give the following accuracy:

$$Accuracy = \frac{903}{1200} = 0.75$$

This was the accuracy of different classes can be compared and it can be found for what classes optimization in trainings data might be required. For the Maximum likelihood the accuracy per class is shown in Table 7.

Class	Accuracy
Forest	0.753
Water	0.498
Clouds	0.898
Grass	0.518
Airport	0.395
Roads	0.408

**Table 7 Accuracy per class for Maximum likelihood**

Here it can be found that the classes Airport and Roads are the least accurate. This is as expected, since these are small classes that have a limited number of pixels present in the dataset, which makes it harder to find enough training data.

When all classes are combined the accuracy of the full dataset can be computed using a slightly modified version of the formula:

$$totalAccuracy = \frac{sum(TrueClass)}{Total}$$

Where *TrueClass* is the number of pixels correctly classified for a certain class and *Total* is the total number of pixels used in the validation of the method.

For the Maximum likelihood algorithm this would give the following accuracy:

$$totalAccuracy = \frac{903 + 598 + 1077 + 622 + 474 + 490}{7200} = \frac{4164}{7200} = 0.578$$

Besides the accuracy, the method can also be validated with the kappa value. This compares the accuracy of the method to how well it would have performed simply by chance. (Markham, 2014) (Confusion Matrix - Another Single Value Metric - Kappa Statistic, 2011)

$$Kappa = \frac{totalAccuracy - randomAccuracy}{1 - randomAccuracy}$$

Where *totalAccuracy* is the accuracy of the method as computed before and the *randomAccuracy* is...

The random accuracy is computed as follows:

$$randomAccuracy = \frac{\sum(Actual_i * Predicted_i)}{Total^2}$$

Where *Actual* is all pixels in a certain class *i*, *Predicted* is all pixels classified as a certain class *i* in all classes and *Total* are all pixels used in the validation of the method.

The kappa of the Maximum likelihood is therefore computed as follows:

$$\begin{aligned} randomAccuracy &= \frac{\sum(Actual_i * Predicted_i)}{Total^2} \\ &= \frac{1200 * 1559 + 1200 * 893 + 1200 * 1824 + 1200 * 1270 + 1200 * 860 + 1200 * 794}{7200^2} \\ &= \frac{8640000}{51840000} = 0.1667 \\ Kappa &= \frac{0.578 - 0.1667}{1 - 0.1667} = \frac{0.4113}{0.8333} = 0.4936 \end{aligned}$$

The total accuracy and kappa values are computed for all three methods and shown in Table 8.



Method	Total accuracy	Kappa
Minimum distance	0.467	0.360
Maximum likelihood	0.578	0.494
Spectral angle	0.331	0.197

**Table 8 accuracy and kappa values of the three classification methods compared**

From these values it becomes clear that Maximum likelihood managed to produce the best result with the given data, but all methods fail to produce a truly reliable classification.

## 6.2 Landsat and Sentinel

The same method used for the validation of the panorama classification can be used to validate the Landsat and Sentinel classifications. In Table 9 the confusion matrix for the Landsat data is shown.

		Classified								
		Water	Buildings	Airport	Roads	Grass	Plantation	Forest	Open area	Total
True	Water	600	0	0	0	0	0	0	0	600
	Buildings	0	19	2	1	0	0	0	1	23
	Airport	0	0	220	7	0	0	0	89	316
	Roads	0	8	61	72	2	5	1	19	168
	Grass	0	20	27	82	265	144	24	18	580
	Plantation	0	7	6	7	5	462	108	5	600
	Forest	2	0	0	0	0	5	593	0	600
	Open area	2	4	47	3	5	2	0	537	600
	Total	604	58	363	172	277	618	726	669	3487

**Table 9 Confusion matrix for the Landsat data**

Most areas have around 600 pixels of validation data. The classes Buildings, Airport, Roads and Grass have less data than the others. This is because these classes are not that common in the area and there simply is not enough data available. From this data then the accuracy per class can be computed. The results are shown in Table 10.

Class	Accuracy
Water	1.000
Buildings	0.826
Airport	0.696
Roads	0.429
Grass	0.457
Plantation	0.770
Forest	0.988
Open area	0.895

**Table 10 Accuracy per class for the Landsat classification**

Here it can be seen that Roads and Grass are hard to detect using this training dataset, while Water Forest and Open area have a very high accuracy.

Lastly the total accuracy and kappa values can be computed for the Landsat 8 data, the averaged Sentinel-2 data and Sentinel-2 with the cloud filter applied. The results are shown in Table 11.

	Total accuracy	Kappa
Landsat 8	0.794	0.756
Averaged Sentinel-2	0.667	0.606
Sentinel-2 with cloud filter	0.605	0.527

**Table 11 Total accuracy and kappa values for the different datasets**

It is clear that the classification of the Landsat 8 data worked best. This is because the cloud filter in Sentinel-2 does not work as good as the cloud filter for Landsat 8. Overall the modern multispectral data is classified much more accurately than the World War II photos. This is more likely to be caused by more distinct features in the bands compared to the artificial bands than the difference in classification method.

# 7 Conclusion

The main goal of this assignment was to find locations that can possibly contain equipment left behind when World War II ended. With the given datasets some locations were found but, since this cannot be validated, it is unknown if the goal is reached. After the expedition this will become clearer and the new information will also enable improvement of the classification for successive searches.

The main question to be answered was to see if it was possible to link historical data to modern data and classify it. Linking historical data is possible, since creating panoramas and georeferencing them worked out, although some errors can be found. These errors can be created in the process, but could also be caused by the fact that the data is over seventy years old and the landscape has changed over time. There is little that can be done about this, since most of what is shown on the photographs looks different now or is gone. Another point that can be made is that the georeferencing is accurate enough for its intended usage.

Classifying the black and white images proved to be more difficult and the resulting classifications are not very accurate. If given more time I would consider adding more features to the dataset, since in the end only four out of the nine features were usable. Increasing the number of features might make it possible to classify the panoramas more accurately. Since that is not the case now, I would advise against using the classified version of the dataset and instead stick to the black and white images, since these work perfectly fine.

Classifying the satellite imagery was more successful. With an accuracy for the classification of Landsat dataset of 79.4%, the classification might be usable in the field. As expected the Landsat 8 data had a higher accuracy than the Sentinel-2 data. This can be explained by the very accurate cloud filter that is available for Landsat but not for Sentinel-2. It is unfortunate, since the higher spatial resolution of Sentinel-2 should make it easier to spot smaller objects like planes. For future researches I would therefore advise to look into a better cloud filter for Sentinel-2 data over improving the Landsat classification, since the spatial resolution is a great benefit.

The thermal bands showed some possible interesting areas, but it is also clear that roads and houses are very warm so it does not automatically mean something interesting is present at those locations. A plane that is partially submerged in water or that is not exposed to direct sunlight might not heat up and will therefore not be visible.

A recurring problem is the fact that very little data from the field is available. Only a few locations from objects are known and all training data are picked by looking at Satellite photographs of the area. This makes it difficult to distinguish between types of vegetation, especially when the quality of the data is not constant over the region of interest. Having known points in the field would help to create more accurate training data and therefore more accurate classifications.

# Bibliography

- Baiocchhi, V., Lelo, K., Vittoria, M., & Mormile, M. (2013). Accuracy of different georeferencing strategies on historical maps of Rome. *Geographica Technica*. Retrieved April 12, 2018, from [https://www.researchgate.net/profile/Keti\\_Lelo/publication/235988132\\_Accuracy\\_of\\_different\\_georeferencing\\_strategies\\_on\\_historical\\_maps\\_of\\_Rome/links/00b4951540deb8db07000000.pdf](https://www.researchgate.net/profile/Keti_Lelo/publication/235988132_Accuracy_of_different_georeferencing_strategies_on_historical_maps_of_Rome/links/00b4951540deb8db07000000.pdf)
- Brown, M., & Lowe, D. G. (2006, December). Automatic Panoramic Image Stitching using Invariant Features. *International Journal of Computer Vision*. Retrieved April 12, 2018, from <https://link.springer.com/content/pdf/10.1007%2Fs11263-006-0002-3.pdf>
- Brownlee, J. (2016, April 08). *Classification And Regression Trees for Machine Learning*. Retrieved April 12, 2018, from Machine Learning Mastery: <https://machinelearningmastery.com/classification-and-regression-trees-for-machine-learning/>
- Confusion Matrix - Another Single Value Metric - Kappa Statistic*. (2011, December 29). Retrieved April 15, 2018, from Standard Wisdom: <http://standardwisdom.com/softwarejournal/2011/12/confusion-matrix-another-single-value-metric-kappa-statistic/>
- Cordell, M. (2013, August 29). *The West Papuan independence movement - a history*. Retrieved April 09, 2018, from The Guardian: <https://www.theguardian.com/world/2013/aug/29/west-papua-independence-history>
- ESA. (n.d.). *Cloud Masks*. Retrieved April 09, 2018, from ESA: <https://sentinel.esa.int/web/sentinel/technical-guides/sentinel-2-msi/level-1c/cloud-masks>
- ESA. (n.d.). *SENTINEL-2*. Retrieved April 09, 2018, from ESA: <https://sentinel.esa.int/web/sentinel/missions/sentinel-2>
- Faymer, S. (2007, January). *Western New Guinea*. Retrieved April 09, 2018, from Wikipedia: [https://en.wikipedia.org/wiki/Western\\_New\\_Guinea](https://en.wikipedia.org/wiki/Western_New_Guinea)
- Free West Papua Campaign. (n.d.). *Free West Papua Campaign*. Retrieved April 09, 2018, from [www.freewestpapua.org](http://www.freewestpapua.org): <https://www.freewestpapua.org/info/history-of-west-papua/>
- Google. (n.d.). *Sentinel-2 MSI: MultiSpectral Instrument, Level-1C*. Retrieved April 09, 2018, from Google Earth Engine: <https://explorer.earthengine.google.com/#detail/COPERNICUS%2FS2>
- Google. (n.d.). *USGS Landsat 8 Surface Reflectance Tier 1*. Retrieved April 09, 2018, from Google Earth Engine: [https://explorer.earthengine.google.com/#detail/LANDSAT%2FLC08%2FC01%2FT1\\_SR](https://explorer.earthengine.google.com/#detail/LANDSAT%2FLC08%2FC01%2FT1_SR)
- Gorlinski, V. (n.d.). *West Papua*. Retrieved April 09, 2018, from Encyclopedia Britannica: <https://www.britannica.com/place/West-Papua>
- Markham, K. (2014, March 25). *Simple guide to confusion matrix terminology*. Retrieved April 15, 2018, from Data school: <http://www.dataschool.io/simple-guide-to-confusion-matrix-terminology/>
- NASA. (n.d.). *Landsat 8 Bands*. Retrieved April 09, 2018, from Landsat Science: <https://landsat.gsfc.nasa.gov/landsat-8/landsat-8-bands/>
- NASA. (n.d.). *Landsat 8 Overview*. Retrieved April 09, 2018, from Landsat Science: <https://landsat.gsfc.nasa.gov/landsat-8/landsat-8-overview/>
- Stichting Papua Erfgoed. (n.d.). *New-Guinea during World War II*. Retrieved April 09, 2018, from Papua Erfgoed: [http://www.papuaerfgoed.org/en/New-Guinea\\_during\\_World\\_War\\_II](http://www.papuaerfgoed.org/en/New-Guinea_during_World_War_II)
- USGS. (n.d.). *What is the C Function of Mask (CFMask) algorithm?* Retrieved April 09, 2018, from Landsat Missions: <https://landsat.usgs.gov/what-cfmask>