Segmentation of permanent forests and plantation

Marine Mercier
marine.mercier@telecom-paris.fr

Fabien Merceron
fabien.merceron@ens-paris-saclay.fr

Abstract

Satellite imagery and machine learning provide great monitoring of forest evolution. These new technologies help to address environmental issues such as the fight against deforestation or estimation of the carbon stocked in a wooded area. A clustering model able to differentiate dense primary forests, less dense forests and palm, rubber or cocoa plantations, would help to protect these fragile and vital environments.

1 Introduction

The objective of this project is to develop a forest segmentation method. It is proposed by Kayrros which aims to differentiate the different types of forest and crops in tropical regions based on satellite images. The goal is to develop a model able to classify the pixels of a remote-sensing image into classes corresponding to the type of vegetation.

To tackle this problem, Kayrros provided us, for each of the 4 areas of interest :

- A time series of Sentinel-1 radar satellite images. It corresponds to 60 images acquired over a period of one year. For each image, we have access to the VH and VV polarizations.
- A couple of Sentinel-2 images. We have far fewer Sentinel-2 than Sentinel-1 images because Sentinel-2 images are more difficult to acquire due to meteorological conditions which is particularly problematic in tropical regions.
- A *kml* file containing a manual annotations of the types of vegetation or a segmentation realised by Kayrros.

As we only have at our disposal 5 ground-truth images, we decided to opt for an unsupervised approach. The model must be able to perform a pertinent clustering for different goals depending on the number of classes specified. For example, if we want to distinguish only two classes, the model should discriminate the forest and non forest pixels. While if we specify a higher number of classes, the model should be able to classify the different types of vegetation (palm oil, natural forest, rubber trees...)

2 Methodology

In the first stage, we focus on the SAR images provided by Sentinel-1.

2.1 Preprocessing

Raw SAR images are really noisy and thus require some pre-processing steps before being exploited. For a SAR image, the range of values is large (some points are very bright meaning they have a very large back-scattering coefficient) and is different between all the images. In order to get a similar range of values across all the images, we adopted a min/max normalization. Let m and M be respectively the minimum and maximum pixel value of the image. The normalized version of the image is obtained by taking :

$$image_{normalized} = \frac{image - m}{M - m}$$

2.2 Denoising SAR images

Now images are in the same range but are still noisy. We first tried a spatial denoising using SAR2SAR, a deep learning algorithm trained to denoise SAR images. SAR2SAR algorithm is based on a U-Net and is trained in a semi-supervised fashion to restore SAR images [3]. However, we did not manage to fully exploit the algorithm and the results are therefore poor.

Then, we realised a temporal denoising. As we have a time series of Sentinel-1 images acquired over a period of one year, we are able to perform a temporal multi-looking by averaging images of the same site. However, most of the image correspond to vegetation and since the vegetation changes with the season, we can't directly average the whole time series. So we decided to restrict to the averaging of the images within a window of one-quarter. Since we have on average 15 images per quarter per site, the temporal multi-looking enable to denoise the images fairly well. The result obtained after each step of the denoising is displayed in Figure 1.

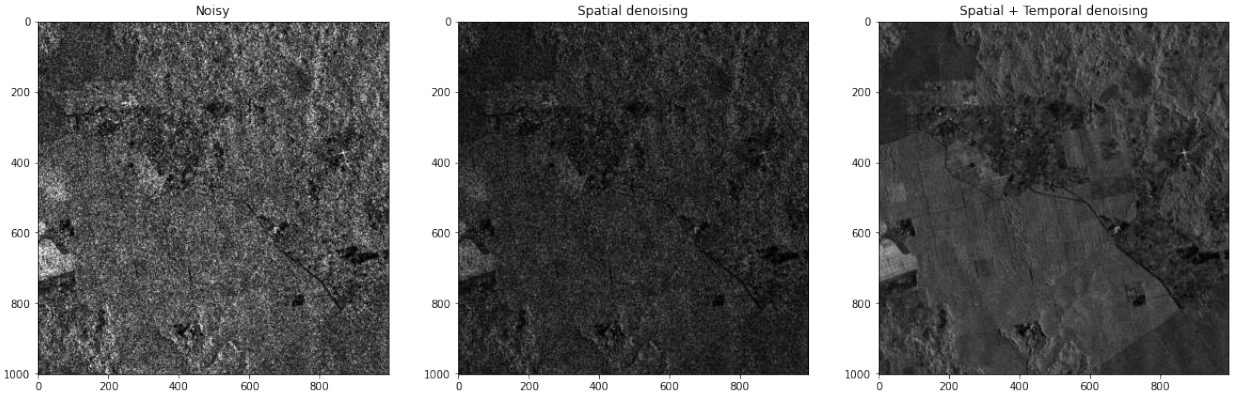


Figure 1: Result obtained on an image from site 1 after all the denoising steps

This denoising step has been applied both on VV and VH polarizations. A good way to combine the information brought by VV and the information brought by VH into a single image is to create an image with 3 channels (VV, VH, VV/VH). It helps to visualize the SAR image with coherent colors for human eyes (Figure 2)

2.3 Superpixels

2.3.1 Need of superpixels

A naive approach would be to directly apply a clustering algorithm on the pixels of the image with 3 channels. Figure 3 shows the results obtained if we try to apply K-Means algorithm directly on

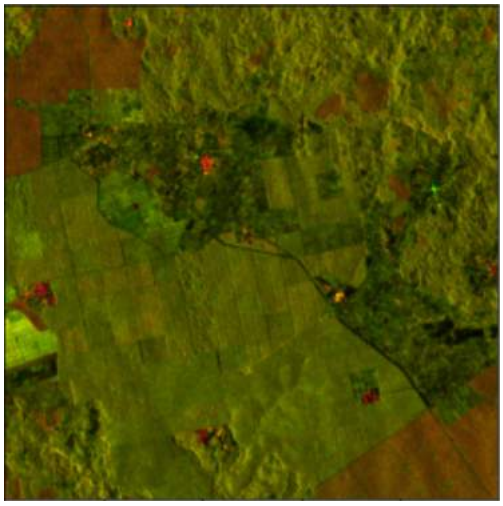


Figure 2: Image obtained when considering 3 channels (VV, VH, VV/VH)

pixels. On this image, we are already able to identify some areas with different types of vegetation. For example the crops appearing in brown on Figure 2 which are actually palm oil trees. These fields are human made and we imagine that the area is completely covered of palm oil trees with no other type of vegetation left inside. However K-Means algorithms classified many points in these fields as other types of vegetation. This issue led us to leverage a specificity of the image we are dealing with : in human exploitation's, trees are usually planted in groups. This point led us to consider superpixels.

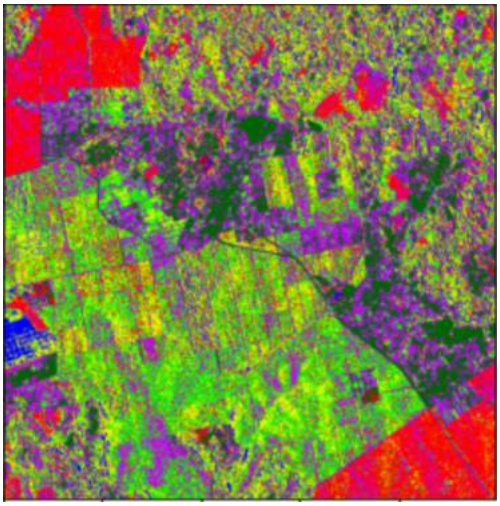


Figure 3: Clustering of pixels using K-Means with K=6

Superpixels algorithms aim to segment an image into regions by merging pixels with common characteristics. They are useful in object detection and help to reduce the number of data to deal with.

2.3.2 Superpixel Methods

We have tested several superpixels algorithms namely SLIC [2], Watershed [5], Quickshift [1] and Felzenszwalb [6]. They are all based on different techniques.

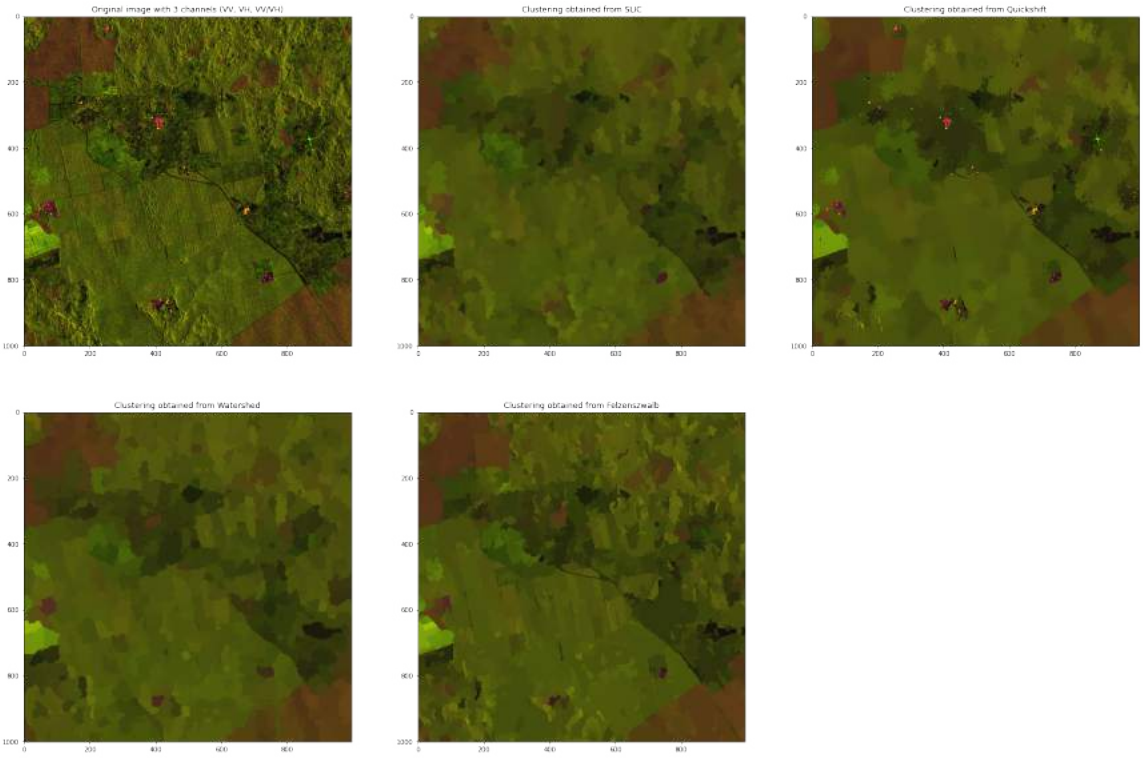


Figure 4: Segmentation produced by the different superpixels algorithm, each output is composed of 1000 superpixels

All these algorithms require parameters tuning. Results are shown in Figure 10. A good superpixel segmentation for our problem would be one where the geometry of the objects is respected and a maximum of the information has been preserved. In other words, pixels belonging to different objects are not merged, but pixels within uniform regions are merged. Using these criteria, and looking at the every site, we judged that Felzenszwalb's segmentation provide the best results. Indeed, they are the only algorithms that respect properly the shape of the crop fields. Given that Quickshift algorithm does not give very good results and is very long to run, we did not keep this method. For the other methods, we will describe the algorithm and the parameters used. We conducted two experiments on the original image 5, a segmentation in approximately 500 superpixels, and a segmentation in approximately 2000 superpixels.

2.3.3 SLIC Superpixels

Simple Linear Iterative Clustering Algorithm (SLIC) is a K-means algorithm performed with 5 dimension : 3 dimensions of color information (R,G,B) and 2 dimension related to pixels location (x,y). First, each pixel is projected in a 5 dimension space. Then, we sample K regularly spaced cluster centers, and move them to the lowest intensity gradient position in a local (3×3) neighbor-

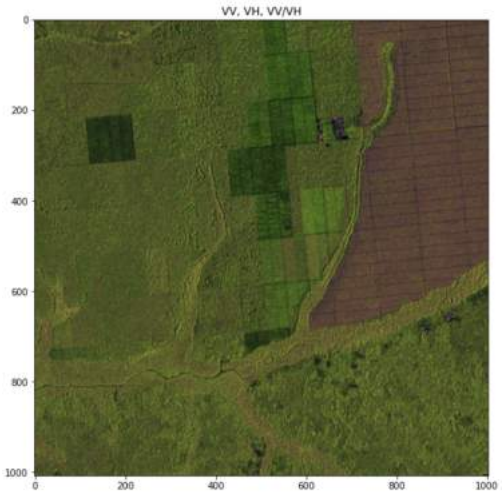
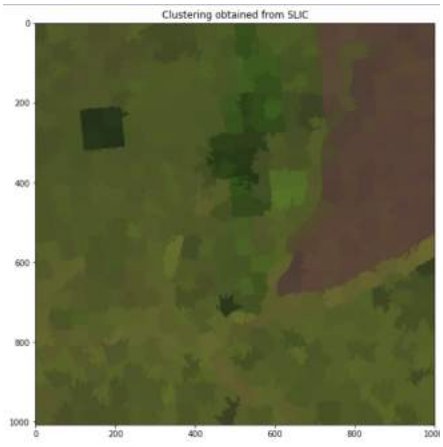


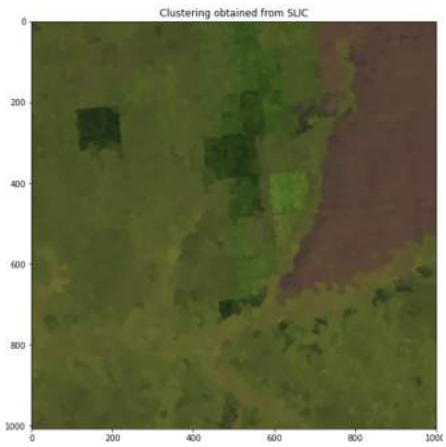
Figure 5: Original image

hood. Last, we use the Kmean algorithm on the given centers. The final classes of the K-mean are divided in connected components, which are the final superpixels.

Two parameters are important in SLIC algorithm, the number of segments and compactness. The number of segments is the expected number of superpixels in the segmented output image while compactness balances color proximity and space proximity. Higher compactness gives more weight to space proximity, making superpixel shapes more cubic. The first segmentation, with $n_segments = 800$ and $compactness = 15$gave 499 superpixels, while $n_segments = 5000$ and $compactness = 10$gave 2105 superpixels.



Output with 499 superpixels



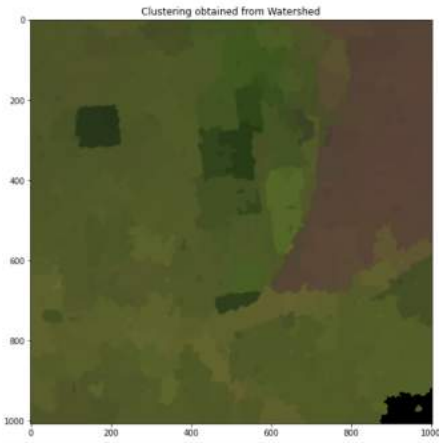
Output with 2105 superpixels

Figure 6: SLIC superpixels decomposition

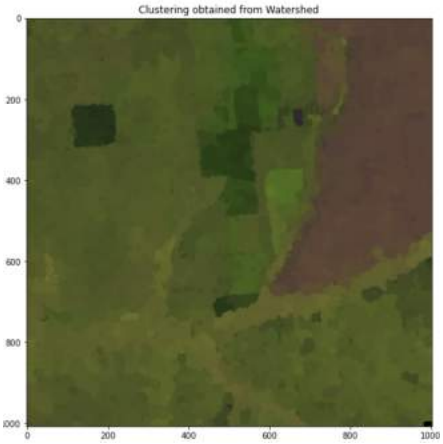
2.3.4 Watershed Superpixels

The watershed algorithm computes the segmentation based on the image flooding from given markers. The algorithm treats pixels as an elevation, and the image as a topography map. Then we

flood bassins from the markers until water attributed to different bassins meet on a watershed line. These lines define the contours of superpixels.



Output with 484 superpixels



Output with 2116 superpixels

Figure 7: Watershed superpixels decomposition

The Watershed algorithm also has two parameters, *markers* representing the number of expected clusters, and compactness defined as previously. In the first experiment we used $markers = 500$, $compactness = 0.0001$ and got 484 superpixels, for the second, we used $markers = 2000$ and $compactness = 0.00001$ to obtain 2116 superpixels.

2.3.5 Felzenszwalb Superpixels

Felzenszwalb's algorithm is a graph-based approach to segmentation. The image to segment is considered as an undirected graph G = (V,E) where the vertices V correspond to the pixels of the image and E is the set of weighted edges linking the pairs of connected pixels. $\forall i, j$, if the edge (v_i, v_j) exists, it is weighted according to some measure of the dissimilarity between the two pixels v_i and v_j. The segmentation aims to cut the graph G into distinct components. The edges should have low weights within a component and large weights between components. Several measure of dissimilarity can be used but in this situation, each pixel is mapped into a feature space (x,y,r,g,b) where (x,y) are the coordinates of the pixel in the image and (r,g,b) the color value of each channel for a RGB image. The euclidean distance between v_i and v_j in the feature space gives the measure of dissimilarity between v_i and v_j and therefore the weight of the edge (v_i, v_j). The algorithm requires to first introduce some definitions. The minimum internal difference M_{int} between a couple of components C_1, C_2 is defined as :

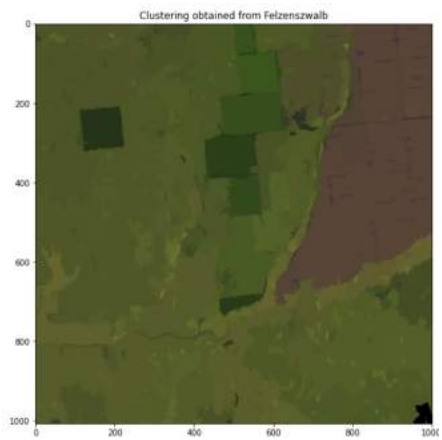
$$M_{int}(C_1, C_2) = min(Int(C_1) + \tau(C_1), Int(C_2) + \tau(C_2))$$

where $Int(C_i)$ is the internal different of the component C_i i.e. the maximum weight of an edge within the component C_i and $\tau(C_i) = k/|C|$ with k a chosen parameter and $|C|$ is the cardinal of C. The segmentation algorithm works as follows :

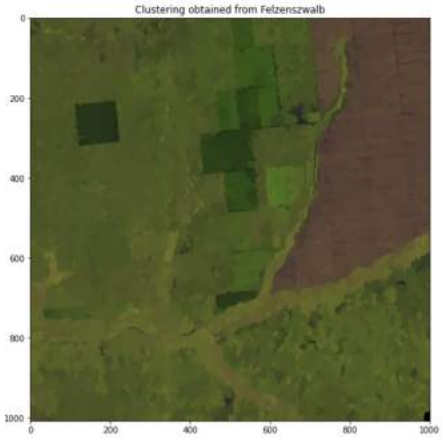
1. At the initialisation step, each pixel belongs to its own component.
2. Repeat step 3 for $q = 1, ..., m$

3. Let C_i^{q-1} be the component of S^{q-1} containing v_i and C_j^{q-1} the component containing v_j and $w(v_i, v_j)$ be the weight associated to the edge (v_i, v_j). If $C_i^{q-1} \neq C_j^{q-1}$ and $w(v_i, v_j) \leq \mathrm{M_{int}}\left(C_i^{q-1}, C_j^{q-1}\right)$ then S^q is obtained from S^{q-1} by merging C_i^{q-1} and C_j^{q-1}. Otherwise $S^q = S^{q-1}$.

The main parameter to tune in Felzenszwalb's algorithm is k. The larger is k, the less and bigger superpixels are returned by the algorithm. The value of k needs to be chosen depending on the image. For example on the image of the Site 1 (image of the Figure 10), k was set to 40. But we could imagine that images containing much larger crop fields would require a smaller value of k. In the skimage implementation, k is called *scale*. We also used the parameters *sigma*, the width of the gaussian kernel used in preprocessing and min_size, the minimum size for each component, which is enforced in postprocessing. With $scale = 100$, $sigma = 1$, $min_size = 100$, we obtained 559 superpixels, while with $scale = 30$, $sigma = 1$, $min_size = 100$, we had 2190 superpixels.



Output with 559 superpixels



Output with 2190 superpixels

Figure 8: Felzenszwalb superpixels decomposition

2.3.6 Size of Superpixels

Felzenszwalb's algorithm gives the best results. We now want to use the pixels in each superpixel to extract relevant features. However, the choice of the size of the superpixels is crucial for the rest of our problem. Smaller superpixels preserve detail and edges better, while larger superpixels give more reliable statistics. The number of superpixels we chose per site, summarized in the table below, does not seem to be proportional with the size of the image.

Site	Size of the image	Number of superpixels chosen
1	(1002, 996)	520
2	(510, 510)	336
3	(1008, 1002)	742
4	(996, 996)	647

2.4 Sentinel-2 vegetation indices

After exploiting the SAR images of every sites, we interested in optical images provided by Sentinel-2. Unlike for Sentinel-1, we only have a couple of optical images per site due to the difficulty of

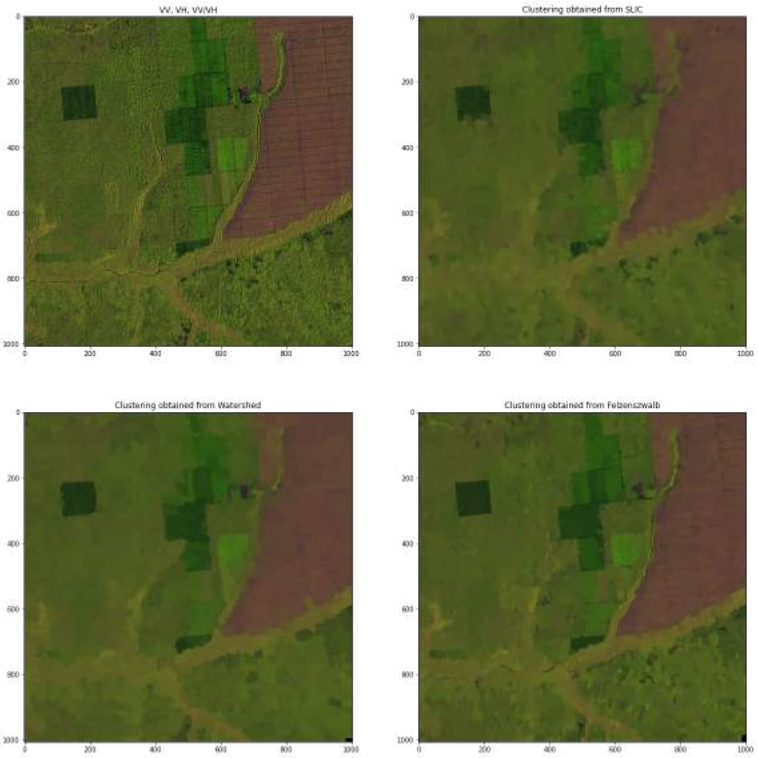


Figure 9: Segmentation with approximately 2000 superpixels

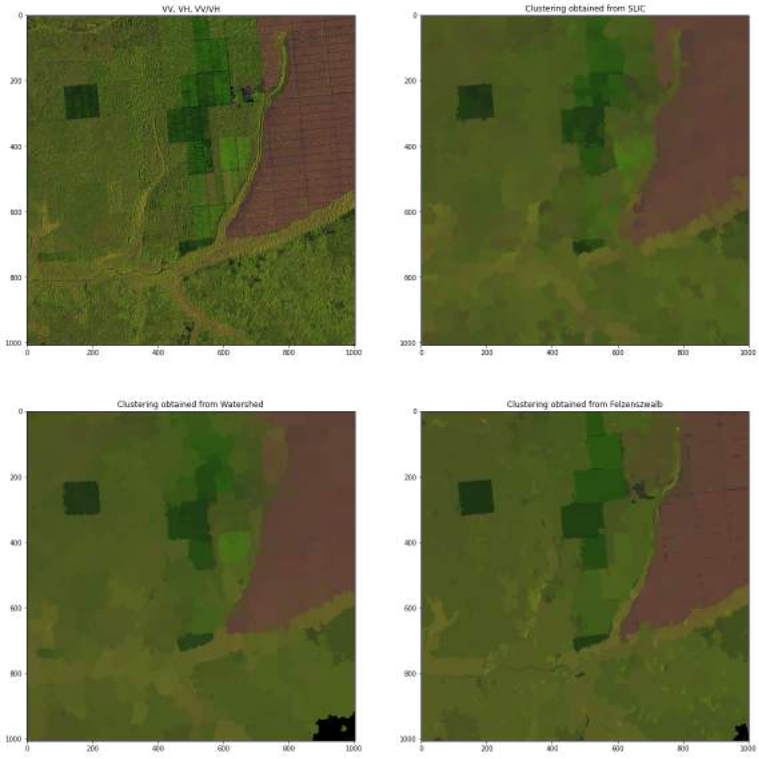


Figure 10: Segmentation with approximately 500 superpixels

acquiring optical images in non-optimal meteorological conditions. However, optical images are of high quality so we don't need to perform any temporal averaging.

Our first idea was to perform some superpixel segmentation as we did for Sentinel-1 images and feed both the segmentation obtained from Sentinel-1 and Sentinel-2 into a clustering algorithm. However, this approach led to poorer results than clustering on Sentinel-1 images only.

Instead, we chose to use spectral indices that can be derived from the different band provided by Sentinel-2 and useful to monitor vegetation among others. Many indicators that could a priori help in our situation of dense and tropical vegetation have been tested :

- **Normalized Difference Vegetation Index (NDVI)** : NDVI is a a widely-used index to monitor vegetation mainly due to its robustness against noise and varying illumination geometries [4]. High NDVI values correspond to areas that reflect more in the near-infrared spectrum. Higher reflectance in the near-infrared correspond to denser vegetation. NDVI is computed as :

$$\frac{NIR - Red}{NIR + Red}$$

- **Bare Soil Index (BSI)** : BSI captures soil variations by quantifying the soil mineral composition, also taking into account the presence of vegetation. BSI is computed as :

$$\frac{Red + SWIR – (NIR + Blue)}{Red + SWIR + NIR + Blue}$$

- **Green Chlorophyll Index (GCI)**: GCI allows to estimate the content of leaf chlorophyll in various species of plants. The chlorophyll content decreases in stressed plants and can therefore be used as a measurement of plant health. GCI is computed as :

$$\frac{NIR}{Green} - 1$$

- **Moisture Stress Index (MSI)**: (MSI): Moisture Stress Index is used for canopy stress analysis. Higher values of the index indicate greater plant water stress and less soil moisture content. We can hope that this index helps to differentiate crop fields where the moisture level is probably monitored versus natural forests. MSI is computed as :

$$MSI = \frac{MidIR}{NIR}$$

2.5 Superpixel features

The information we have extracted from Sentinel-1 and Sentinel-2 images can be gathered before clustering the superpixels. We decided to apply some spatial filtering techniques to enhance the results. In particular, we used a gaussian smoothing as well as a Laplacian of Gaussian to detect the edges, on each channel of the (VV,VH, VV/VH) image and on each vegetation index individually. We applied the filters on the pixelized image but we computed the average values of the Gaussian smoothing over each superpixel in order to retrieve the superpixel geometry. Indeed, conserving the same superpixels for each feature instead of generating different superpixels for every feature is helpful for the stability of the clustering. For example, Figure 11 shows the superpixels previously built by Felzenszwalb's algorithm but colored using the NDVI index.

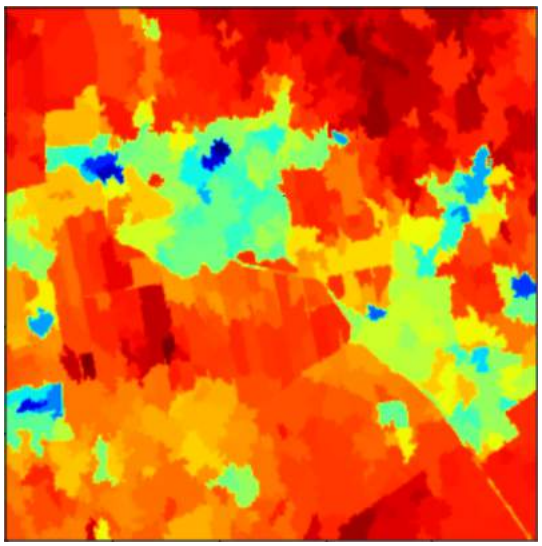


Figure 11: Superpixels from Sentinel-1 colored with the NDVI index

As a result, we have a stack of images that will be used as superpixel features for the clustering. It should be noted that we have also tried to include other superpixel features such as the variance of the superpixel because we thought it would help to differentiate uniform fields from natural forests. However, it didn't yield to better results. Multiple combination of the vegetation indices we presented in section 2.4 have been tested. The qualitative evaluation we conducted led to the conclusion that the best results were obtained when a combination of NDVI and NDMI was used in addition to the superpixel features from Sentinel-1.

2.6 Clustering

The dimension of the superpixel features space is N where N equals to 6 (3 channels of the Sentinel-1 image times 2 i.e. the number of filters) plus the number of vegetation indices considered. As we did for the pixelized image 3, we apply a K-Means clustering but now directly on superpixels and in a feature space of dimension N.

The choice of K, the number of classes depends on the image. Some images combine many types of vegetation while other images contain a limited number of different species. In order to determine the proper number of clusters in an unsupervised way, we computed the intra-class variance i.e. the sum of squared distances from each data point to its cluster center in the feature space :

$$\sum_{j=1}^{K} \sum_{x \in c_j} d(x, \mu_j)$$

where μ_j is the vector of coordinates of the center of the cluster j.

By plotting the intra-class variance for consectuive values of K, we are able to use the *elbow method*, a well-known method to choose the number of clusters. The elbow corresponds to the number of clusters from which the intra-class variance stops decreasing significantly. Figure 12 shows the intra-class variance we obtain on site 1 for various number of clusters. One can identify an elbow at $k = 4$ because after this value, the intra-class variance decrease is marginal. Therefore, we will keep a number of clusters equals to 4 for site 1.

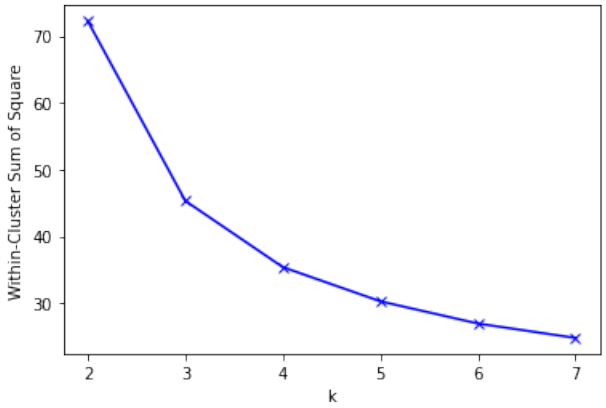


Figure 12: Intra-class variance obtained when applying a K-Means clustering the superpixel features of site 1.

3 Qualitative results

As a ground-truth is available for site 1, one can compare it to the K-Means clustering with 4 classes. The comparison is displayed in Figure 13. On the ground-truth image, the vegetation labelled in green corresponds to palm oil, the vegetation in yellow is uniform vegetation and the purple regions have no label. First of all, one can notice that the number of clusters is different between the ground-truth segmentation and our clustering. This is mainly due to fact that a group of crop fields on the left of the image appears much brighter than the other crops on the Sentinel-1 image and also appears differently on the NDVI image 11. Thus, it makes sense that this area forms its own cluster even though it is not labelled in the ground-truth. In addition, the crop fields in brown on the Sentinel-1 image that are actually palm oil are well detected in our clustering. The main shortcoming is to differentiate the uniform vegetation from other types of vegetation (not labelled in the ground-truth image). The results we obtained on the other areas of interest are displayed in appendix A. For the other areas of interest than site 1, we were not able to exploit the ground-truth files so instead we compared our clustering results to a segmentation realised by Kayrros.

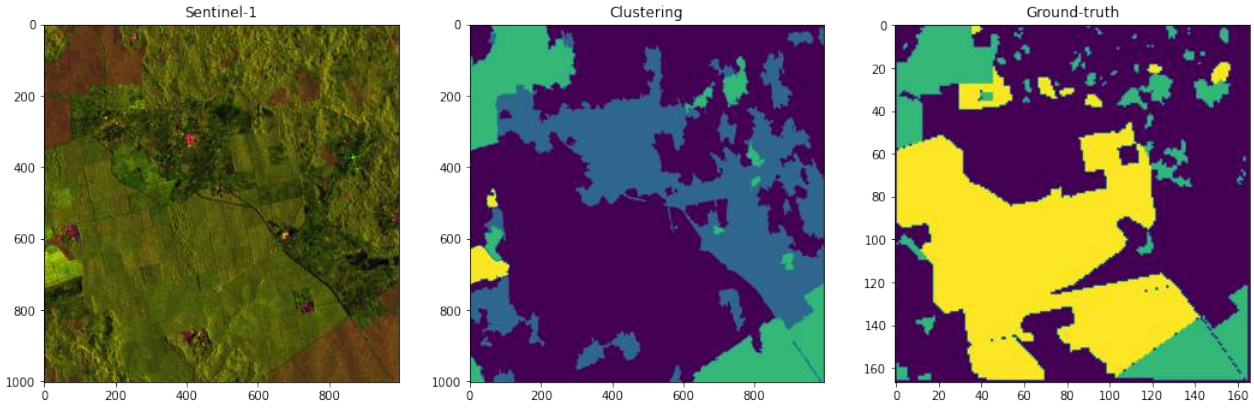


Figure 13: Qualitative results of the clustering on site 1

4 Conclusion and future work

The unsupervised clustering algorithm we developed during this projects gives satisfying results visually, but we couldn't compare our results with a gorundtruth. Therefore we could not evaluate or compare our models, and we had to make decisions based on visual rendering alone. We noticed that the quality of the final clustering relies heavily on the superpixels quality. The parameter finetuning for superpixels is done manually, and depending on the studied area these parameters, including the ideal number of superpixels, may change. This is a major shortcoming of our solution, since manual setup is time consuming.

To further develop this method we could look for a metric to evaluate the performance of the models. This would help to motivate the decisions made during the development process. Then, we could continue to improve superpixel features. The interest of using superpixels is to try to gain information from the statistics of the pixels included in each superpixel. To do this, we could use different indicators concerning the distribution of the pixels that we did not have time to explore (like histograms, quantiles or the Gray Level Co-Occurrence Matrix). We could also study the images over a longer period of time as we focused on a quarterly study. Using a longer period would allow us to exploit the information of the changing seasons.

5 Acknowledgement

We want to acknowledge Martin Esguerra and Aurelien De Truchis from Kayrros as well as Enric Meinhardt-Llopis for their time and advices.

References

[1] Vedaldi A. and Soatto S. Quick shift and kernel methods for mode seeking. *European Conference on Computer Vision*, 2008.

[2] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Suesstrunk. Slic superpixels compared to state-of-the-art superpixel methods. *TPAMI*, 2012.

[3] Emanuele Dalsasso, Loïc Denis, and Florence Tupin. Sar2sar: A semi-supervised despeckling algorithm for sar images. *IEEE J. Sel. Top. Appl. Earth Obs. Remote. Sens*, 14:4321–4329, 2021.

[4] H. Jin and L. A Eklundh. Physically based vegetation index for improved monitoring of plant phenology. *Remote Sens. Environ.*, 152:512–525, 2014.

[5] Peer Neubert and Peter Protzel. Compact watershed and preemptive slic: On improving trade-offs of superpixel segmentation algorithms. *ICPR*, 2014.

[6] Felzenszwalb P.F. and Huttenlocher D.P. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 2004.

Appendices

A Results on the other areas of interest

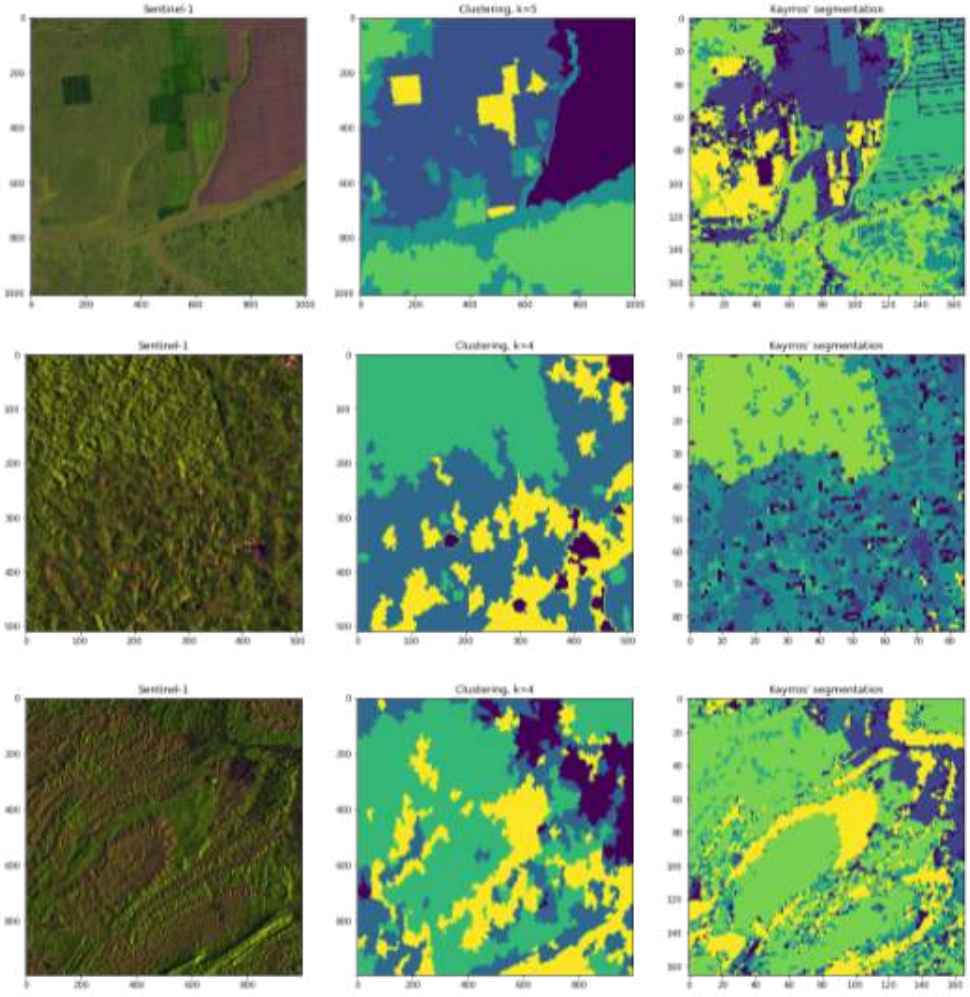


Figure 14: Clustering results the other areas of interest. From top to bottom : site 2, site 3 and site 4.