

Sentinel-2 Agriculture

Design Definition File




Algorithm Theoretical Basis Document for L4 dynamic crop mask product



Milestone	Milestone 2
Version	1.2
Authors	CESBIO - Silvia Valero, David Morin UCL - Guadalupe Sepulcre, Nicolas Matton, Sophie Bontemps, Nicolas Bellemans

© UCL-Geomatics 2015

This document is the property of the Sen2-Agri partnership, no part of it shall be reproduced or transmitted without the express prior written authorisation of UCL-Geomatics (Belgium).

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 
	Issue	Page	Date	
	1.rev.2	2	17/02/2016	

This page is intentionally blank



	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	3	17/02/2016	

Table of recorded changes

Issue record sheet

Issue/Rev.	Date	Reason
1.0	19/04/2015	1 st version for ESA
1.1	01/07/2015	2 nd version to ESA
1.2.	17/02/2016	3 rd version to ESA. Updated version according to the Actions Items and to the algorithms improvements made during the system development phase.

Detailed record sheet

From version 1.1 to 1.2

RID	Section	Problem description	Change
/	1.		Modification of the Flowchart (Figure 1-1.) and his text description. Both crop mask (with and without in situ data) have now a different gap filling step.
	2.1		In the previous release, the gap filling was a common step. Previous section 2. is now incorporated in the former section 3. as section 2.1.
	3.1.		The smoothing of the data is now joined to the gap filling step using the Whitaker filter. Processing details added.






	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	4	17/02/2016	

Table of contents



1	Logical model	9
2	Mode based on in-situ data	11
2.1	Preliminary step : gap filling	11
2.2	Pre-processing: spectral indices computation	13
2.3	Feature extraction step	14
2.3.1	Temporal features	14
2.3.2	Statistic features.....	19
2.3.3	Feature image concatenation	19
2.4	Classification step	20
2.4.1	Constructing the classification model.....	20
2.4.2	Classification of the image times series : Random Forest.....	26
2.5	Processing chain working in the iterative operational model.....	27
2.5.1	The first year of Sentinel-2 images	28
2.5.2	The rest of the years	28
2.6	Processing chain model working for a large area covering several satellite footprints	29
3	Mode without in-situ data.....	31
3.1	Preliminary step: data smoothing and gap filling.....	31
3.2	Features extraction	33
3.2.1	Temporal features definition	34
3.2.2	Spectral features extraction	36
3.3	Reference map preparation.....	38
3.4	Trimming.....	39
3.5	Random forest classifier.....	44
3.5.1	RF model estimation.....	44
3.5.2	Classification	45
4	Common final steps	46
4.1	Classifier Evaluation: Computing evaluation quality measures	46
4.2	Crop mask post-filtering.....	47
4.2.1	Principal Component Analysis	48
4.2.2	Mean-Shift segmentation algorithm	48

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE 
	Issue	Page	Date	
	1.rev.2	5	17/02/2016	

4.2.3 Majority vote 49



4.2.4 Operational context 49

5 Appendix - Additional information on the codes of the « in-situ » processing chain 51

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	6	17/02/2016	




List of figures

Figure 1-1. Flowchart of the processing chain developed to generate the cropland mask	9
Figure 2-1. Flowchart describing the processing chain when field data exists	11
Figure 2-1. Gap filling workflow	12
Figure 3-2. Pre-processing workflow, computing NDVI, NDWI and Brightness indices	13
Figure 3-3. Temporal features generation workflow	14
Figure 3-4. Maximum NDVI values identification from a full NDVI time series	16
Figure 3-5. Identification of the largest increasing period in a full NDVI time series	17
Figure 3-6. Identification of bare soil transitions in a full NDVI time series	18
Figure 3-7. Statistic features generation workflow	19
Figure 3-8. Workflow of the temporal and statistic features concatenation	20
Figure 3-9. Workflow of the classification model construction	21
Figure 3-10. Workflow of the in-situ dataset splitting sub-task	21
Figure 3-11. Illustration of the in-situ dataset split into two subsets for training and validation using a ration equal to 0.5	22
Figure 3-12. Workflow for reducing the size of the training sample	23
Figure 3-13. Workflow of the classifier training	25
Figure 3-14. Workflow of the RF classification	27
Figure 3-15. “In-situ” classification chain running at instants T and T+1, when we are during the first S2 year	28
Figure 3-16. “In-situ” classification chain running in Year X-1, when we are not during the first S2 year	29
Figure 3-17. “In-situ” classification chain running in Year X, when we are not during the first S2 year	29
Figure 3-18. Crop mask production logical workflow when working over large areas	30
Figure 3-1. Workflow of the smoothing preliminary step	31
Figure 4-2. Workflow of the temporal and spectral features extraction process	33
Figure 4-3. Workflow of the reference map preparation	38
Figure 4-4. Workflow of the trimming process	41
Figure 5-1. Workflow of the classification assessment	46
Figure 5-2. Workflow of the post-filtering step	47
Figure 5-3. Post-filtering workflow, at instant T	50
Figure 5-4. Post-filtering workflow, at instant T+1	50

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	7	17/02/2016	



List of tables

Table 2-1. Variables for the linear interpolation and gap filling algorithm.....	12
Table 3-1. Variables for the spectral indices computation	14
Table 3-2. Variables for the spectral indices computation	15
Table 3-3. Global features description	15
Table 3-4. Description of features detecting largest local NDVI transitions	15
Table 3-5. Maximum NDVI features description.....	16
Table 3-6. Greenness period features description	17
Table 3-7. Senescence onset features description	18
Table 3-8. Bare soil transition features description.....	18
Table 3-9. Variables for the spectral indices computation	19
Table 3-10. Statistic features description	19
Table 3-11. Variables for the spectral features concatenation.....	20
Table 3-12. Variables for the splitting the field dataset for training and validation.....	22
Table 3-13. Variables for the splitting the field dataset for training and validation.....	24
Table 3-14. Variables for training the Radom Forest.....	26
Table 3-15. Variables for the RF classification algorithm	27
Table 4-1. Variables for the data smoothing	31
Table 4-2. Variables for the temporal features identification.....	34
Table 4-3. Variables for the spectral features extraction.....	36
Table 4-4. Variables for the erosion of the reference	38
Table 4-5. Variables for the trimming	41
Table 4-6. Variables for the RF model estimation in the « no in-situ » mode	44
Table 4-7. Variables for the RF classification for the « no in-situ » mode	45
Table 5-1. Variables for the map validation algorithm	47
Table 5-2. Variables for the Principal Component Analysis.....	48
Table 5-3. Variables for the Mean-Shift segmentation	48
Table 5-4. Variables for the majority voting filtering	49

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 
	Issue	Page	Date	
	1.rev.2	8	17/02/2016	

List of algorithms

Algorithm 2-1. Linear interpolation and gap filling.....	12
Algorithm 3-1. Spectral indices computation.....	14
Algorithm 3-2. OTB application performing the feature images concatenation	20
Algorithm 3-3. OTB application performing the feature images concatenation	23
Algorithm 3-4. Random selection of training samples.....	25
Algorithm 3-5. OTB application computing the vector image statistics	26
Algorithm 4-1. Smoothing based on the Whitaker function	32
Algorithm 4-2. Temporal features identification.....	35
Algorithm 4-3. Spectral features extraction	37
Algorithm 4-4. « SelectInList » program	38
Algorithm 4-5. Reference erosion	39
Algorithm 4-6. Trimming.....	42
Algorithm 4-7. MVG_IT function.....	43
Algorithm 5-1. Computation of quality indices	47
Algorithm 5-2. Majority vote filtering	49

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	9	17/02/2016	

1 Logical model

The classification process transform the L2A_R imagery into a 1/0 mask, 1 indicating the presence of cropland and 0 the presence of non-cropland.

The processing chain is run by tiles.

The processing chain can run in two different modes, depending if the user counts with field data for training the classifier or not:

- If the user has field data to train the algorithm, an extensive number of features related to the NDVI, the NDWI profiles and the brightness will be extracted from the imagery time series for feeding a supervised algorithm based on a Random Forest (RF) classifier;
- If there is no field data, training samples will be selected from an existing reference map (local/regional/global land cover map, crop mask, etc.) through an iterative trimming. These samples will be used to feed the same supervised algorithm than in the first case (i.e. based on a RF classifier) but the this classifier will here be run based on reflectance values coming from determinate moments of the growing cycle.

The choice between the two modes (“field data” and “no field data”) is made once for all the Area of Interest.

The principal steps of the classification process are schematized in Figure 1-1.

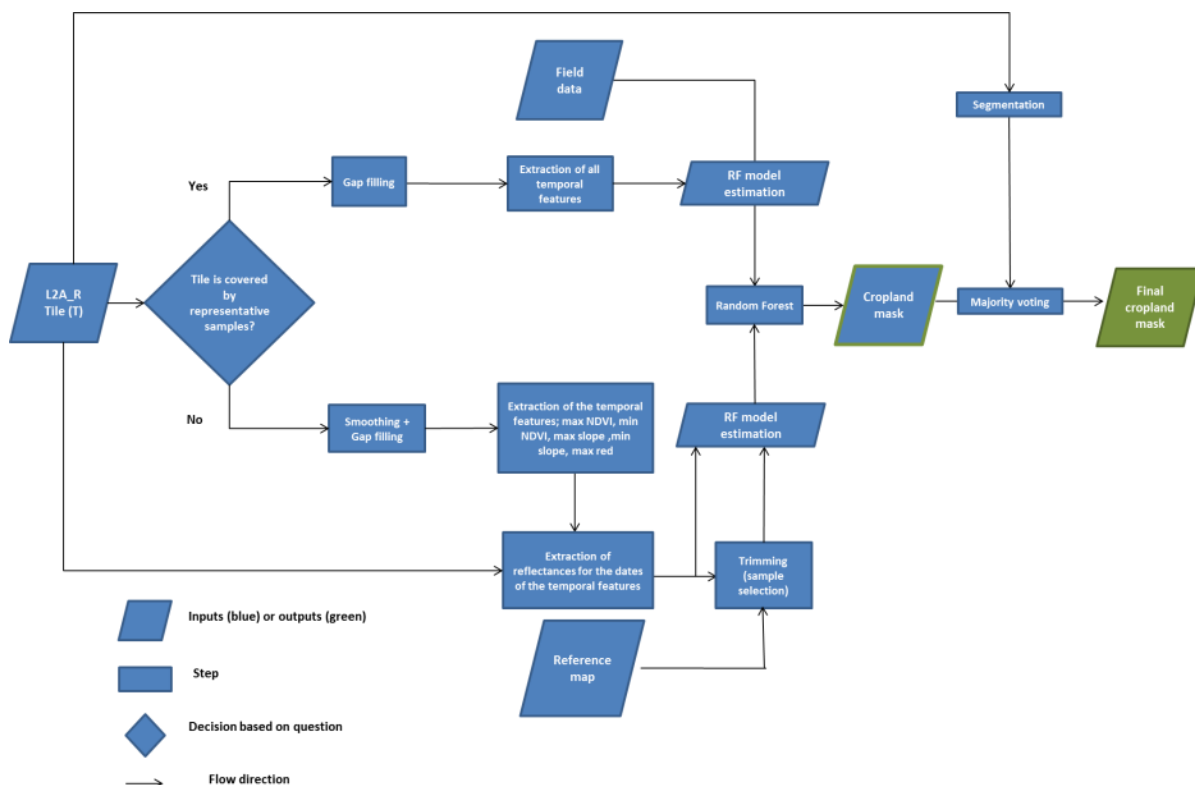







Figure 1-1. Flowchart of the processing chain developed to generate the cropland mask

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 
	Issue	Page	Date	
	1.rev.2	10	17/02/2016	

Two steps are common for the 2 modes:

- the supervised algorithm based on the RF classifier;
- the a posteriori filtering of the cropland mask.

The next sections of the document present in detail all the steps of the processing chain, distinguishing between the 2 modes and highlighting the common ones.

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	11	17/02/2016	

2 Mode based on in-situ data

The processing chain to run when field data exist is illustrated in **Erreur ! Source du renvoi introuvable.**, which must be interpreted from left to right. This flowchart can be easily divided in three main tasks: the feature extraction, the classification and the post-filtering step. The post-filtering step is common with the “no field data module” and will thus be detailed later, in section 4. All other steps are detailed in this section.

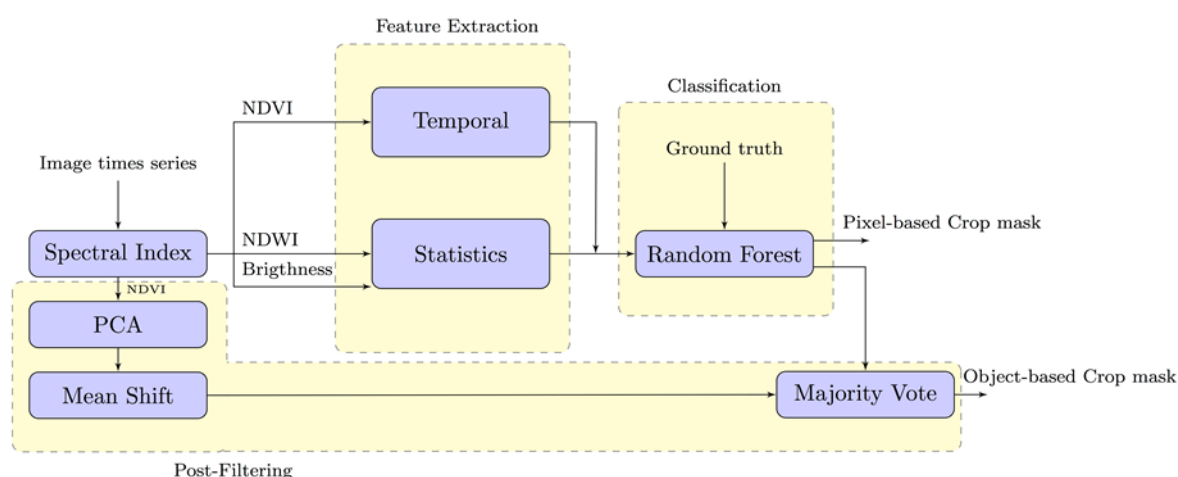
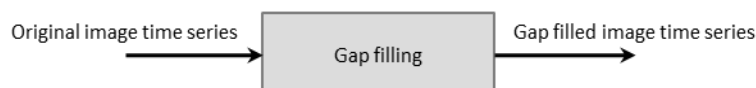


Figure 2-1. Flowchart describing the processing chain when field data exists

The classification chain uses as input the reflectance image times series that don't contain gaps (i.e. the output of the preliminary step detailed in section **Erreur ! Source du renvoi introuvable.**). As shown in **Erreur ! Source du renvoi introuvable.**, these time series will be used in the first purple block whose name is *Spectral Index*. This task corresponds to the computation of three spectral indices: the NDVI, the NDWI and the Brightness index (which is described in section 2.2). The output of the classification chain (i.e. before the common filtering step) is the pixel-based crop mask product will be the output of the classification chain.

2.1 Preliminary step : gap filling

The goal of the linear interpolation is to produce a reflectance image time series which is (i) gap-filled with respect to missing data (which can be due to clouds, cloud shadows and saturated pixels) and (ii) temporally sampled on a regular grid. The workflow of the gap filling is illustrated in Figure 2-2. If several sensors are available (for instance Sentinel-2 and LANDSAT-8) their respective reflectance time series are processed independently. These time series are afterwards concatenated before providing them to the classification step and being processed as single time series.



© UCL-Geomatics 2015

This document is the property of the Sen2-Agri partnership, no part of it shall be reproduced or transmitted without the express prior written authorisation of UCL-Geomatics (Belgium).



	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	12	17/02/2016	



Figure 2-2. Gap filling workflow

Algorithm 2-1 describes the procedure, using the parameters given in Table 2-1. In the case where the whole series is available, t_0 and t_end correspond to the first and last dates of the regular grid.

Table 2-1. Variables for the linear interpolation and gap filling algorithm

Input variables	Role	Default value
tocr	Original L2A surface reflectance values	-
mask	Validity mask for each acquisition date	-
input_dates	Dates of each image acquisition	-
sampling_period	Temporal sampling rate	5
t_0	Starting sampling date	-
t_end	Last date	-
radius	Radius of the temporal window	15
Output variables	Role	Default value
rtocr	Resampled L2A surface reflectance values	-

Algorithm 2-1. Linear interpolation and gap filling

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 <div>+ AGRICULTURE</div>
	Issue	Page	Date	
	1.rev.2	13	17/02/2016	

Data: *rtocr, mask, input_dates, sampling_period, t.0, t.end*

Result: *rtocr*

```

begin
  for pixel ∈ rtocr, mpixel ∈ mask do
    odc ← 0;
    od ← t.0 + sampling_period × odc;
    outpix ← 0;
    while od ≤ t.end do
      pvd ← find_previous_valid_date(od, mask, input_dates);
      nvd ← find_next_valid_date(od, mask, input_dates);
      pweight ← od - pvd;
      nweight ← nvd - od;
      outpix[od] ←  $\frac{pixel[pvd] \times pweight + pixel[nvd] \times nweight}{pweight + nweight}$ ;
      odc ← odc + 1;
      od ← t.0 + sampling_period × odc;
    end
    rtocr[positon(pixel)] ← outpix;
  end
end

```

The same algorithm can be implemented on-line, since the research of valid dates is performed inside a temporal window of size $2 \times \text{radius} + 1$. The function *find_previous_valid_date* (resp. *find_next_valid_date*) searches backward (resp. forward) the first date for which a valid value is available.

The minimum time series length required for the gap filling computation is 2 dates: one from the past and one from the future to fill the gap. It does not exist any condition about the minimum distance between the dates.

2.2 Pre-processing: spectral indices computation

The pre-processing workflow is illustrated in Figure 2-3, resulting in the three spectral indices: NDVI, NDWI and Brightness.

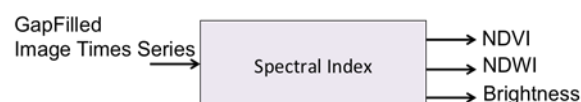




Figure 2-3. Pre-processing workflow, computing NDVI, NDWI and Brightness indices

The spectral indices computation step produces the relevant features for the classification. The features are computed for each date of the resampled and gap-filled time series and concatenated together into a single multi-channel image file.

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	14	17/02/2016	

Algorithm 2-2 describes the procedure, which can easily be implemented using the ORFEO Toolbox BandMath application. Variables of the algorithm are given in Table 2-2. The NDVI computation will use the B8 band (not the B8a). The SWIR band will be resampled at 10 m.

Table 2-2. Variables for the spectral indices computation

Input variables	Role	Default value
rtocr	Resampled L2A surface reflectance values	-
Output variables	Role	Default value
fts	Full gap-filled spectral indices times series	-

Algorithm 2-2. Spectral indices computation

```

Data: rtocr
Result: fts
begin
  for pixel  $\in$  rtocr do
    ndvi  $\leftarrow \frac{\text{pixel}[NIR] - \text{pixel}[R]}{\text{pixel}[NIR] + \text{pixel}[R]}$ ;
    ndwi  $\leftarrow \frac{\text{pixel}[SWIR] - \text{pixel}[NIR]}{\text{pixel}[SWIR] + \text{pixel}[NIR]}$ ;
    bright  $\leftarrow \sqrt{\text{pixel}[G]^2 + \text{pixel}[R]^2 + \text{pixel}[NIR]^2 + \text{pixel}[SWIR]^2}$ ;
    concatenate(pixel, ndvi, ndwi, bright) end
    fts[positon(pixel)]  $\leftarrow$  outpix;
  end

```

2.3 Feature extraction step

Three different indices have been generated: the NDVI, the NDWI and the Brightness. These indices are used in order to compute different features: temporal features (see sub-section 2.3.1) and statistic features (see sub-section 2.3.2).

17 temporal features are derived from the NDVI. Concerning the other two spectral indices, 5 statistics features are computed for each of them. In total, 27 features are computed, which are concatenated as a vector image (see sub-section 2.3.3) to be used in the subsequent classification step (that will be detailed in section 2.4).

2.3.1 Temporal features

The workflow developed to generate temporal features is illustrated in Figure 2-4.



Figure 2-4. Temporal features generation workflow

The features are computed using a C++ code compatible with OTB has been developed (see the section *NDVI feature computation task* presented in the Appendix of Section 5). Input and output variables are given in Table 2-3. The different features are described below.



	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	15	17/02/2016	

Table 2-3. Variables for the spectral indices computation

Input variables	Role	Default value
fts - NDVI	The full gap-filled NDVI times series	-
Output variables	Role	Default value
temporal_feat	A vector image made of 17 channels, one for each temporal feature	-

- **Global features**

These features characterize the global trends on NDVI times series. The three features detailed Table 2-4 are computed.

Table 2-4. Global features description

Notation	Description	Equation
x_{\max}	Maximum NDVI value	$\max_i x(t_i) = \{ x(t_i) \mid \forall j : x(t_j) \leq x(t_i) \}$
x_{mean}	Mean NDVI value	$\frac{1}{T} \sum_{j=1}^T x(t_j)$
x_{std}	Standard deviation NDVI value	$\sqrt{\frac{1}{T} \sum_{j=1}^T (x(t_j) - x_{\text{mean}})^2}$

Parameters:



- $x(t_i)$ corresponds to the NDVI value at instant t_i .
- T is the length of the NDVI times series
- mean is the mean NDVI value computed by using the T values

- **Features detecting largest local NDVI transitions**

These features describe the difference between consecutive windowed NDVI values. They are presented in Table 2-5. The goal is to detect the largest transitions corresponding to the greenness and the senescence onset periods.

Table 2-5. Description of features detecting largest local NDVI transitions

Notation	Description	Equation
$x_{\text{Dif-max}}$	Maximum NDVI difference found in a slicing temporal neighborhood having a size w (default value $w=2$).	$\max_i \frac{1}{w} \sum_{j=i}^{i+w-1} x(t_j) - \frac{1}{w} \sum_{j=i+w}^{i+2w-1} x(t_j)$
$x_{\text{Dif-min}}$	Minimum NDVI difference found in a slicing temporal neighborhood having a size w (default value $w=2$).	$\min_i \frac{1}{w} \sum_{j=i}^{i+w-1} x(t_j) - \frac{1}{w} \sum_{j=i+w}^{i+2w-1} x(t_j)$
$x_{\text{Dif-Dif}}$	Difference between $x_{\text{Dif-max}}$ and $x_{\text{Dif-min}}$ value estimating the transition jump	$x_{\text{Dif-max}} - x_{\text{Dif-min}}$

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	16	17/02/2016	

Parameters:

- $x(t_i)$ and $x(t_j)$ correspond to the NDVI values at instants t_i and t_j .
- w is the size of the slicing temporal window used to compute the differences

- **mbiFeatures associated to the maximum NDVI value:**

These features are associated to the area containing the maximum NDVI value, as identified in Figure 2-5. The features are described in Table 2-6.

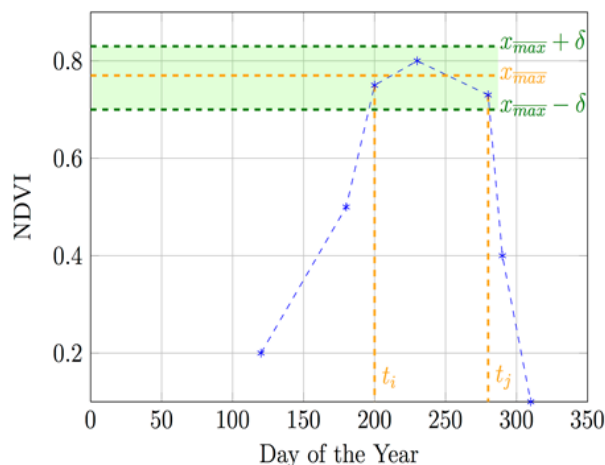




Figure 2-5. Maximum NDVI values identification from a full NDVI time series

Table 2-6. Maximum NDVI features description

Notation	Description	Equation
$x_{\overline{max}}$	Maximum mean NDVI value found in a slicing temporal neighborhood having a size w (default value $w=2$).	$\max_i \frac{1}{w} \sum_{j=i}^{i+w-1} x(t_j)$
$x_{\overline{max}-Lg}$	It estimates the length of the flat zone containing the peak area associated to $x_{\overline{max}}$ value.	$\max_{j,i} t_j - t_i = \{ \forall k \in [i, j] \mid x_{\overline{max}} - \delta \geq x(t_k) \leq x_{\overline{max}} + \delta \}$
$x_{\overline{max}-Sur}$	It estimates the surface of the flat zone area containing the peak value associated to $x_{\overline{max}}$	$x_{\overline{max}-Lg} \cdot x_{\overline{max}}$

Parameters:

- $x(t_i)$ and $x(t_j)$ correspond to the NDVI values at instants t_i and t_j .
- w is the size of the slicing temporal window used to compute the differences
- delta δ parameter is a user threshold, whose default value is equal to 0.05. In order to estimate the surface, a rectangular shape has been considered.

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	17	17/02/2016	

- **Features associated to the greenness onset**

These features are associated to the largest increasing period of the NDVI function. As it can be seen in Figure 2-6, the greenness onset period is highlighted in orange. These temporal features are subject to the next constraint :

$$\{\forall k \in [i, j] \mid x'(t_k) \geq 0\}$$

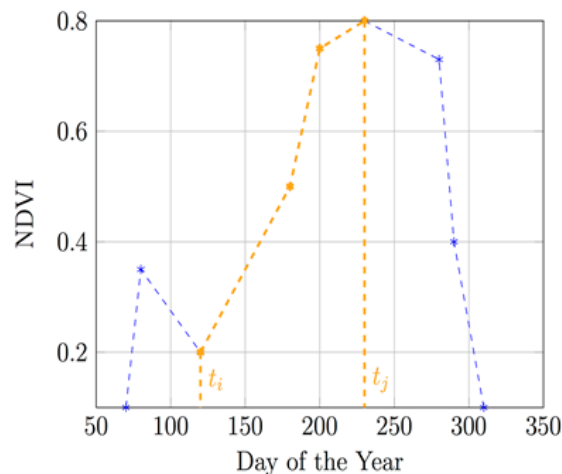


Figure 2-6. Identification of the largest increasing period in a full NDVI time series

Table 2-7 describes the features associated to the greenness period. In order to estimate the surface, a triangular shape has been considered.

Table 2-7. Greenness period features description

Notation	Description	Equation
XPos-Sur	It estimates the maximum surface of the first positive derivative period	$\max_{j,i} \frac{(x(t_j) - x(t_i))(t_j - t_i)}{2}$
XPos-Lg	It estimates the length of the first positive derivative period associated to XPos-Sur value.	$t_j - t_i$ subject to $\operatorname{argmax}_{t_i, t_j} \frac{(x(t_j) - x(t_i))(t_j - t_i)}{2}$
XPos-Rt	It estimates the rate of the first positive derivative period associated to XPos-Sur	$\frac{x(t_j) - x(t_i)}{t_j - t_i}$ subject to $\operatorname{argmax}_{t_i, t_j} \frac{(x(t_j) - x(t_i))(t_j - t_i)}{2}$

Parameters:

- $x(t_i)$ and $x(t_j)$ correspond to the NDVI values at instants t_i and t_j .

- **Features associated to the senescence onset:**

These features are associated to the largest decreasing period of the NDVI function. They are presented in Table 2-8.



	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		
	Issue	Page	Date	
	1.rev.2	18	17/02/2016	

Table 2-8. Senescence onset features description

Notation	Description	Equation
$x_{Neg-Sur}$	It estimates the maximum surface of the first negative derivative period	$\max_{j,i} \frac{(x(t_i) - x(t_j))(t_j - t_i)}{2}$
x_{Neg-Lg}	It estimates the length of the first negative derivative period associated to $x_{Neg-Sur}$ value.	$t_j - t_i$ subject to $\arg\max_{t_i, t_j} \frac{(x(t_i) - x(t_j))(t_j - t_i)}{2}$
x_{Neg-Rt}	It estimates the rate of senescence of the first negative derivative period associated to $x_{Neg-Sur}$	$\frac{x(t_i) - x(t_j)}{t_j - t_i}$ subject to $\arg\max_{t_i, t_j} \frac{(x(t_i) - x(t_j))(t_j - t_i)}{2}$

Parameters:

- $x(t_i)$ and $x(t_j)$ correspond to the NDVI values at instants t_i and t_j .

• Features characterizing bare soil transitions

These features detect if there is a transition between the bare soil before the greenness onset or after the senescence onset. Bare soil transition is illustrated in Figure 2-7 and corresponding features are given in Table 2-9. The resulting value is 0 or 1.

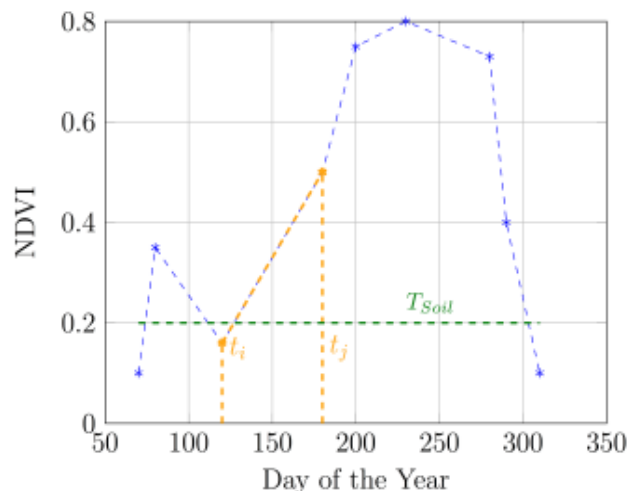




Figure 2-7. Identification of bare soil transitions in a full NDVI time series

Table 2-9. Bare soil transition features description

Notation	Equation
x_{Pos-Tr}	$\begin{cases} 1 & \text{if } x(t_i) \leq T_{soil} \text{ and } x(t_{i+1}) \geq T_{soil} \\ 0 & \text{otherwise} \end{cases}$
x_{Neg-Tr}	$\begin{cases} 1 & \text{if } x(t_i) \geq T_{soil} \text{ and } x(t_{i+1}) \leq T_{soil} \\ 0 & \text{otherwise} \end{cases}$

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	19	17/02/2016	

Parameters:

- $x(t_i)$ corresponds to the NDVI values at instant t_i
- T_{soil} is a threshold defining non cultivated areas

2.3.2 Statistic features

The workflow developed to generate statistic features is illustrated in Figure 2-8.

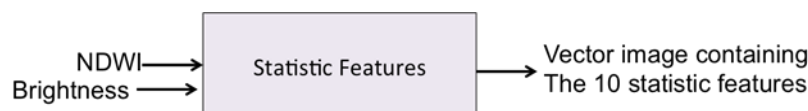


Figure 2-8. Statistic features generation workflow

The features are computed using a C++ code compatible with OTB has been developed (see the section *Statistics feature computation task* from the Appendix presented in Section 5). Input and output variables are given in Table 2-10. The statistic features are described in Table 2-11.

Table 2-10. Variables for the spectral indices computation



Input variables	Role	Default value
fts - NDWI	The full gap-filled NDWI times series	-
fts - Brightness	The full gap-filled Brightness times series	-
Output variables	Role	Default value
statistic_feat	A vector image made of 10 channels: 5 statistic features computed on Brightness and 5 statistic features computed on NDWI	-

Table 2-11. Statistic features description

Notation	Description
x_{max}	Maximum value of the complet index times series
x_{min}	Minimum value of the complet index times series
x_{mean}	Mean value of the complet index times series
x_{median}	Median value of the complet index times series
x_{std}	Standard deviation value of the complet index times series

2.3.3 Feature image concatenation

The complete set of temporal and statistic features are then concatenated as shown in Figure 2-9. The resulting feature vector image is the output of the “feature extraction” step and will be used as a input for the next step : the classification.

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	20	17/02/2016	

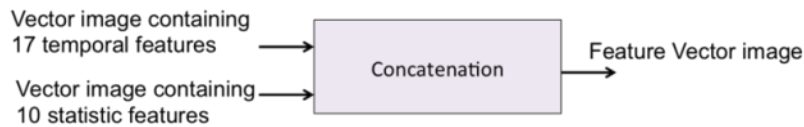


Figure 2-9. Workflow of the temporal and statistic features concatenation

Concatenation can be done using the OTB application presented in Algorithm 2-3. Input and output variables are given in Table 2-12.

Table 2-12. Variables for the spectral features concatenation

Input variables	Role	Default value
temporal_feat	A vector image made of 17 channels, one for each temporal feature	-
statistic_feat	A vector image made of 10 channels: 5 statistic features computed on Brightness and 5 statistic features computed on NDWI	-
Output variables	Role	Default value
Feature Vector Image	A vector image containing the 27 temporal and statistic features	-

Algorithm 2-3. OTB application performing the feature images concatenation

```
otbcli_ConcatenateImages -il Temporal_Features.tif Statistic_Features.tif
-out Feature_Concatenate_Images.tif
```



2.4 Classification step

The classification step can be divided into two subtasks:

1. The construction of the classification model ;
2. The use of the classification model in the complete image times series and its evaluation.

2.4.1 Constructing the classification model

The workflow underlying the construction of the classification model is illustrated in Figure 2-10. It is made of 3 main tasks: splitting polygons for training and testing, randomly selecting training pixel samples and random forest training. They are detailed in the above sub-sections.

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	21	17/02/2016	

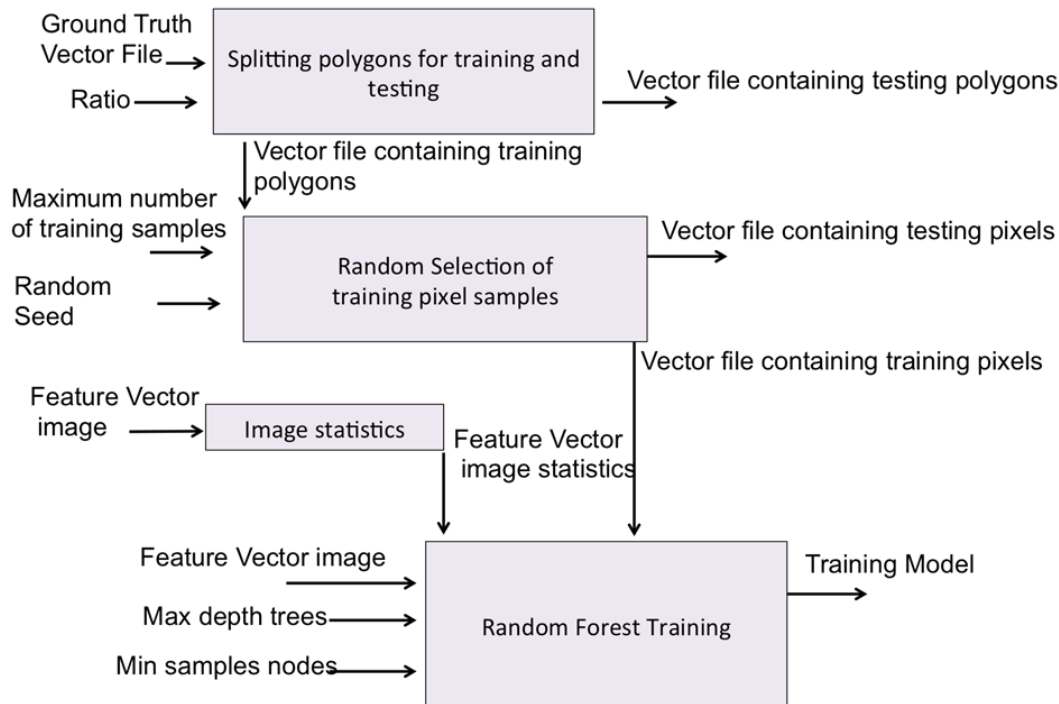


Figure 2-10. Workflow of the classification model construction

2.4.1.1 Splitting polygons for training and testing

This first task consists in splitting the in-situ dataset into 2 disjoint subsets: the training set and the validation set (Figure 2-11).

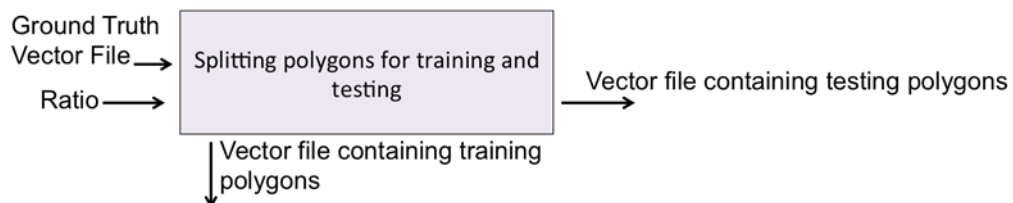




Figure 2-11. Workflow of the in-situ dataset splitting sub-task

These sets are composed of polygons, not individual pixels. The algorithm is therefore a random sampling without replacement of the polygons of each class with probability $p = \text{ratio}$ value for the training set and $1 - p$ for the validation set. An illustration is the whole process is provided in Figure 2-12, with a ratio of 0.5 between the 2 subsets.

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	22	17/02/2016	

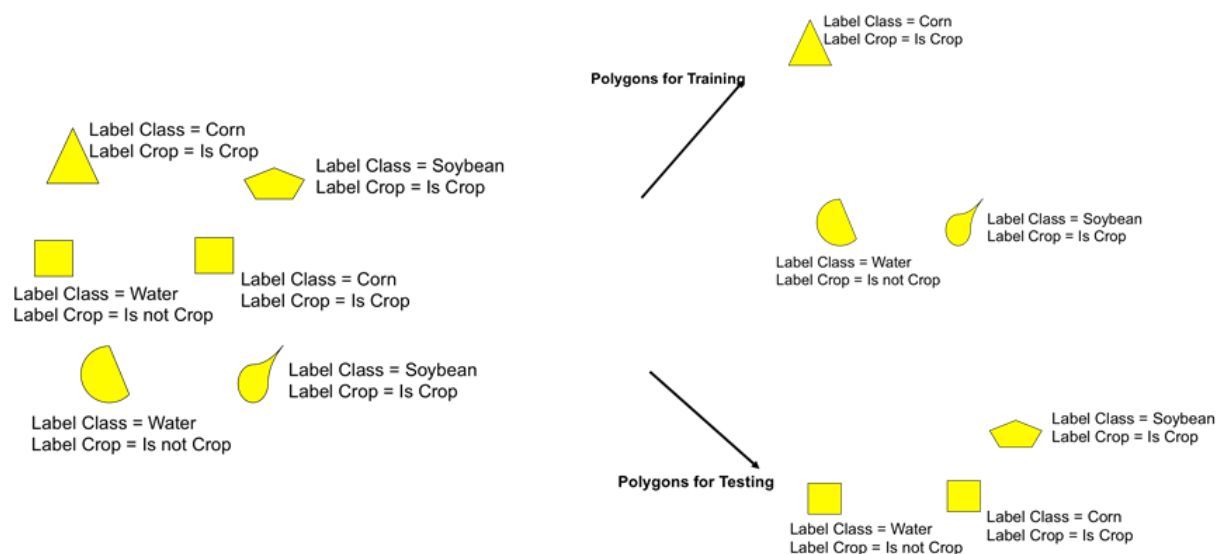




Figure 2-12. Illustration of the in-situ dataset split into two subsets for training and validation using a ratio equal to 0.5

Algorithm 2-4 describes the procedure. Input and output variables of the algorithm are given in Table 2-13. The output sets are made of polygons and not individual pixels.

Table 2-13. Variables for the splitting the field dataset for training and validation

Input variables	Role	Default value
Reference_polygons	Vector file containing in-situ data	-
Sample_ratio	Parameter expressing the ratio between the number of training and validation polygons per class (corn, soybean, rice, no-crop, etc.)	0.75
Output variables	Role	Default value
Training_polygons	Vector file containing in-situ data for training	-
Validation_polygons	Vector file containing in-situ data for validation	-

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	23	17/02/2016	

Algorithm 2-4. OTB application performing the feature images concatenation

Data: *reference_polygons, sample_ratio, list_of_class_labels*

Result: *training_polygons, validation_polygons*

```

begin
  training_polygons ← ∅;
  validation_polygons ← ∅;
  for cl ∈ list_of_class_labels do
    for poly ∈ reference_polygons do
      if class_of(poly) = cl then
        p ← random(0, 1);
        if p ≤ sample_ratio then
          add poly to training_polygons;
        end
      else
        add poly to validation_polygons;
      end
      remove poly from reference_polygons;
    end
  end
end
end

```

2.4.1.2 Random selection of training pixel samples : Crops and Not Crops

This step consists in randomly selecting a number of samples (i.e. pixels) for each class : “crop” and “no-crop”. The workflow is illustrated in Figure 2-13. The output selection will be class consistent, meaning that if 33% of crops belong to soybean class, 33% of the crop class training samples will also belong to the soybean class.

This reduction of the training sample size is needed to reduce the time complexity of the RF classifier. If it is possible from the operational point of view, this step could be removed and the set of the training pixels samples will correspond to the set of the training polygon samples.

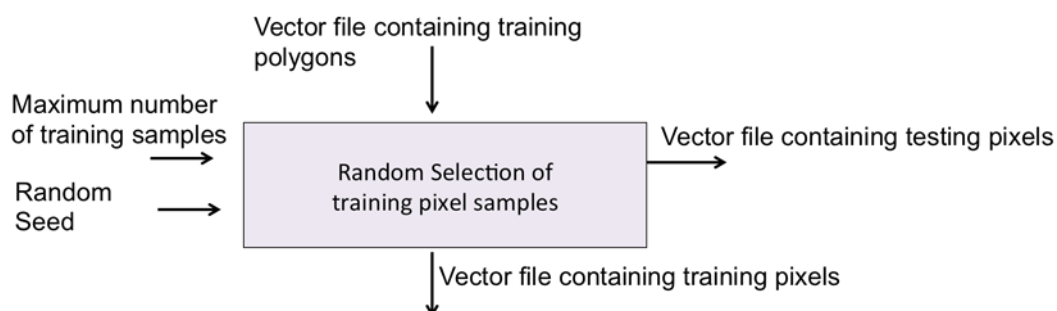




Figure 2-13. Workflow for reducing the size of the training sample



This random selection is done using a C++ code compatible with OTB, which has been developed (Algorithm 2-5). This code is inspired from the OTB class “*otbListSampleGenerator*”.

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	24	17/02/2016	

Input and output variables are given in Table 2-14. As for the input vector file containing training polygons, it must be remarked that the polygons of this vector file as associated with two kinds of label: class label and crop label. For instance, a polygon will be composed by the class label “corn” and the crop label “Is a crop”.

Table 2-14. Variables for the splitting the field dataset for training and validation

Input variables	Role	Default value
reference_polygons	Vector file containing in-situ data for training obtained by the previous splitting algorithm	-
Nb_tr_samples	Maximum number of training samples: this is the limited number of training samples that will be used for training. For instance, if this number is equal to 1000, it means that the output data will be made of 1000 crop and 1000 no crop samples.	1000/5000
random_seed	Random seed that will be used to do the random selection of the pixels. If the third parameter is not specified, a random seed will be generated automatically	0
Output variables	Role	Default value
training_vector_file	Vector file containing the limited in-situ data for training. Polygons will be associated with one label: crop (with value 1) or no-crop (with value 0)	-
testing_vector_file	vector file containing the data from Reference_polygons which are not included in the training_vector_file. Polygons will be associated with one label: crop (with value 1) or no-crop (with value 0)	-

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	25	17/02/2016	

Algorithm 2-5. Random selection of training samples

Data: *reference_polygons*, *Nb_tr_samples*, *random_seed*
Result: *training_vector_file*, *testing_vector_file*

```

begin
  for poly ∈ reference_polygons do
    Nb_samples(Label_class_of(poly)) ← Nb_samples(Label_class_of(poly)) + area_polygon(poly);
    if Label_crop_of(poly) = crop then
      | total_nb_crop_samples ← total_nb_crop_samples + area_polygon(poly);
    else
      | total_nb_no_crop_samples ← total_nb_no_crop_samples + area_polygon(poly);
    end
  end
  end
  Nb_tr_samples ← minimum(Nb_tr_samples, total_nb_crop_samples, total_nb_no_crop_samples)
  for cl ∈ Nb_samples do
    if cl = crop then
      | Nb_tr_samples_to_be_selected(cl) ← Nb_samples(cl) * Nb_tr_samples / total_nb_crop_samples
    else
      | Nb_tr_samples_to_be_selected(cl) ← Nb_samples(cl) * Nb_tr_samples / total_nb_no_crop_samples
    end
    training_vector_file ← Select randomly Nb_tr_samples_to_be_selected(cl) samples according to
    random_seed
    testing_vector_file ← Include all the samples that are not included in training_vector_file file
  end
end

```

2.4.1.3 Random Forest Training step

The training of the classifier step is illustrated in Figure 2-14.

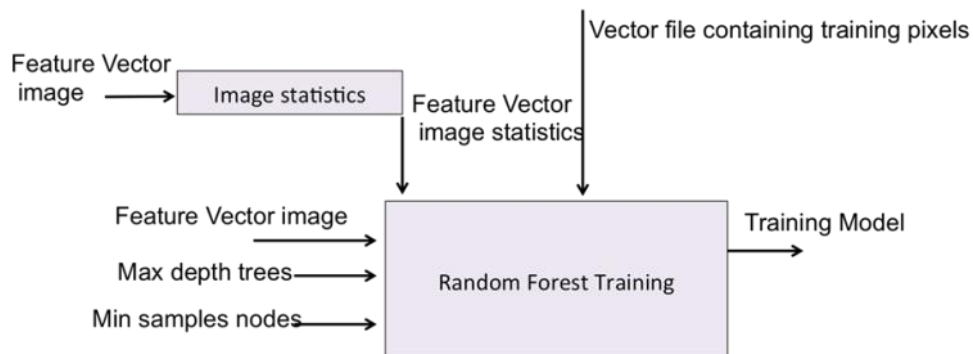


Figure 2-14. Workflow of the classifier training

This task has been developed on C++ (see the section *Random Forest Training Step* task from the Appendix presented in Section 5) and it is based on the “*otbcli_TrainImagesClassifier*” application. The OTB application is composed of two steps:

1. Random selection of training and testing samples;
2. Training of the classifier by using the previous training samples.

In our case, the splitting task is done by using the algorithm presented in the previous section (2.4.1.2). For this reason, the developed code implemented in C++ and compatible with OTB Library focuses on the second step.

Input and output variables are presented in Table 2-15.



	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		
	Issue	Page	Date	
	1.rev.2	26	17/02/2016	

Table 2-15. Variables for training the Radom Forest

Input variables	Role	Default value
Feature Vector Image	vector image containing the 27 temporal and statistic features	-
Feature Vector Image Statistics	the mean and the variance of each feature composing the feature vector image. This result can be obtained by applying the OTB application presented in 2.3.1 of the ATBD crop type document	-
Max depth tree	maximum depth of the trees used for Random Forest classifier	25
Min samples nodes	minimum number of samples in each node used by the classifier	5
training_vector_file	vector file containing the training pixels	-
Output variables	Role	Default value
Training model	Resulting RF model to be used in the classification step	-

Concerning the feature vector image statistics, the goal here is to compute the mean and the standard deviation of each input feature of the classifier so that the samples can be normalized inside the training and classification steps.



In order to compute it, the Orfeo Toolbox ComputeImagesStatistics application is used in order to produce an XML file containing the statistics for each channel of the image of features (Algorithm 2-6).

Algorithm 2-6. OTB application computing the vector image statistics

```
otbcli_ComputeImagesStatistics -il feature-time-series.tif -out statistics.xml
```

2.4.2 Classification of the image times series : Random Forest

This step consists in the classification of all the pixels of the image times series made of all the statistic and temporal features (see section 2.3). The general workflow is illustrated in Figure 2-15.

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	27	17/02/2016	

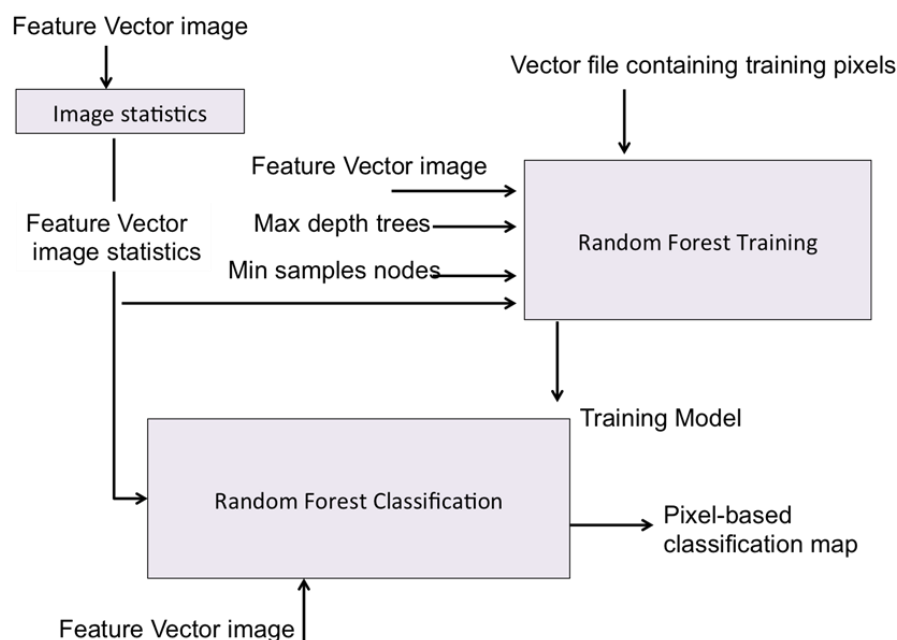


Figure 2-15. Workflow of the RF classification

A C++ code compatible with OTB has been developed for this application. The sources can be found in the files presented in the section *Classification of the complete image times series task* of the Appendix presented at Section 5) and it is based on the “*otbcli_ImageClassifier*” application.



Input and output variables are given in Table 2-16.

Table 2-16. Variables for the RF classification algorithm

Input variables	Role	Default value
Feature Vector Image	Vector image containing the 27 temporal and statistic features	-
Feature Vector Image Statistics	Mean and variance of each feature composing the feature vector image	-
Training model	Resulting RF model to be used in the classification step	-
Output variables	Role	Default value
Pixel-based classification map	Resulting crop mask product	-

2.5 Processing chain working in the iterative operational model

The processing chain detailed in the section 2.4 (i.e. the classification step) will have to run in an operational mode in which the cropland mask will be obtained iteratively. Two different

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	28	17/02/2016	

cases must be taking into account regarding the satellite acquisitions: the first year of Sentinel-2 images and the other years. These two situations are analyzed here after.

2.5.1 The first year of Sentinel-2 images

The “in-situ processing chain” works if it exists ground truth data from the current year. Satellite data captured from the 6 first months will be necessary to produce the first cropland mask product. Considering that the first cropland map will be constructed at instant T, Figure 2-16 shows how the cropland product will be computed in an iterative way for the following instants of time.

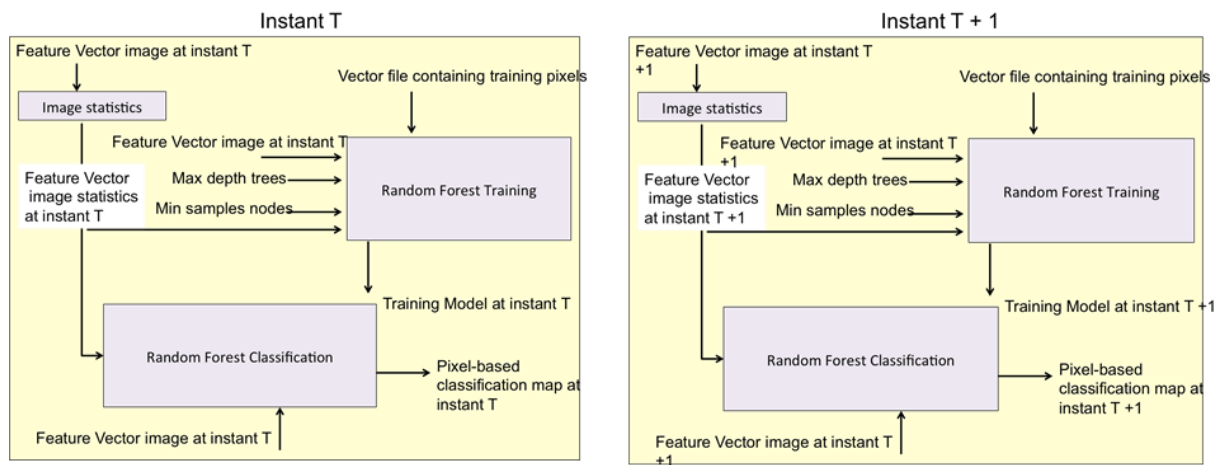


Figure 2-16. “In-situ” classification chain running at instants T and T+1, when we are during the first S2 year



The right figure shows how since the instant T, the feature image will be recomputed at each instant of time. Besides, this image will be used: (i) to compute the image statistics, (ii) to train the classifier and (iii) to classify the complete image.

2.5.2 The rest of the years

For the rest of the years, some considerations have been done:

1. A complete year of data acquisitions exists from the previous year;
2. The ground truth in-situ data exists from the previous year;
3. The satellite data of the current year have been interpolated, in order to have the same acquisition dates than during the previous year.

According to these last assumptions, a classifier model is computed for the previous year. This allows developing the system presented in Figure 2-17 and Figure 2-18. In these figures, Year X-1 corresponds to the previous year, whereas Year X corresponds to the current year.

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	29	17/02/2016	

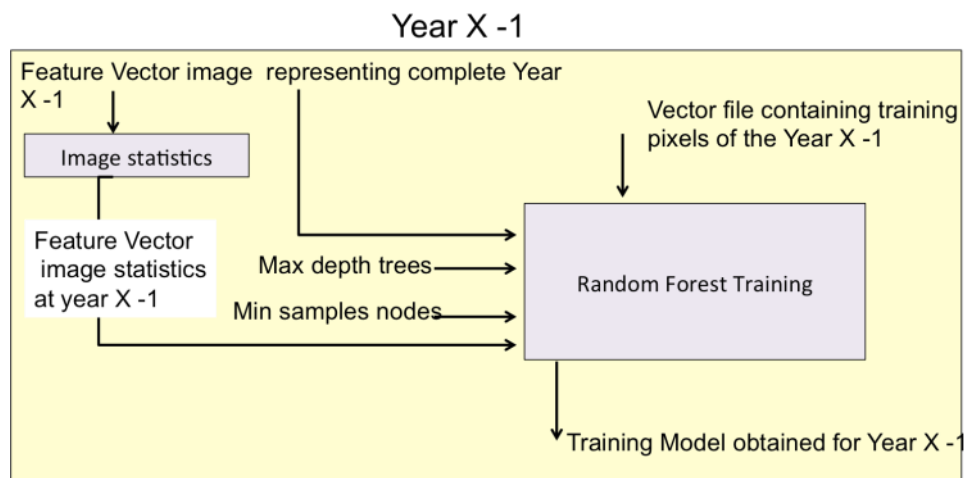


Figure 2-17. “In-situ” classification chain running in Year X-1, when we are not during the first S2 year

By using the classifier model obtained for the previous year (i.e. built on the in-situ data and the images from the previous years), the classification of the current year will be done, after each new acquisition captured at instant T, as illustrated in Figure 2-18.

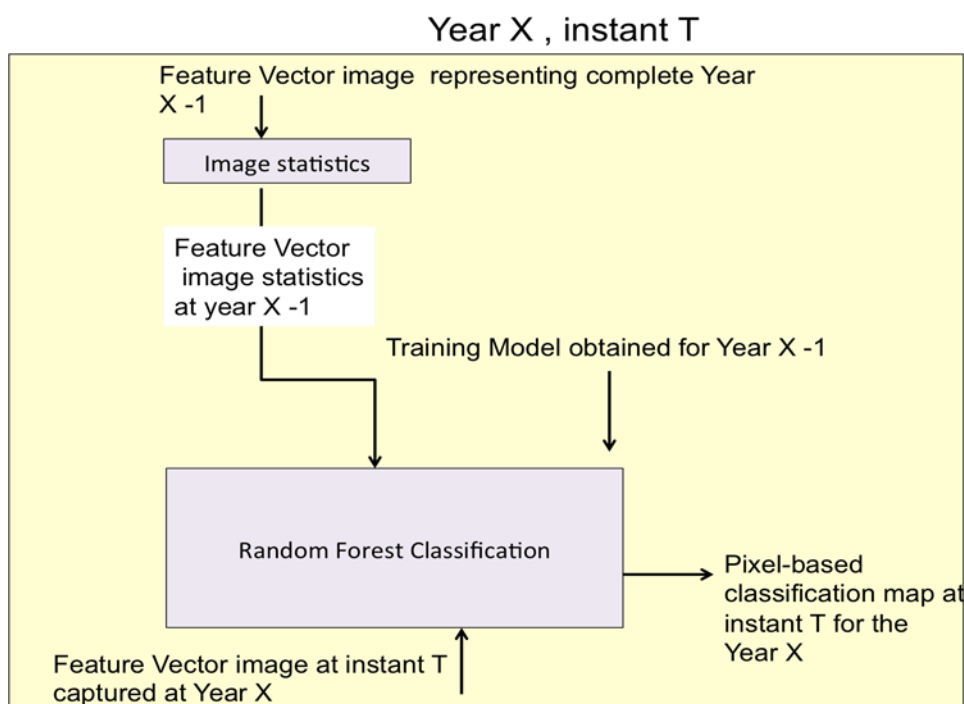




Figure 2-18. “In-situ” classification chain running in Year X, when we are not during the first S2 year

2.6 Processing chain model working for a large area covering several satellite footprints

When working large areas, as shown in Figure 2-19, the construction of the crop mask will need an additional input variable, which has been namely *Reference map*. This map

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	30	17/02/2016	

corresponds to an image covering the satellite footprint. In it, the values of the pixels correspond to the classifier model that must be used to classify the satellite footprint.

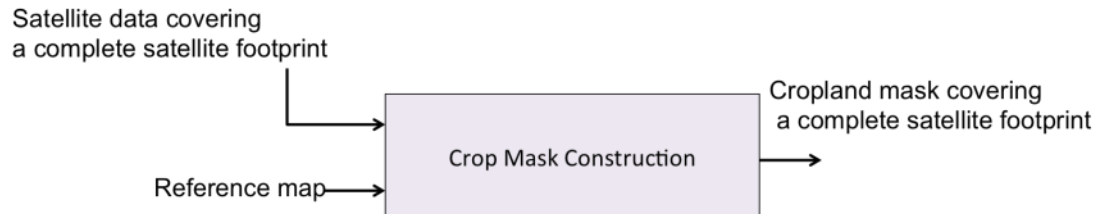




Figure 2-19. Crop mask production logical workflow when working over large areas

For instance, if all pixels of the reference map have a value equal to 1, it means that the classifier model number 1 will be used in order to classify the complete footprint. The classifier model number 1 can be for example a classifier model obtained from a neighborhood footprint. When the reference map has pixels associated with different values, it means that different models will be used for classifying the footprint.

It can be noted that this approach can also contribute to solve the problem where in-situ data is only available in one satellite footprint.

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	31	17/02/2016	

3 Mode without in-situ data

3.1 Preliminary step: data smoothing and gap filling

A Whittaker filter is applied to the time series, both on the reflectance bands and on the NDVI (Figure 3-1). The filter is applied using the R function “*Whit1*”. *Whit1* is a function from the “ptw” R package, which is a weighted Whittaker smoothing with a first order finite difference penalty. Additional info can be found at <http://cran.r-project.org/web/packages/ptw/ptw.pdf>.

The goal of this Whittaker filter is (i) to smooth the reflectance time series taking into account the non-constant temporal step. To do so, the temporal gaps in the original time series are filled by empty images in order to get one image for each date of the period. Every pixels of the empty images have a weight of 0 in the weight vector of the Whittaker function .

And (ii) to fill the gap in the reflectance time series. To do so, the missing data in the original acquisitions are identified thanks to validity masks which distinguish the valid pixels (value=1) from non-valid pixels (value=0) based on each image cloud mask. These validity masks feed the Whittaker and correspond to the weights of the acquisitions.

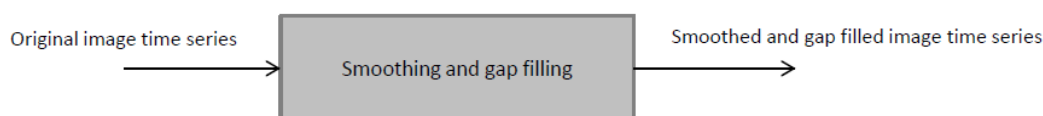




Figure 3-1. Workflow of the smoothing preliminary step

The Whittaker filter is presented in detail in Algorithm 4-1, while all input and output variables are listed in Table 3-1. Inputs are the original image time series ; they are presented here separately for the reflectance channels (band<n>) and for the NDVI (NDVI). In the outputs, each raster file will have a number of bands equal to the number of image acquisition dates.

The smoothing is applied separately for each time series (4 spectral channels with SPOT 4 Take 5 dataset and NDVI). The time series is given as a vector.

Table 3-1. Variables for the data smoothing

Input variables	Role	Default value
band<n>	original time series (reflectance values) separated per bands.	-
NDVI	original NDVI time series.	-
Lambda	Smoothing parameter of the Whitaker function (larger values lead to more smoothing)	2
Validity_mask	Derived from the cloud mask of each acquisition. The validity mask distinguish the valid pixels (value=1) from non-valid pixels (value=0). Only the pixel classified as <i>land</i>	

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		
	Issue	Page	Date	
	1.rev.2	32	17/02/2016	

	are considered as <i>valid</i> . One band per input image.	
weight	Weights array of length corresponding to the input interval in days. Default weights correspond to one signal to be smoothed. If weight is equal to 0, the pixel is gap-filled but its original value is not taken into account in the smoothing. For each acquisition, the weights of the pixels are determined by its “validity masks”. In order to take into account the non-constant temporal step between acquisition, when we don’t have any acquisition for one date in the period, the weight is equal to 0.	1
Output variables	Role	Default value
band<n>_smoothed	Raster files including the smoothed and gap-filled reflectance values for all the dates	-
NDVI_smoothed	Raster files including the smoothed and gap-filled NDVI values for all the dates	-

Algorithm 3-1. Smoothing based on the Whitaker function

Algorithm “smoothing/gap-filling” is

Temporary output:

n raster files, n corresponding to the different spectral bands

band_<n>

Creating the validity masks of each acquisition

For each of the acquisition:

select the cloud masks of the original image time series corresponding

Assign a value of 1 to the *land* pixels and a value of 0 to the other categories

Store and save these layers in *validity_mask* (one band per input image).

Creating the smoothed and gap-filled temporal series of the different spectral bands####

dates <- length in days of input date interval

For each pixel of the *band<n>*:

weight <- new array of length *dates* filled with 0

for each pixel index, value in *validity_mask*:

if value is VALID:

d <- date of image *index*

weight[*d*] <- 1

for each band in the input raster:



pixel <- band reflectance values for the current pixel

vec <- array of length *dates* filled with 0

for each index, value in *pixel*:

d <- date of image *index*

vec[*d*] <- value of the pixel

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	33	17/02/2016	

Create the smooth reflectance values, “out_band”, by applying the R **whit1** function with the following parameter:

whit1(*band*<*n*>,*weight*,2)

Save the output “out_band” in *band*<*n*>_smoothed

Creating a smoothed temporal series of NDVI

Create the smooth NDVI values, “outW_ndvi”, by applying the R **whit1** function with the following parameter:

whit1(*NDVI_smoothed*, *weight*, 2)

Save the output “outW_ndvi” in *NDVI_smoothed*

3.2 Features extraction

This step is organized in two consecutive sub-tasks: the definition of temporal features and the spectral features extraction (Figure 3-2). They are detailed in the two sub-sections 3.2.1 and 3.2.2.

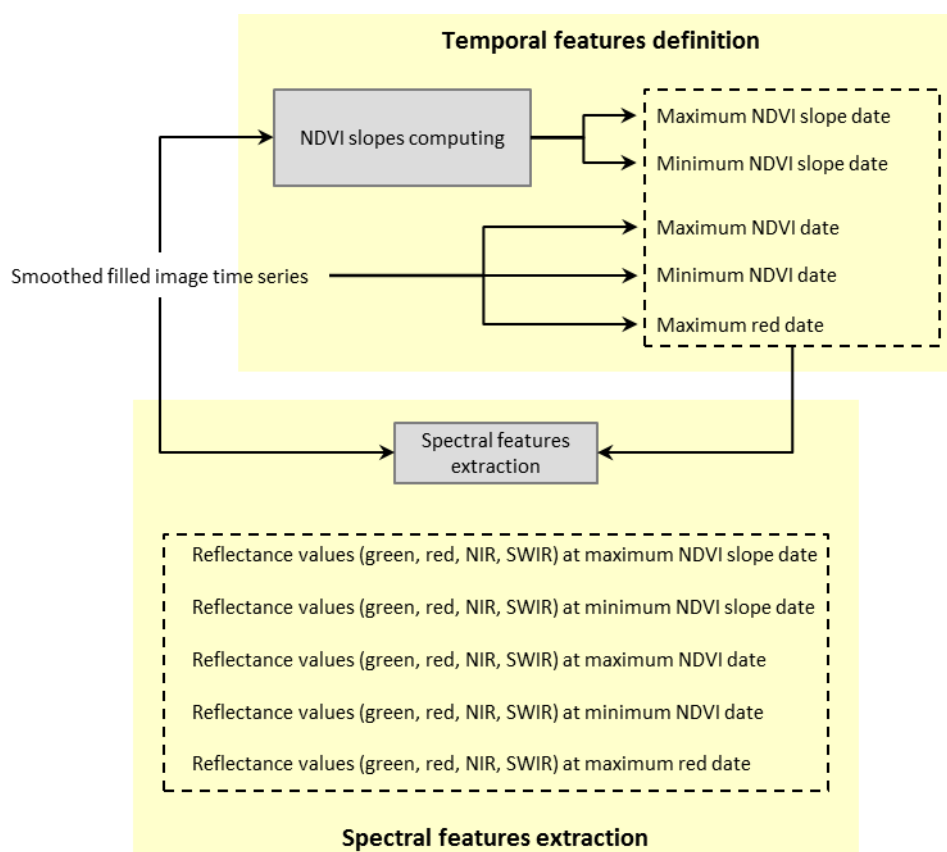




Figure 3-2. Workflow of the temporal and spectral features extraction process

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	34	17/02/2016	

3.2.1 Temporal features definition

This step identifies key dates during the growing cycle, based on the reflectance values and NDVI time series.

First, it computes the slopes of the “NDVI vs time” curve using the OTB function “otbcli_BandMath” (<https://www.orfeo-toolbox.org/CookBook/CookBooks132.html>). The algorithm calculates the slope of the NDVI vs time curve according to the following equation:



$$Slope_t = \frac{(NDVI_{t+1} + NDVI_{t-1})}{(t_{t+1} - t_{t-1})}$$

Then, it identifies on a per-pixel basis the date when the maximum and minimum slopes are achieved and the maximum and minimum NDVI are achieved. The algorithm also identifies when the maximum value of the red band is achieved. These processes are based on the R functions “which.max” and “which.min”. They are detailed in Algorithm 3-2. The algorithm runs by tile and stores the dates in 5 separate raster (one by feature).

Input and output of this step are presented in Table 3-2, as well as the parameter of the otbcli_BandMath function. The R functions “which.max”, “which.min” don’t have parameters.

Table 3-2. Variables for the temporal features identification

Input variables	Role	Default value
band<n>_smoothed	Raster files including the smoothed reflectance values for all the dates	-
NDVI_smoothed	Raster files including the smoothed NDVI values for all the dates	-
Image dates	Text file giving the acquisition dates of each image within the time series	-
Exp	Parameter of the otbcli_BandMath function, which is a mathematical expression	E.g. (im1(b3)-im1(b1))/diff _t)
Output variables	Role	Default value
date_max_slope	Raster file which stores, for each pixel, the date corresponding to the maximum slope of the “NDVI vs time” curve	-
date_min_slope	Raster file which stores, for each pixel, the date corresponding to the minimum slope of the “NDVI vs time” curve	-
date_max_NDVI	Raster file which stores, for each pixel, the date corresponding to the maximum NDVI value	-
date_min_NDVI	Raster file which stores, for each pixel, the date corresponding to the minimum NDVI value	-

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	35	17/02/2016	

date_max_red	Raster file which stores, for each pixel, the date corresponding to the maximum red value	-
--------------	---	---

Algorithm 3-2. Temporal features identification

Algorithm “temporal features identification” is

Temporary output:

d raster (Tif) files, d corresponding to the different dates of acquisition corresponding to the slope of the curve “NDVI vs time”.

NDVI_slope_<d>

Computing the slopes of NDVI####

Read the image dates in the *Image_dates* text file

For d in 3:(length(dates)) (*//The loop starts with the 3rd date, because the equation needs three followed dates to be computed*)

 Compute “diffdate” as : $\text{dates}[d] - \text{dates}[d-2]$

 Run the **otbcli_BandMath** function:

 system(**otbcli_BandMath** -il *NDVI_smoothed* -out *NDVI_slope_<d>* -exp
 '(im1b[d-2]-im1b[d])/diffdate')

End of For

Identification of the temporal features

Look for maximum and minimum NDVI slope####

List the files which contain dates of slopes (output of previous steps):

 inFileList <-list of files including all *NDVI_slope_<d>*

In the list, identify the maximum and minimum slopes using the **which.max** and **which.min** R functions:

 whmax = **which.max**(inFileList)

 whmin = **which.min**(inFileList)

save “whmax” as a raster in *date_max_slope*

save “whmin” as a raster in *date_min_slope*

Look for max/min ndvi value####

Identify the maximum and minimum NDVI values using the **which.max** and **which.min** R functions:

 whMm = **which.max**(*NDVI_smoothed*)

 whmm = **which.min**(*NDVI_smoothed*)

save “whMm” as a raster in *date_max_NDVI*



save “whmm” as a raster in *date_min_NDVI*

Look for max red value####

Identify the maximum Red value using the **which.max** R function:

 whmax = **which.max**(*band2_smoothed*)

save “whmax” as a raster in *date_max_red*

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		
	Issue	Page	Date	
	1.rev.2	36	17/02/2016	

Correction of dates to correspond with smoothed data (since we cannot compute the slope for the first & last dates) => offset=+1

Run the **otbcli_BandMath** function:

```
system( otbcli_BandMath -il date_max_slope -out date_max_slope -exp 'im1b1+1')
```

```
system( otbcli_BandMath -il date_min_slope -out date_min_slope -exp 'im1b1+1')
```



3.2.2 Spectral features extraction

This step consists in associating the key dates to their spectral values: for each pixel, the spectral information corresponding to the dates identified for the 5 key dates of the growing cycle are identified and stored in a new raster. This is achieved thanks to a C++ program developed in the project and presented in Algorithm 3-3.

Input and output data are listed in Table 3-3. In the outputs, each raster file contains n layers (n being the number of spectral band).

Table 3-3. Variables for the spectral features extraction

Input variables	Role	Default value
band<n>_smoothed	Raster files including the smoothed reflectance values for all the dates	-
date_max_slope	Raster file which stores, for each pixel, the date corresponding to the maximum slope of the “NDVI vs time” curve	-
date_min_slope	Raster file which stores, for each pixel, the date corresponding to the minimum slope of the “NDVI vs time” curve	-
date_max_NDVI	Raster file which stores, for each pixel, the date corresponding to the maximum NDVI value	-
date_min_NDVI	Raster file which stores, for each pixel, the date corresponding to the minimum NDVI value	-
date_max_red	Raster file which stores, for each pixel, the date corresponding to the maximum red value	-
inbands	Vector of the different raster files corresponding to each of the smoothed spectral bands	E.g. ('band1_smo.tif', 'band2_smo.tif', 'band3_smo.tif', 'band4_smo.tif')
Output variables	Role	Default value
ref_max_slope	Raster file which stores, for each pixel, the reflectance values corresponding to the maximum slope date	-
ref_min_slope	Raster file which stores, for each pixel, the	-

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		
	Issue	Page	Date	
	1.rev.2	37	17/02/2016	

	reflectance values corresponding to the minimum slope date	
ref_max_NDVI	Raster file which stores, for each pixel, the reflectance values corresponding to the maximum NDVI date	-
ref_min_NDVI	Raster file which stores, for each pixel, the reflectance values corresponding to the minimum NDVI date	-
ref_max_red	Raster file which stores, for each pixel, the reflectance values corresponding to the maximum red date	-

Algorithm 3-3. Spectral features extraction

Algorithm “spectral features extraction” is

Extraction of reflectance values in each spectral band for the 2 dates corresponding to the maximum and minimum slope

define the parameters:

```
infile<- date_max_slope
inbands<-(band1_smoothed, band2_smoothed, band3_smoothed, band4_smoothed)
outfile<- ref_max_slope
```

run the **selectInList** function:

```
system(selectInList outfile infile inbands)
```

define the parameters:

```
infile<- date_min_slope
inbands<-(band1_smoothed, band2_smoothed, band3_smoothed, band4_smoothed)
outfile<- ref_min_slope
```

run the **selectInList** function:

```
system(selectInList outfile infile inbands)
```

Extraction of reflectance values in each spectral band for the 2 dates corresponding to the maximum and minimum NDVI

define the parameters:

```
infile<- date_max_NDVI
inbands<-(band1_smoothed, band2_smoothed, band3_smoothed, band4_smoothed)
outfile<- ref_max_NDVI
```



run the **selectInList** function:

```
system(selectInList outfile infile inbands)
```

define the parameters:

```
infile<- date_min_NDVI
inbands<-(band1_smoothed, band2_smoothed, band3_smoothed, band4_smoothed)
outfile<- ref_min_NDVI
```

run the **selectInList** function:

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	38	17/02/2016	

system(**selectInList** outfile infile inbands)

Extraction reflectance values in each spectral band for the date corresponding to the maximum Red

define the parameters:

infile<- *date_max_red*

inbands<-(*band1_smoothed, band2_smoothed, band3_smoothed, band4_smoothed*)

outfile<- *ref_max_red*

run the **selectInList** function (see Algorithm 3-4):

system(**selectInList** outfile infile inbands)

Algorithm 3-4. « SelectInList » program

Program SelectInList is:

for each pixel : (*// loop over all the pixels*)

id = dateID (*// dateID is the file containing the key date to select (date_max_slope, date_min_slope, date_max_NDVI, date_min_NDVI, date_max_red)*)

for each input file i:

output(i) = pixelMB(i)(id) (*// pixelMB is a list of multiband files (band<n>_smoothed) with the first index for list item and the second for band*)

3.3 Reference map preparation

The reference is the map that will be used to define the training samples that will feed the coming classification algorithm. This is the key dataset that allows generating a cropland mask in the absence of field data.

This step consists in removing, from the reference map, the pixels that are at the border of the different classes (or said differently, that are between two classes) as they have a higher probability to be wrongly classified (Figure 3-3).

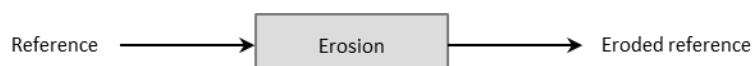


Figure 3-3. Workflow of the reference map preparation

The process is based on a function named **classErosion** (Algorithm 3-5), written in python during the benchmarking and that relies on the python **erosion** function from the “*skimage.morphology*” module.

Input and output variables are given in Table 3-4, as well as the parameters of the python **erosion** function.

Table 3-4. Variables for the erosion of the reference

Input variables	Role	Default value
reference	Raster file containing the original reference map, with the same extent and projection than the image time series	-



	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	39	17/02/2016	

Image	Image array corresponding to each class	classinRst
selem	Binary array made of 0 and 1 values	sq(3)
Output variables	Role	Default value
eroded_reference	Raster file containing the eroded reference map, with the same extent and projection than the image time series. Removed pixels are coded with NaN values	-

Algorithm 3-5. Reference erosion

Function classErosion **is**

```
def classErosion(Reference, Eroded_reference, size=3):
```

```
    ## Reading the input #####
```

```
    Reference is opened and stored in “inRst”
```

```
    ## Identifying the different classes included in the reference #####
```

```
    The different classes of “inRst” are identified and included in “uniqueClass” (where the classes are identified from 1 to u)
```

```
    ## Defining the size of the erosion #####
```

```
    selem = sq(size) (where size is “3” according to Table 3-4 => selem is squared matrix of 3x3 ‘ones’)
```

```
    ## Erosion of the different classes #####
```

```
    for u in “uniqueClass”:
```

```
        all the areas where “inRst” == u are saved in “classinRst”
```

```
        “eroded” = erosion(classinRst, selem)
```

```
        the resulting eroded areas are saved in “outRst”
```

```
    ## Saving the output file #####
```




```
    An array is written compiling the “outRst” coming from each class
```

```
    The array is saved in Eroded_reference
```

3.4 Trimming

Trimming consists in truncating a distribution from its least probable values that behave like outliers. The common purpose of this procedure is to reduce the sensitivity to outliers for many parameter estimates, such as the sample mean and variance.

Trimming, coupled with the previous reference map erosion, are two key steps that give robustness to the classification process. They will allow obtaining, from an inexact pre-existing reference map data that will be “clean” enough to train a classifier. They are therefore the core of the “without in-situ mode”.

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 
	Issue	Page	Date	
	1.rev.2	40	17/02/2016	

The spectral distribution corresponding to each class of the eroded reference is iteratively trimmed in order to eliminate the pixels that behave as “outliers” from their Gaussian distribution point of view.



The trimming will not be performed on all spectral channels, but on the ones that are the most characteristic for crops – no crops. The selected channels are the red and NIR reflectance values at the minimum NDVI date as well as the green, red and NIR reflectance values at the maximum NDVI date. This selection could be updated when S2 data will be available to better use the S2 spectral richness.

The trimming algorithm is described in Algorithm 3-6. It uses a “*MVG_IT*” function (Algorithm 3-7) specifically created in R that computes iteratively the outliers of a Gaussian distribution. “*MVG_IT*” uses, in turn, two functions from the base core of R, which are the “*qchisq*” and “*mahalanobis*” ones.

From the trimming output (i.e. the identification of clean pixels in the reference map), a selection of samples is done randomly. This selection will constitute the training dataset for the subsequent RF classifier.

It shall be noted that even if the trimming was done on a selection of spectral channels, the RF classifier will run with all available spectral information: red, NIR and SWIR reflectance values from the dates corresponding to the maximum red, minimum and maximum NDVI and minimum and maximum slope.

The workflow of the trimming process is presented in Figure 3-4.

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	41	17/02/2016	

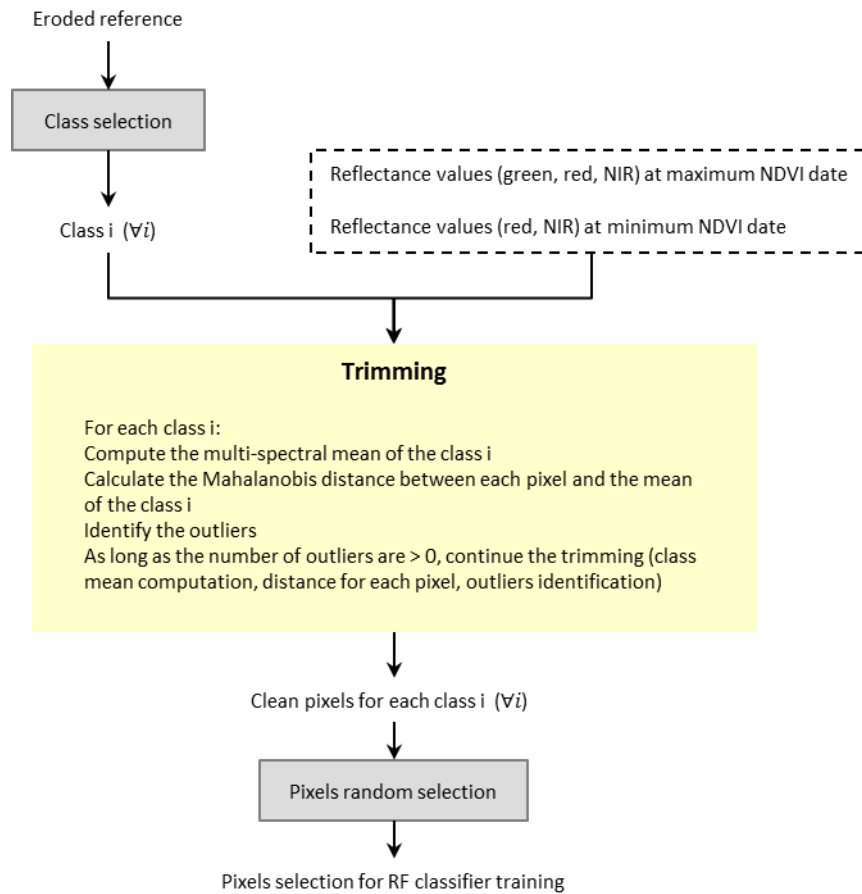




Figure 3-4. Workflow of the trimming process

Input and output variables of the trimming are presented in Table 3-5.

Table 3-5. Variables for the trimming

Input variables	Role	Default value
eroded_reference	Raster file containing the eroded reference map, with the same extent and projection than the image time series. Removed pixels are coded with NaN values	-
ref_max_slope	Raster file which stores, for each pixel, the reflectance values corresponding to the maximum slope date	-
ref_min_slope	Raster file which stores, for each pixel, the reflectance values corresponding to the minimum slope date	-
ref_max_NDVI	Raster file which stores, for each pixel, the reflectance values corresponding to the maximum NDVI date	-
ref_min_NDVI	Raster file which stores, for each pixel, the	-

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	42	17/02/2016	

	reflectance values corresponding to the minimum NDVI date	
ref_max_red	Raster file which stores, for each pixel, the reflectance values corresponding to the maximum red date	-
ntrain	Number of samples needed for the RF training	1000
TrimBands	Selection of spectral bands that will be used for the trimming	Red and NIR from the minimum NDVI date // green, red and NIR from the maximum NDVI date
x	Parameter of the MVG_IT function: vector or matrix of reflectance values of a specific class for which the Gaussian model will be fitted	E.g. subset(trimRst, LC_class == class.i) where trimRst are the selected spectral bands defined by TrimBands
alpha	Parameter of the MVG_IT function: confidence interval used to define what is an “outlier”	0.01
x	Parameters of the qchisq function: vector of quantile values	(1-alpha), where alpha is 0.01
df	Parameters of the qchisq function: degrees of freedom	number of spectral band used for the trimming : 5
x	Parameters of the mahalanobis function: vector or matrix of data	x.trim (matrix of reflectance values considered as “clean” at the current iteration)
center	Parameters of the mahalanobis function: vector of the mean distribution of each spectral band	colMeans(x.trim)
cov	Parameters of the mahalanobis function: covariance matrix of the distribution	cov(x.trim)
Output variables	Role	Default value
training_sample	Data frame which contains the samples that will be used for the training	-

Algorithm 3-6. Trimming



Algorithm Trimming is

```

## Define input working files ####
Store Eroded_Reference in “labelRst”
Store the spectral features (ref_max_slope, ref_min_slope, ref_max_ndvi, ref_min_ndvi, ref_max_red) in “featRst”
Define the number of samples needed for the RF training in “ntrain”

## Convert input working files in a data frame ####

```

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	43	17/02/2016	

Store “featRst” as data frame in “metricRst”
Store “labelRst” as data frame in “LCstack”
Store “metricRst” and “LCstack” as data frame in “Rststack”

Select reflectance values to use in the trimming (i.e. reflectance values whose Gaussian distribution will be cleaned)

Select the bands *TrimBands* (see Table 3-5 for this parameter) from “Rststack”
Select the class information corresponding to “LCstack” (class information) from “Rststack”
Store this information (selected bands and class information) in “trimRst”

Create a list of classes

List all classes ID in a list “classList”

Run the trimming, with the function MVG_IT

Create the list “trimList” that will receive the results of the trimming for the different classes

For each class ‘i’ from the “classList”

class.i <- i

X <- selection of trimRst corresponding to class.i

X.NP_IT <- **MVG_IT**(X)

trimList[i]<- including the result X.NP_IT for the class ‘i’

From the trimming output trimList, make a random selection for each class of n pixels that will be used to train the RF

For each class ‘i’ from the “classList”

subset.idx<- selects a random sample of “ntrain” pixels of the class i

out.id<-list of ones and zeros with the size of “trimList”, ‘1’ corresponding to the selected pixels (subset.idx)

trimmed_values.sbs<-subset of “trimList” corresponding to out.id==1

Training_Sample <- trimmed_values.sbs

Algorithm 3-7. MVG_IT function

Function MVG_IT is

MVG_IT <- function(x, alpha = 0.01){

chi <- **qchisq**((1-alpha), df=NCOL(x))

(computes the Chi-Squared distribution and selects the threshold corresponding to the alpha p-value of the distribution per class)

x.trim <- x

(do) while a outlier is still identified



md <- as.matrix(**mahalanobis**(x.trim, colMeans(x.trim), cov(x.trim)))

(Computing the Mahalanobis distance to the mean for each pixel)

x.trim <- x.trim which md is lower or equal to chi

(Selecting the “clean” pixels, i.e. those pixels whose mahalanobis distance to the mean is lower than the threshold)

}

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		
	Issue	Page	Date	
	1.rev.2	44	17/02/2016	

it returns x.trim as output of the function

3.5 Random forest classifier

The classification is a supervised method relying on a RF classifier, just like in the “in-situ” mode. The RF classifier is applied to the reflectance values associated with the key dates (maximum and minimum NDVI slopes, maximum and minimum NDVI, maximum red). The RF classifier uses as training data the “clean” pixels selected by the trimming.

This step follows exactly the same logical than the RF classification used for the “in-situ” mode. The presentation is thus rather brief.

3.5.1 RF model estimation

The process follows the same logical than for the “in-situ” mode, detailed in section 2.4.1.3. It is thus based on the “*otbcli_TrainImagesClassifier*” OTB application. The OTB application is composed of two steps, which are run here:



1. Random selection of training and testing samples;
2. Training of the classifier by using the previous training samples.

In our case, the random selection is already done in the previous step (as a final step of the trimming - see section 3.4). Just like for the “in-situ mode”, the developed code implemented in C++ and compatible with OTB Library therefore focuses on the second step.

Input and output variables listed in Table 3-6.

Table 3-6. Variables for the RF model estimation in the « no in-situ » mode

Input variables	Role	Default value
Feature Vector Image	Vector image containing the spectral features “ref_max_slope”, “ref_min_slope”, “ref_max_NDVI”, ref_min_NDVI”, “ref_max_date”	-
Feature Vector Image Statistics	Mean and variance of each feature composing the feature vector image	-
training_sample	Data frame which contains the samples that will be used for the training	-
rf.nbtrees	Number of decision trees	100
rf.min	Minimum number of samples per nodes	5
rf.max	Maximum depth of each tree	25
Output variables	Role	Default value
training_model	Resulting RF model to be used in the classification step	-

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	45	17/02/2016	

3.5.2 Classification



The classification process applies the RF model built in the previous step using the training dataset on the image time series in order to generate the cropland mask.

Just like for the in-situ data mode, the RF algorithm is based on a C++ code compatible with OTB which has been developed for this application. The sources can be found in the files presented in the section *Classification of the complete image times series task* of the Appendix presented at Section 5) and it is based on the “*otbcli_ImageClassifier*” application.

Input and output variables are listed in Table 3-7.

Table 3-7. Variables for the RF classification for the « no in-situ » mode

Input variables	Role	Default value
Feature Vector Image	Vector image containing the spectral features “ref_max_slope”, “ref_min_slope”, “ref_max_NDVI”, ref_min_NDVI”, “ref_max_date”	-
Feature Vector Image Statistics	Mean and variance of each feature composing the feature vector image	-
training_model	Resulting RF model to be used in the classification step	-
Output variables	Role	Default value
Pixel-based classification map	Resulting crop mask product	-

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	46	17/02/2016	

4 Common final steps

4.1 Classifier Evaluation: Computing evaluation quality measures

The classification results will be evaluated by using the in-situ data:

- Supervised Approach: in this case, it will be done by using all the vector files which contain the testing pixels and polygons which have not been used for the training. (Table 2-13).
- Unsupervised Approach: the evaluation can be done by using all the in-situ data.

The workflow of this evaluation is illustrated in Figure 4-1.

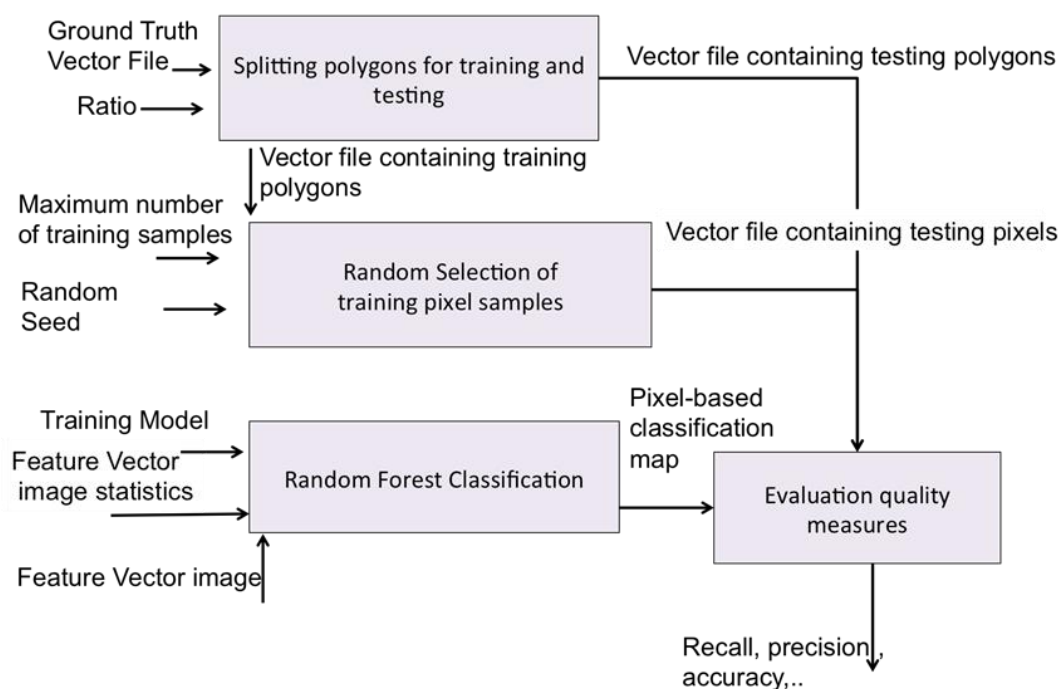


Figure 4-1. Workflow of the classification assessment

The evaluation quality indices (recall, precision, accuracy, etc.) that are computed are detailed in crop mask chapter of the Design Justification File.

All these indices are derived from a confusion matrix result. This confusion matrix can be computed by using the OTB application “*otbcli_ComputeConfusionMatrix*”, as it is shown in the Appendix of Section 5). The algorithm is run by tiles, as the whole processing chain.



	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	47	17/02/2016	

Table 4-1. Variables for the map validation algorithm

Input variables	Role	Default value
Pixel-based classification map	Resulting crop mask product	-
Validation_polygons	Vector file containing in-situ data for validation (to be used only to validate the product obtained without in-situ data)	-
Reference_polygons	Vector file containing all in-situ data (to be used only to validate the product obtained without in-situ data)	-
Output variables	Role	Default value
confusion_matrix_validation	File containing the confusion matrix	-
quality_metrics	File containing the quality metrics	-

Algorithm 4-1. Computation of quality indices

```
otbcli_ComputeConfusionMatrix -in crop-type-map.tif\
                                -out confusion-matrix-validation.csv\
                                -ref vector\
                                -ref.vector.in validation_polygons.shp\
                                -ref.vector.field Class > quality-metrics.txt
```

4.2 Crop mask post-filtering

As presented in **Erreur ! Source du renvoi introuvable.**, the post-filtering task is made of 3 steps: (i) the Principal Component Analysis (PCA), (ii) the mean-shift algorithm and (iii) the majority vote. These steps are detailed in Figure 4-2.

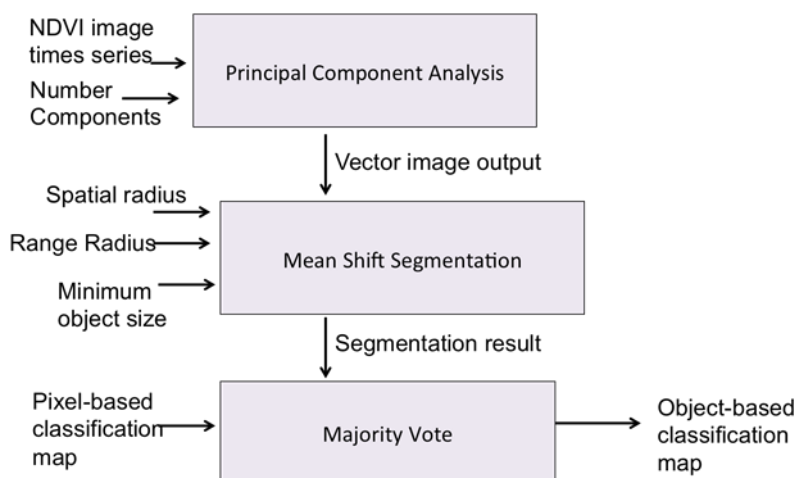




Figure 4-2. Workflow of the post-filtering step

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	48	17/02/2016	

4.2.1 Principal Component Analysis

The Principal Component Analysis steps is run with a ++ code compatible with OTB, which has been developed (see code described in section *Principal Component Analysis task* of the Appendix presented in Section 5). It uses the otb filter class “*otb::PCAIImageFilter*”.

Input and output variables are listed in Table 4-2.

Table 4-2. Variables for the Principal Component Analysis

Input variables	Role	Default value
NDVI image times series	Vector image containing the complete NDVI image times series	-
Number of Components	Number of principal components composing the output image	Maximum: 6 images Minimum: number of available images
Output variables	Role	Default value
Vector image	Vector image containing the “Number of Components” principal component images	-



4.2.2 Mean-Shift segmentation algorithm

The Mean-Shift segmentation is run with a ++ code compatible with OTB, which has been developed (see the code described in the section *Mean Shift Segmentation task* of the Appendix presented in Section 5). It uses the OTB filter class “*otb::MeanShiftVectorImageFilter*”.

Input and output variables are listed in Table 4-3.

Table 4-3. Variables for the Mean-Shift segmentation

Input variables	Role	Default value
Vector image output	vector image containing the result of the PCA	-
Spatial radius	spatial radius of the neighborhood	10
Range radius	range radius defining the radius (expressed in radiometry unit) in the multispectral space	0.65
Minimum object size	Minimum size of a region (in pixel unit) in segmentation. Smaller clusters will be merged to the neighboring cluster with the closest radiometry. If set to 0 no pruning is done	10
Output variables	Role	Default value
Segmentation result	segmentation result image	-

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	49	17/02/2016	

4.2.3 Majority vote

The majority vote algorithm is presented in Algorithm 4-2.

Input and output variables are listed in Table 4-4.

Table 4-4. Variables for the majority voting filtering

Input variables	Role	Default value
Segmentation result	Segmentation result image	-
Pixel-based classification map	Pixel-based classification results obtained by the RF classifier (coming from the “in-situ” mode or not)	-
Output variables	Role	Default value
Object-based classification map	Object-based classification map result	-

Algorithm 4-2. Majority vote filtering



```

Data: Input_seg_image, ClassificationMap
Result: Result_image
begin
  for region  $\in$  Input_seg_image do
    RegionClass  $\leftarrow$  Compute_the_majority_class(ClassificationMap(region));
    for pixelP  $\in$  region do
      | Result_image(pixelP)  $\leftarrow$  RegionClass;
    end
  end
end

```

4.2.4 Operational context

It must be remembered that in the operational context, the post-filtering task must also be done following an iterative procedure. This iterative approach will be possible from the first satellite data acquisitions. This can be obtained by following the next flowcharts, Figure 4-3 and Figure 4-4.

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2 + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	50	17/02/2016	

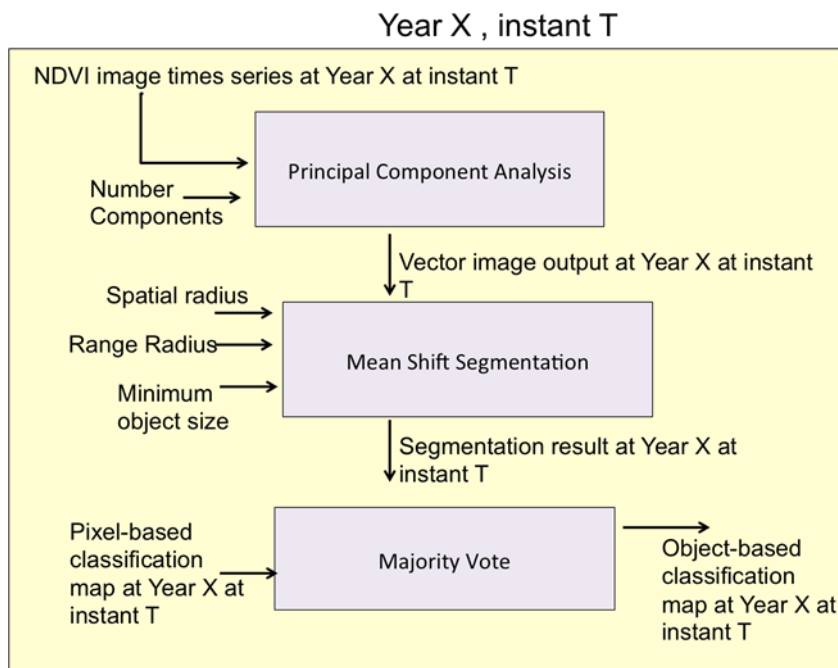


Figure 4-3. Post-filtering workflow, at instant T

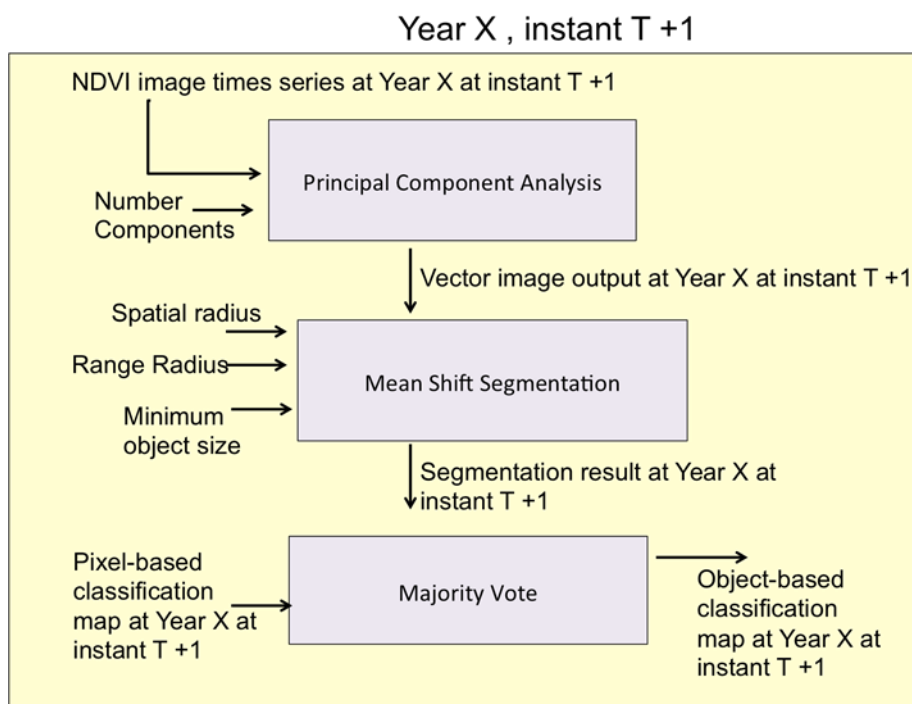





Figure 4-4. Post-filtering workflow, at instant T+1

	Ref	Sen2Agri_DDF_v1.2 L4 dynamic crop mask ATBD		 sentinel-2  + AGRICULTURE
	Issue	Page	Date	
	1.rev.2	51	17/02/2016	

5 Appendix - Additional information on the codes of the « in-situ » processing chain

The code, associated with the « in-situ » mode (including the common steps with the “no in-situ” mode) is available on: http://tully.ups-tlse.fr/valeros/s2agri_cropmask

Codes from the OTB project are the following ones:

- **NDVI feature computation task:**

Includes/NdviParameterExtraction.hpp

tests/IndicesComputation/NdviParameterExtractionApplication.cxx

- **Statistics feature computation task:**

includes/IndexParameterExtraction.hpp

tests/IndicesComputation/IndexParameterExtractionApplication.cxx

- **Random selection of training pixel samples task:**

includes/ShpFileSampleGeneratorFast.hpp

src/ShpFileSampleGeneratorFast.cxx

- **Random Forest Training Step task:**

test/sClassification/ ShpFileMRFCClassificationFast.cxx

- **Classification of the complete image times series task:**

tests/Classification/SVMClassificationMap.cxx

- **Principal Component Analysis task:**

tests/PostFiltering/ACP.cxx

- **Mean Shift Segmentation task:**

tests/PostFiltering/SegmentationACP.cxx

Two algorithms were coded in Python:

- Quality measures (how to evaluate the crop masks – section 4.1)
- Majority vote (how to apply the majority vote – section 4.2.3)