Climate Change Initiative Extension (CCI+) Phase 1

New Essential Climate Variables (NEW ECVS)

High Resolution Land Cover ECV (HR_LandCover_cci)

Algorithm Theoretical Basis Document

(ATBD)

Prepared by:

Università degli Studi di Trento
Fondazione Bruno Kessler
Università degli Studi di Pavia
Università degli Studi di Genova
Université Catholique de Louvain
Politecnico di Milano
Université de Versailles Saint Quentin
CREAF
e-GEOS s.p.a.
Planetek Italia
GeoVille

Changelog

Issue	Changes	Date
1.0	First version.	02/07/2019
1.1	Revision according to "CCI_HRLC_Ph1_KO+9_RID_ESA.xlsx"	25/09/2019
2.0	Document entirely updated according to current status of the processing chain. Final legend is included.	03/01/2020

Detailed Change Record

Issue	RID	Description of discrepancy	Sections	Change
1.1	FR-01	The ATBD should report a detailed description of the algorithms and methodologies (reported in the technical proposal) that should be used to achieve the objective of the project. We understand that RR#1 activities will provide better indications on which algorithm candidates on classification, but a more detailed description of the listed methods is needed.	Sections 6,7,8	Sections are integrated with more detailed information and mathematical insights.
1.1	FR-02	Why as Global Product to use as reference the unique map described is CORINE LC? CLC is not global.	7.1.1 (removed)	Mention to CORINE LC product as global product has been removed.
1.1	FR-03	Why training the S1 data using as reference the 300m CCI-LC maps? We are going to lose the HR of S1 data, or am I wrong? Please add some reference document using this technique	8	Further clarification has been added.

Contents

1 Introduction

1.1 Executive summary

Algorithm development is specifically driven to address the technical requirements as provided by the outcome of Task 1 of the project. Best performing algorithms have been selected among proposed candidates through an internal benchmarking-testing iteration by the Earth Observing Science (EOS) team. The processing chain as developed by the end of the first year of project activity is presented in this version of the document.

1.2 Purpose and scope

The Algorithm Theoretical Basis Document (ATBD) details algorithms in the processing chain needed to produce the land cover products as presented in the PSD [AD3]. It is intended to provide information for the understanding of the processing chain. The ATBD version 2.0 aims at providing the first version of the processing chain, as the outcome of the first year of activities devoted to benchmarking and testing. This version of the document integrates in the whole project workflow as illustrated in Figure 1.

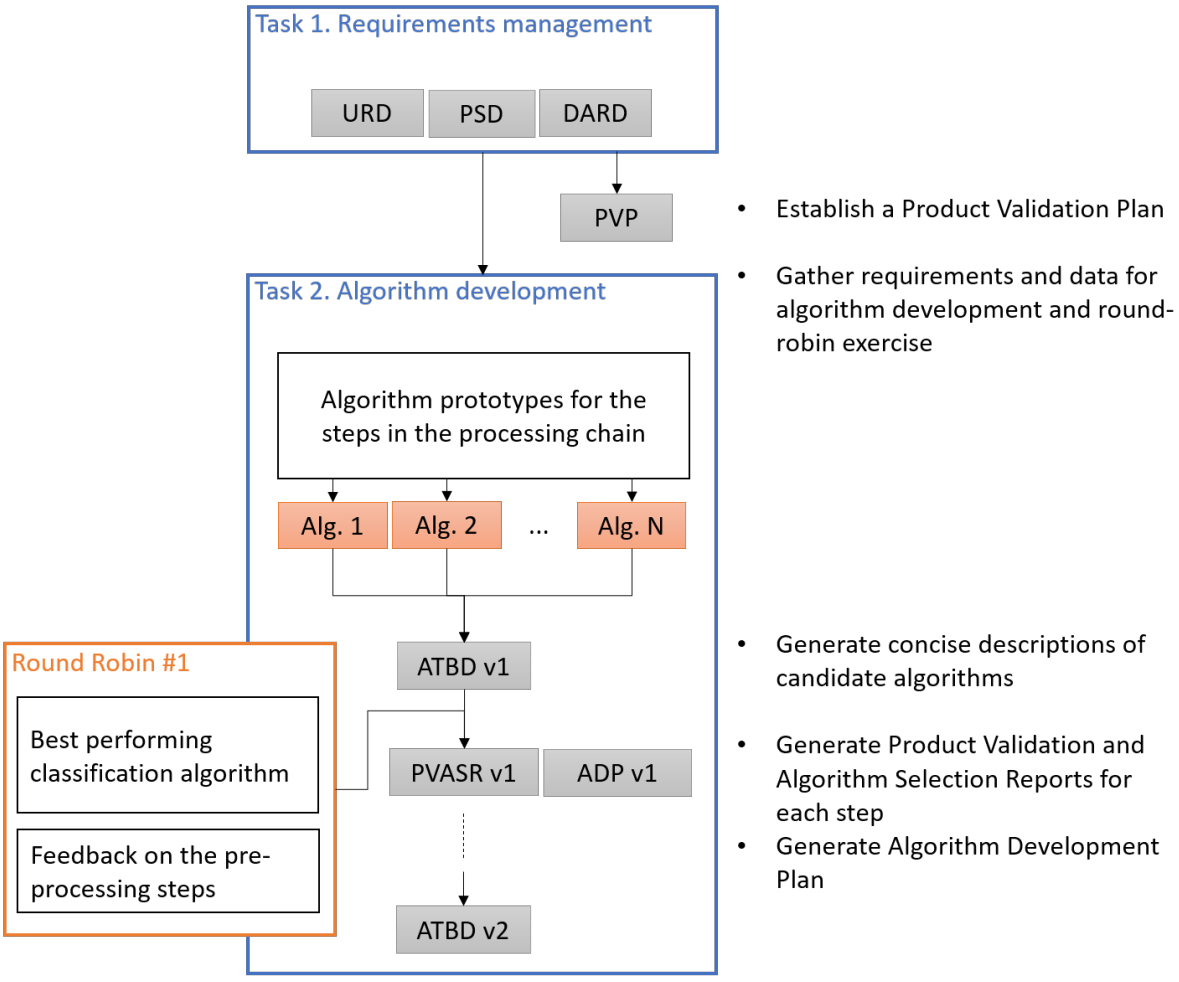


Figure 1. Concept of the ATBD v1 in the workflow of Task 2 of the CCI+ HRLC project.

The main blocks of computation can be identified as:

- Optical pre-processing.
- SAR pre-processing.
- Multi-sensor geolocation.
- Optical data classification.
- SAR data classification.

- Decision fusion.
- Multitemporal change detection and trend analysis.

1.3 Applicable documents

Ref. **Title, Issue/Rev, Date, ID**

[AD1] CCI HR Technical Proposal, v1.1, 16/03/2018

[AD2] CCI Extension (CCI+) Phase 1 – New ECVs – Statement of Work, v1.3, 22/08/2017, ESA-CCI-PRGM-EOPS-SW-17-0032.

[AD3] CCI_HRLC_Ph1-PSD, latest version

[AD4] CCI_HRLC_Ph1-URD, latest version

1.4 Acronyms and abbreviations

6S	Second Simulation of a Satellite Signal in the Solar Spectrum
AC	Atmospheric correction
AMI	Active Microwave Instrument
AOT	Aerosol Optical Thickness
ASAR	Advanced Synthetic Aperture Radar
ATBD	Algorithm Theoretical Basis Document
BEAST	A Bayesian Estimator of Abrupt change, Seasonal change, and Trend
BFAST	Breaks For Additive Seasonal and Trend
BOA	Bottom of Atmosphere
BoW	Bag of visual Words
CCI	Climate Change Initiative
CD	Change Detection
CFMask	C Version of Function Of Mask
CMA	Climate Modeling Grid - Aerosol
CMG	Climate Modeling Grid
CNN	Deep Convolutional Neural Network
CVA	Change Vector Analysis
DARD	Data Access Requirement Document
DDV	Dark Dense Vegetation
DEM	Digital Elevation Model
DM	Dissimilarity Measure
DTW	Dynamic Time Warping
ECV	Essential Climate Variables
ERS	European Remote Sensing
ETM	Enhanced Thematic Mapper
ETM+	Enhanced Thematic Mapper Plus
FC	Fully Connected
FS	Feature Space
GCOS	Global Climate Observing System
GMM	Gaussian Mixture Model

GSFC	Goddard Space Flight Center
HLS	Harmonized Landsat/Sentinel-2
HR	High Resolution
IFK	Improved Fisher Kernel
INT	Integer
IRMAD	Iteratively-Reweighted Multivariate Alteration Detection
L-5/7/8	Landsat-5/7/8
LandTrendr	Landsat-based detection of Trends in Disturbance and Recovery
LaSRC	Landsat Surface Reflectance Code
LC	Land Cover
LCC	Land Cover Change
LEDAPS	Landsat Ecosystem Disturbance Adaptive Processing System
LLC	Locality-constrained linear coding
LOP	Linear Opinion Pool
LPF	Low Pass Filter
LSTM	Long Short Term Memory
LTS	Landsat Time Series
LUT	Lookup Table
MDDTW	Multi-Dimension DTW
MEaSUREs	Making Earth Science Data Records for Use in Research Environments
MF-DTW	Multi-Feature DTW
MGRS	Military Grid Reference System
MLP-NN	Multi-Layer Perceptron Neural Network
MMU	Minimum Mapping Unit
MODIS	Moderate Resolution Imaging Spectroradiometer
MR	Medium Resolution
MSS	Multispectral Scanner
NA	Not Applicable
NASA	National Aeronautics and Space Administration
NCEP	National Centers for Environmental Prediction
NDI	Normalized Difference Index
NDVI	Normalized Difference Vegetation Index
NIR	Near infrared
NSPI	Neighbourhood Similar Pixel Interpolator
OA	Overall Accuracy
OLI	Operational Land Imager
OMI	Ozone Monitoring Instrument,
PCA	Principal Component Analysis
PSD	Product Specification Document
QA	Quality Assessment
RBF	Radial Basis Function

RD	Range Doppler
REFEREE	Learning a transferable change Rule From a recurrent neural network for change detection
RF	Random Forest
RNN	Recurrent Neural Network
S-1/2	Sentinel-1/2
S2AC	Sentinel-2 Atmospheric Correction
SAR	Synthetic Aperture Radar
SIFT	Scale Invariant Feature Transform
SITS	Satellite Image Time Series
SLC	Scan-line corrector
SM	Similarity Measure
SoW	Statement of Work
SR	Surface Reflectance
SRTM	Shuttle Radar Topography Mission
SSFA	Supervised Slow Feature Analysis
ST	Similarity Trend
STWR	Spatially and temporally weighted regression
SVM	Support Vector Machine
SWIR	Short-wave infrared
TIMESAT	Time Series of Satellite data
TIRS	Thermal Infrared Sensor
TM	Thematic Mapper
TOA	Top of Atmosphere
TOMS	Total Ozone Mapping Spectrometer
TS	Time Series
UTM	Universal Transverse of Mercator
VHR	Very High Resolution
VLAD	Vector of locally aggregated descriptors
WGS84	World Geodetic System 1984
XML	Extensible Markup Language

2 Processing chain overview

The CCI HRLC project will deliver to the climate community regional land cover (LC) and land cover change (LCC) products over three areas in Africa Sahel band, Amazon and Siberia URD [AD4]. LC maps will be provided at 10m resolution for year 2018 (the so-called Static Map) and at 30m resolution for the historical record of LC and LCC from 1990 on, every five years. The high-resolution classification legend as agreed by the Consortium is listed in URD [AD4]. The processing chain under development, outlined in Figure 2, is novel and it does not rely on already existing land cover products.

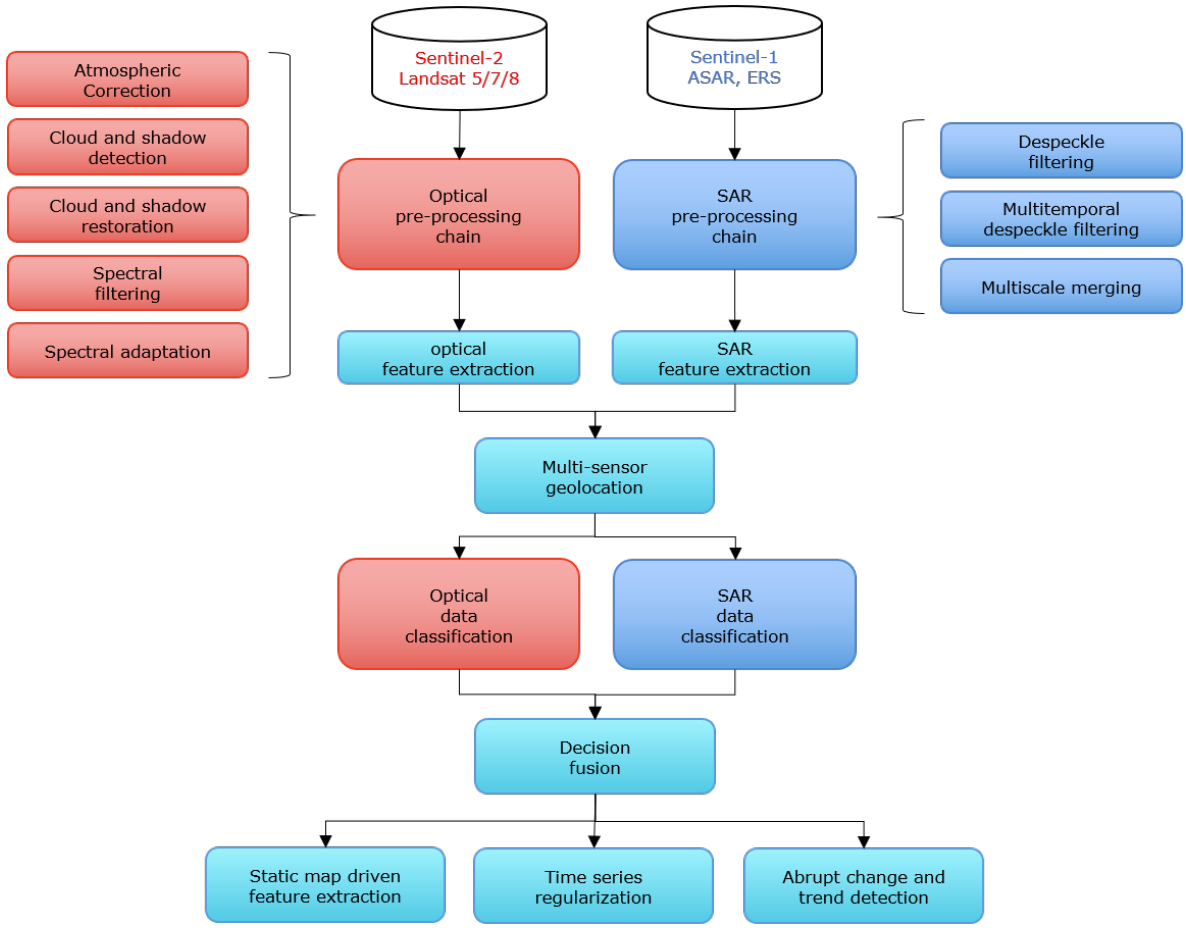


Figure 2. Block-based representation of the processing chain for the production of HRLC maps.

The high-level workflow of the processing chain is presented in Figure 2. Optical multispectral imagery is the main source of data as input for the classification. The optical processing chain is consistent with the possibility to work mainly with images at 10/30m resolution and generating an output at 10/30m, based on multitemporal multispectral data from S-2 and L-8 in the recent years and legacy Landsat-5/7/8 data in the past. The SAR processing chain will be implemented mainly for S-1 in the recent years, and ERS and ASAR data sets in the past (whenever and wherever HR mode data are available). Microwave data sets are useful for classes where SAR has proven to be accurate at medium resolution, such as water bodies and coastal lines, and the option to use SAR for urban areas is considered as well. The products obtained by the optical and the SAR processing chains will be then integrated in the data fusion module in order to produce the final HRLC products. This design choice of fusion at the decision level makes it possible to develop advanced and ad hoc processing approaches for optical, SAR, and multisensor data, while keeping the system modular and scalable. The output products will be then

analyzed in the multitemporal change detection and trend analysis block for identifying different change components to be used for the historical time series HRLC products every 5 years.

Table 1. Final high resolution land cover classification legend defined by the Climate Research Group for the choice of the best performing classification algorithm.

LC CLASS							
CODE	**1st LEVEL**	**CODE**	**2nd LEVEL**	**CODE**	**3rd LEVEL**	**CODE**	**4th LEVEL**
PRIMARILY VEGETATED CLASSES							
Areas where the sum of all vegetation cover exceeds 50 % at the time of fullest development and where the snow and/or ice, open water or built-up cover less than 50% of the surface. Areas where the lifeform can be further distinguished into trees, shrubs, cropland, grassland and lichens and mosses.							
10	Tree cover evergreen broadleaf						
20	Tree cover evergreen needleleaf						
30	Tree cover deciduous broadleaf						
40	Tree cover deciduous needleleaf						
50	Shrub cover evergreen						
		51	Broadleaf				
		52	Needleleaf				
60	Shrub cover deciduous						
		61	Broadleaf				
		62	Needleleaf				
70	Grasslands						
		71	Natural or semi-natural				
		72	Managed (pastures)				
80	Croplands						
		81	Winter crops				
				811	Rainfed		
				812	Irrigated		
						8121	Sparkling
						8122	Flooding
		82	Summer crops				
				821	Rainfed		
				822	Irrigated		
						8221	Sparkling
						8222	Flooding
		83	Multicropping				
				831	Rainfed		
				832	Irrigated		
						8321	Sparkling
						8322	Flooding
90	Vegetation aquatic or regularly flooded						
100	Lichen and Mosses						
PRIMARILY NON VEGETATED CLASSES							

Areas where the sum of all vegetation cover is below 50 % at the time of fullest development. Snow and/or ice, open water or built-up cover less than 50% of the surface.							
110	Bare areas						
		111	Unconsolidated	1111	Sands		
				1112	Bare soil		
		112	Consolidated				
AREAS DOMINATED BY THE BUILT-UP, SNOW/ICE OR WATER COVER CLASSES							
Areas where the snow and/or ice, open water or built-up cover more than 50 % of the surface. Areas where the sum of all vegetation cover is below 50 % at the time of fullest development.							
120	Built-up						
		121	Buildings				
		122	Artificial roads				
130	Open Water seasonal						
140	Open Water permanent						
150	Snow and/or Ice						
		151	Snow				
		152	Ice				

3 Optical pre-processing

Pre-processing operations are intended to correct for sensor- and platform-specific radiometric and geometric distortions of data and harmonization. Radiometric corrections may be necessary due to variations in scene illumination and viewing geometry, atmospheric conditions, and sensor noise and response. Each of these will vary depending on the specific sensor and platform used to acquire the data and the conditions during data acquisition. Cloud coverage is a systematic issue related to optical imagery and it requires specific processing aimed at precisely locating cloud and shadow pixels, with possible restoring steps to recover spectral information over occluded pixel locations. All the steps needed to prepare optical images for classification, see Figure 3, are detailed in the following sections.

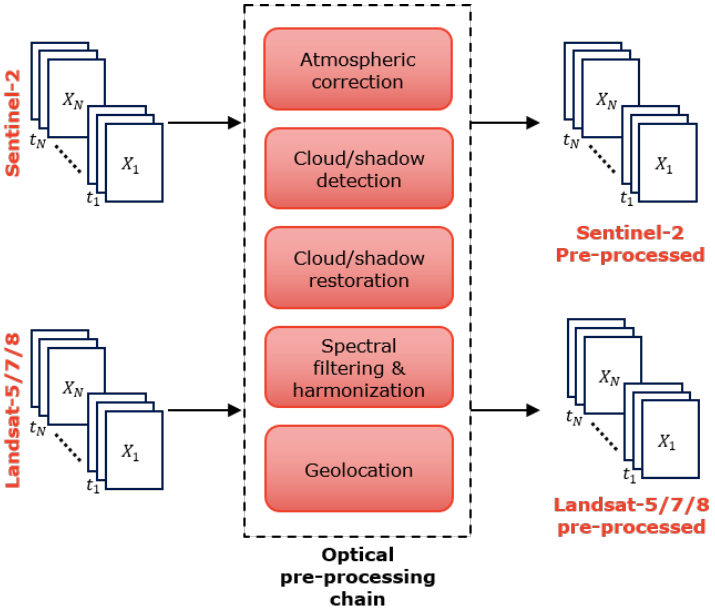


Figure 3. Optical pre-processing chain.

3.1 Atmospheric correction

Surface Reflectance (SR) is the amount of light reflected by the surface of the Earth. It is a ratio of surface radiance to surface irradiance, and as such is unitless, with values between 0-1. Working with SR allows for a meaningful comparison between multitemporal images acquired over the same region by compensating for atmospheric effects such as aerosol scattering and thin clouds, thus allowing for the detection and characterization of Earth surface changes.

3.1.1 Sentinel-2 – sen2cor

The sen2cor processor allows calculation of Bottom of Atmosphere (BOA) reflectance from Top of Atmosphere (TOA) reflectance images available in Level-1C products. Sentinel-2 atmospheric correction (S2AC) is based on an algorithm proposed in [1]. The method performs atmospheric correction based on the LIBRADTRAN radiative transfer model presented in [2].

The model is run once to generate a large LUT of sensor-specific functions (required for the AC: path radiance, direct and diffuse transmittances, direct and diffuse solar fluxes, and spherical albedo) that accounts for a wide variety of atmospheric conditions, solar geometries and ground elevations. This database is generated with a high spectral resolution (0.6 nm) and then resampled to S-2 spectral responses. This LUT is used as a simplified model (running faster than the full model) to invert the radiative transfer equation and to calculate BOA reflectance. All gaseous and aerosol properties of the atmosphere are either derived by the algorithm itself or fixed to an a priori value.

S2AC employs Lambert's reflectance law. Topographic effects can be corrected during the surface retrieval process using an accurate Digital Elevation Model (DEM). S2AC accounts for and assumes a constant viewing angle per tile (sub-scene). The solar zenith and azimuth angles can either be treated as constant per tile or can be specified for the tile corners with a subsequent bilinear interpolation across the scene.

3.1.2 Landsat 5/7/8 – LEDAPS, LaSRC

Landsat-4/5 TM and Landsat-7 ETM+ Surface Reflectance are generated using the Landsat Ecosystem Disturbance Adaptive Processing System (LEDAPS) algorithm, a specialized software originally developed through a National Aeronautics and Space Administration (NASA) Making Earth System Data Records for Use in Research Environments (MEaSUREs) grant by NASA Goddard Space Flight Center (GSFC) and the University of Maryland [3]. The software applies Moderate Resolution Imaging Spectroradiometer (MODIS) atmospheric correction routines to Level-1 data products. Water vapor, ozone, geopotential height, aerosol optical thickness, and digital elevation are input with Landsat data to the Second Simulation of a Satellite Signal in the Solar Spectrum (6S) radiative transfer models to generate TOA reflectance, surface reflectance, TOA brightness temperature, and masks for clouds, cloud shadows, adjacent clouds, land, and water. Landsat 8 OLI Surface Reflectance are generated using the Landsat Surface Reflectance Code (LaSRC) [4], which makes use of the coastal aerosol band to perform aerosol inversion tests, uses auxiliary climate data from MODIS, and a unique radiative transfer model. LaSRC hardcodes the view zenith angle to “0”, and the solar zenith and view zenith angles are used for calculations as part of the atmospheric correction.

While both the LEDAPS and LaSRC algorithms produce similar SR products, the inputs and methods to do so differ. The table below illustrates both of them.

Table 2. Differences between Landsat-4/5/7 and Landsat-8 surface reflectance algorithms.

Parameter	Landsat-4/5/7 (LEDAPS)	Landsat-8 (LaSRC)
Global Coverage	Yes	Yes
TOA Reflectance	Visible (Bands 1–5,7)	Visible (Bands 1–7, 9 OLI)

TOA Brightness Temperature	Thermal (Band 6)	Thermal (Bands 10 & 11 TIRS)
SR	Visible (Bands 1-5, Band 7)	Visible (Bandsat 1-7) (OLI only)
Thermal bands used in Surface Reflectance processing	Yes (Brightness temperature Band 6 is used in cloud estimation)	No
Radiative transfer model	6S	Internal algorithm
Thermal correction level	TOA only	TOA only
Thermal band units	Kelvin	Kelvin
Pressure	NCEP Grid	Surface pressure is calculated internally based on the elevation
Water vapor	NCEP Grid	MODIS CMA
Air temperature	NCEP Grid	MODIS CMA
DEM	GTOPO5	GTOPO5
Ozone	OMI/TOMS	MODIS CMG Coarse resolution ozone
AOT	Correlation between chlorophyll absorption and bound water absorption of scene	MODIS CMA
Sun angle	Scene center from input metadata	Scene center from input metadata
View zenith angle	From input metadata	Hard-coded to "0"
Undesirable zenith angle correction	SR not processed when solar zenith angle > 76 degrees	SR not processed when solar zenith angle > 76 degrees
Pan band processed	No	No
XML metadata	Yes	Yes
Top of Atmosphere Brightness Temperature calculated	Yes (Band 6 TM/ETM+)	Yes (Band 10 & 11 TIRS)
Cloud mask	CFMask	CFMask
Data format	INT16	INT16
Fill values	-9999	-9999
QA bands	Cloud Adjacent cloud Cloud shadow DDV Fill Land water Snow Atmospheric opacity	Cloud Adjacent cloud Cloud shadow Aerosols Cirrus Aerosol In

3.2 Cloud and cloud shadow detection

Identification of clouds, cloud shadows in optical images is necessary. The well-known program named Fmask has been designed to accomplish these tasks for use with images from Landsat-4/5/7 [5]. Recently, these results have been improved and extended to Landsat-8 and Sentinel-2 [6]. The processing chain will therefore implement the last available improvements:

- improved Fmask algorithm for Landsat-4/5/7;
- a new version for use with Landsat-8 that takes advantage of the new cirrus band;
- a prototype algorithm for Sentinel 2 images.

Though Sentinel 2 images do not have a thermal band to help with cloud detection, the new cirrus band is found to be useful for detecting clouds, especially for thin cirrus. By adding a new cirrus cloud probability and removing the steps that use the thermal band, the Sentinel-2 scenario achieves noticeable improvements with respect to Landsat.

3.3 Cloud and cloud shadow restoration

When addressing the presence of clouds, it is necessary to exploit the temporal information. The most accurate results have indeed been achieved by hybrid methods, which usually combine the temporal information with spatial/spectral features. Recently, different approaches have proposed to solve this issue. Worth mentioning are cloud removal technique based on a modified neighbourhood similar pixel interpolator (MNSPI) [7], and a more sophisticated cloud removal method which combines multitemporal and dictionary learning methods [8]. Although these methods accurately solve the cloud restoration problem, they are time-consuming and inoperable for the project purposes.

The solution is therefore a fast and effective technique recently proposed that can achieve a good trade-off between restoration accuracy and computational burden. In greater detail, the method:

- automatically detects a short TS (4-5 images) of cloud free image temporally close to the target one;
- for each non-valid pixel of the target image, it identifies the most similar pixels present in the scene by analyzing their temporal patterns;
- restores invalid pixels using the most similar ground-clear one in the target image.

To reduce the computational effort, the method exploits a short TS of images and performs the detection of similar temporal patterns using an efficient KD-tree search algorithm.

3.4 Spectral filtering and harmonization

3.4.1 Landsat-7 SLC-off

The scan-line corrector (SLC) of the Landsat-7 Enhanced Thematic Mapper Plus (ETM+) sensor failed in 2003, resulting in about 22% of the pixels per scene not being scanned. The SLC failure has seriously limited the scientific applications of ETM+ data. While there have been several methods developed to fill in the data gaps, each method has shortcomings, especially for heterogeneous landscapes. Based on the assumption that the same-class neighbouring pixels around the un-scanned pixels have similar spectral characteristics, and that these neighbouring and un-scanned pixels exhibit similar patterns of spectral differences between dates, recently, a simple and effective method has been developed that interpolates the values of the pixels within the gaps [9]. We refer to this method as the Neighborhood Similar Pixel Interpolator (NSPI). Results indicate that NSPI can restore the value of un-scanned pixels very accurately, and that it works especially well in heterogeneous regions. In addition, it can work well even if there is a relatively long-time interval or significant spectral changes between the input and target image. The filled images appear reasonably spatially continuous without obvious striping patterns.

Relevant to the CCI HRLC project is that, supervised classification was done for validation on both gap-filled simulated SLC-off data and the original “gap free” data set, and it was found that classification results, including

accuracies, were very comparable. This gives nice expectations for this algorithm to provide gap-filled products generated by NSPI. In addition, the simple principle and high computational efficiency of NSPI will enable processing large volumes of SLC-off ETM+ data.

3.4.2 Landsat radiometric normalization

Land-cover changes generally alter the reflectance of the land surface, which can be detected using multitemporal Landsat data sets. The analysis of land-cover change using multitemporal Landsat data is complicated by the presence of substantial radiometric differences between Landsat scenes. This is due mainly to drifting in the radiometric performance of the individual sensors over time [10].

Regression techniques are based on the observation that within a given spectral band there is an overall linear relationship between the reflectance values for two images acquired on the same ground area. In image pairs where such linearity exists, regression analysis can be used to derive a gain and offset for radiometrically normalizing the subject image to match the reference image [10].

To address radiometric normalization, a recent approach based on spatially and temporally weighted regression (STWR) model for cloud removal to produce continuous cloud-free Landsat images [11] is used. This method makes full utilization of cloud-free information from input Landsat scenes and employs a STWR model to optimally integrate complementary information from invariant similar pixels. Moreover, it integrates a prior modification term to minimize the biases derived from the spatially-weighted-regression-based prediction for each reference image.

3.4.3 Sentinel-2 / Landsat data harmonization

Both project requirements of mapping LC and LCC historically back to 1990 and the backpropagation approach (Sentinel-2 Static Map of 2018 as a reference and LCC-drive mapping backward in time) require strong integration between Sentinel-2 and Landsat data and call for the application of a concept that has been termed Analysis Ready Data (ARD) [12]. Data products must be gridded to a common reference and processed to comparable geophysical parameters regardless of their sensor of origin [13]. The Harmonized Landsat and Sentinel-2 (HLS) project [14] is a NASA initiative aimed at producing a Virtual Constellation of surface reflectance data acquired by the Operational Land Imager (OLI) and Multispectral Instrument (MSI) aboard Landsat-8 and Sentinel-2 remote sensing satellites, respectively.

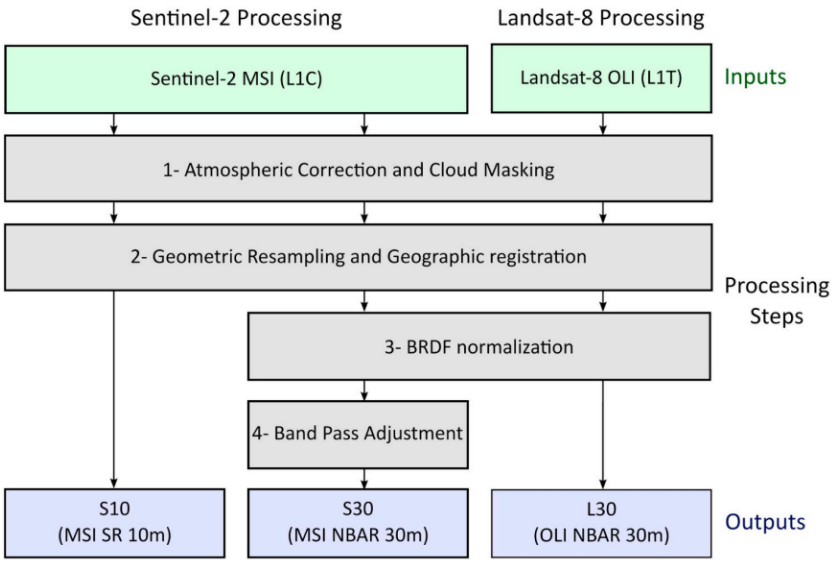


Figure 4. Overview of the Harmonized Landsat/Sentinel-2 processor.

The HLS products are based on a set of algorithms, see Figure 4, to obtain seamless products from both sensors (OLI and MSI): atmospheric correction, cloud and cloud-shadow masking, spatial co-registration and common

gridding, bidirectional reflectance distribution function normalization and spectral bandpass adjustment. Three products are derived from the HLS processing chain:

- **S10**: full resolution MSI SR at 10 m, 20 m and 60 m spatial resolutions;
- **S30**: a 30 m MSI Nadir BRDF (Bidirectional Reflectance Distribution Function)-Adjusted Reflectance (NBAR);
- **L30**: a 30 m OLI NBAR. All three products are processed for every Level-1 input products from Landsat 8/OLI (L1T) and Sentinel-2/MSI (L1C).

4 SAR pre-processing

We considered Sentinel-1 data acquired in Interferometric Wide swath (IW) mode and Ground Range Detected (GRD) type, which derive from an application of a proper multi-looking and ground range projection based on an Earth ellipsoid model. The datasets are in High resolution (HR) and provide images with a native range by azimuth resolution 20×22 meters and pixel spacing equals to 10x10 m. Over land surfaces, the orbital period of each satellite is about 12 days. Consequently, acquisitions have been available since 2015 for time-periods of 6 or 12 days depending on the study region.

For processing and analyzing the data, several codes have been developed in Python programming language, which were then deployed by means of dockers, i.e. general automated applications that can be launched in every OS.

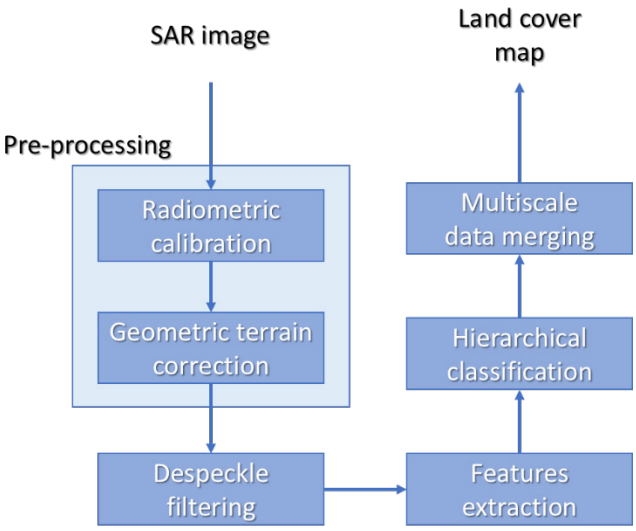


Figure 5. Block scheme of how SAR pre-processing integrates in the whole SAR processing chain.

Before applying any classification algorithm on S1 data, a preliminary pre-processing phase is required, and it consists in the following basic steps, see also Figure 5:

- Radiometric calibration of data;
- Geometric terrain correction;
- Despeckle filtering.

4.1 Radiometric calibration

Radar images are firstly calibrated with respect to their intrinsic sensor and signal acquisition properties, for expressing the echoes of distributed target (e.g. grass, dirt, etc.) in terms of the radar backscattering coefficient. In other words, the VV and VH intensities are expressed in terms of sigma naught. Generally, this operation was performed during the generation of a SAR product, but for the land cover map generation is not recommended to use raw data because of the inconsistency of the uncalibrated signal. The radiometric calibration is therefore

needed since the grey-value of SAR imagery must be adjusted respect the backscattering signals of the objects present into the scene.

4.2 Geometric terrain correction

Due to the active nature of the system, every SAR image is acquired in slant looking geometry. If the ground is elevated because of hills and valleys, the time of the signal to travel to the Earth surface and back to the sensor is distorted, causing geometric shifts in the image (foreshortening, layover and shadow). These can only be corrected if a model representing the topography under the image is known. In particular, the Range Doppler (RD) Terrain Correction is applied, and it shifts all pixels to their correct locations according to ancillary data Shuttle Radar Topography Mission (SRTM) 3 arc sec (i.e. around 20 m of resolution) DEM as input. RD Terrain Correction increases the location accuracy of your image. The first two steps of pre-processing phase have been conducted using ESA Sentinel-1 toolbox implemented in the official Sentinel Application Platform software provided by ESA (for more detailed information, ones should refer to the proper Wiki for Developer Documentation to [15].

4.3 Despeckle filtering

The SAR images are inherently affected by speckle that is a "noise like" signal due to the coherent nature of the electromagnetic scattering [16]. Even though speckle carries itself information about the illuminated area, it degrades the appearance of images and affects the performance of scene analysis tasks carried out by computer programs (e.g., segmentation and classification). To mitigate this problem several suitable filtering methodologies have been developed for reducing the disturbance significantly and preserve at the same time all the relevant scene features, such as radiometric and textural information. The speckle in SAR is a *multiplicative* effect, i.e. it is in direct proportion to the local grey level in any area. Speckle filtering is needed to suppress the noise in order to allow better interpretation and backscatter analysis. However, it is essential mentioning that the speckle filter not only suppress the noise, but also remove observations that are not affected by noise and contain valuable land surface information (i.e. soil moisture, biomass and flood extent). The process of removal of speckle in SAR image is very essential for the analyst to interpret. A filter should remove speckle without sacrificing image structures.

There are various speckle removal methods. Speckle removal is necessary for quantitative, analysis but there exists a tradeoff between speckle removal and resolution. Speckle Suppression can be done using various techniques. The first technique is Lee filter, known for being one of the first approach designed for suppressing speckle effect [17]. Second technique is time series-based processing. Proper developed docker containers provide both classical Lee method and a better suitable and advanced de-speckle filter (called *multitemporal de-speckle filter)* that exploits a SAR time series. Multitemporal denoising methods take advantage of the increasing availability of SAR image time-series to solve the spatial denoising problems, for the benefit of a better spatial resolution preservation.

4.3.1 Lee filter

The Lee filter is an adaptive filter, and reportedly to be the first model-based filter dedicated to speckle noise suppression [18]. It is also derived from the Minimum Mean-Square Error (MMSE) algorithm that converts the multiplicative model into an additive one, thereby reducing the problem of dealing with speckle to a known tractable case (more details are reported in [19]). In Lee filter, the statistical distribution of the values of the pixels within the moving kernel is utilized to estimate the value of the pixel of interest. This assumes that the mean and variance of the pixel of interest are equal to the local mean and local variance of all pixels within the user-selected moving kernel. The resulting grey level value Y for the smoothed pixel is:

$$Y = I_c W + I_m(1 - W),$$

where:

- $W = \left(1 - \frac{{C_u}^2}{{C_i}^2}\right)$;
- $C_u = \sqrt{1/ENL}$;
- $C_i = S/I_m$
- I_c is the central pixel of filter kernel;
- I_m value is the mean between all pixels falling within kernel;
- S is the standard deviation of all pixels falling within kernel;

W is the weighting function that measures the estimated noise variation coefficient C_u over the image variation coefficient C_i. The number of looks parameter ENL is the Equivalent Number of Looks of the radar image, which is used to estimate the noise variance and control the amount of smoothing applied to the image by the filter. The user may experimentally adjust the ENL value to control the effect of the filter. A small ENL value leads to more smoothing while a large ENL preserves more image features.

Several works [20], [21] have proven, with quantitative assessments, that a good tradeoff between speckle suppression, details and textures preservation is achieved with 5x5 or 7x7 moving kernel size. Moreover, the Lee filter is reportedly superior in its ability to preserve prominent edges, linear features, point target, and texture information, by minimizing either the mean square error or the weighted least square estimation.

4.3.2 Multitemporal despekle filter

The proposed approach is a ratio-based multitemporal denoising framework based on the use of a ratio image composed of a noisy image and the temporal mean of the stack. This ratio image is easier to denoise than a single image thanks to its improved stationarity. Besides, temporally stable thin structures are well preserved thanks to the multi-temporal mean [22]. Because of the improved spatial stationarity of the ratio images, denoising these ratio images with a speckle-reduction method is more effective than denoising images from the original multi-temporal stack. The amount of data that is jointly processed is also reduced compared to other methods through the use of the 'super-image' that sums up the temporal stack in order to fully exploit the significant information of the multi-temporal stack.

The method consists in three steps that are grouped into the following list and represented in Figure 6:

1. Super image;
2. Denoising of the ratio image;
3. Computation of the final image through the multiplication between denoised ratio and super image.

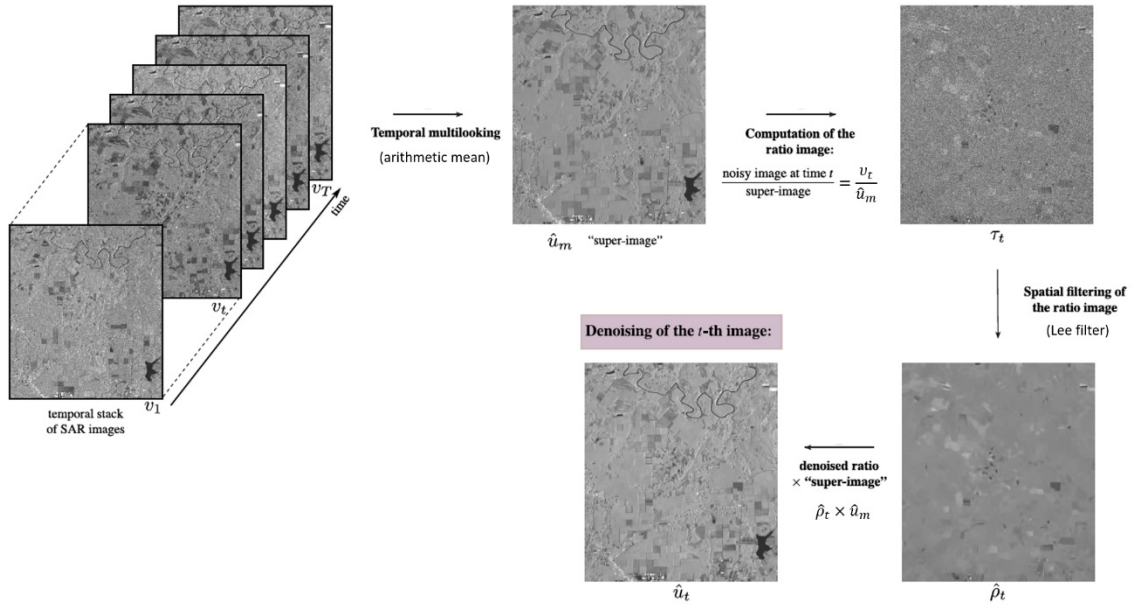


Figure 6. Summary of multitemporal despeckle method applied on SAR time series

The temporal averaging (also called temporal multi-looking) of SAR time series generates an image with reduced speckle and a preserved spatial resolution, that has been identified as 'super-image'.

Given a time series of spatially registered and radiometrically calibrated SAR images T amplitude values $\{\upsilon_1, \upsilon_2, \dots, \upsilon_t, \dots, \upsilon_T\}$, the super image has been computed the arithmetic mean for its good properties [23], in particular in terms of 18odelling the statistics of the super-image [23]. Hence, the arithmetic mean is calculated at pixel p by:

$$\hat{u}_m(p) = \frac{1}{T}\sum_{t=1}^{T} \upsilon_t(p) \qquad t \in [1, T].$$

After temporal averaging the second step consists in using the super-image to form the ratio image τ_t between the image υ_t at time t and the super image $\hat{u}_m$, at each spatial location p:

$$\tau_t(p) = \frac{\upsilon_t(p)}{\hat{u}_m(p)}$$

It contains the residual speckle noise between the two images, and the radiometric shifts when changes occur. When the length of the time series increases and in the absence of change, the super image $\hat{u}_m$ converges to u_t, the reflectivity of the scene (the signal of interest). The ratio image τ_t then tends to pure speckle (i.e., a collection of independent identically distributed random variables with unitary mean and the same number of looks as the original image). In contrast, when changes occur in the time series, these changes impact the super image which then differs from the reflectivity u_t of the image at time t. Processing the ratio image τ_t is necessary to correctly recover the reflectivity u_t. Anyway, the ratio image still needs of speckle reduction methodologies since both the noisy image υ_t and the super-image $\hat{u}_m$ suffer from speckle (although speckle in the super-image is strongly reduced). The use of this additional spatial filtering step to form the ratio image seems beneficial in terms of restoration quality: the obtained image is smoother.

Finally, in the latter step the filtered image is recovered by multiplying the denoised ratio image with the original super image $\hat{u}_m$. The estimated image $\hat{u}_t$ at location p is given by:

$$\hat{u}_t(p) = \hat{u}_m(p) \cdot \hat{\rho}_t(p)$$

Based on the processing of SAR stack corrupted by speckle noise, the approach has showed the potential to better preserve structures in multi-temporal SAR images while efficiently removing speckle. A classic application of this approach has been well reported in Figure 6Multiscale merging.

In case we do not have enough IW S-1 data to cover the area of interest, it may become necessary to merge data at different spatial resolution. This is achieved by a multi-scale SAR merging following [24].

In this work, among all available option, we selected the Discrete Wavelet Transform and Histogram Matching framework (DWT/HM) because among all other filters, the DWT is the most common way for dealing the multiscale signal representation at pixel-level in simply and effective manner, due its ease implementation and the low computational cost [25].

Firstly, let us model the vector $\boldsymbol{X} = [\boldsymbol{X}_1, \boldsymbol{X}_2, \dots, \boldsymbol{X}_k]$ that represents a multiscale SAR dataset, with k satellite imagery that having different resolution levels. In particular, the elements are arranged in ascending order in terms of resolution level, where the data with subscript 1 has finest resolution while the k-th element denotes the product with coarsest resolution.

4.3.3 Discrete Wavelet Transform and Histogram Matching framework (DWT/HM)

Generally, the wavelet transform decomposes a signal into a set of basis so-called wavelets. The wavelet representation provides a way for analyzing signals in both time and frequency domains. This makes it ideal for representing non-stationary signals, to which most real-world signals belong. The DWT transforms a discrete time signal to a discrete wavelet representation [26]. This procedure carries out a lossy compression, since components of signal that are known to be redundant, are discarded. The classical DWT is implemented by considering two filters: low-pass (LPF) and high-pass (HPF) filters. The DWT method is implemented also in bi-dimensional (2D) case. In fact, in image processing, the image $\boldsymbol{X}_m$, with $m = 1,2,..,k$, is filtered by means a high-pass and a low-pass filter combination. After the filtering, the outputs are all downscaled by a factor of two. In figure 2, a simple diagram that report the basic architecture of the DWT procedure is shown.

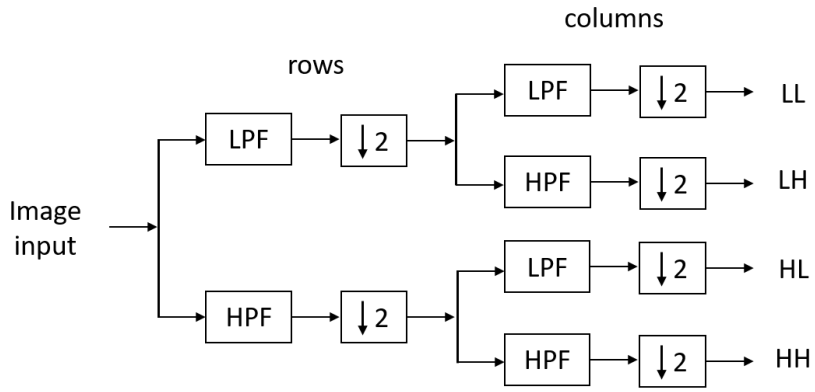


Figure 7. Block scheme of 2D DWT algorithm: the 1-Level 2D analysis DWT image decomposition process.

The original image is decomposed into four sub-band images, it deals with row and column directions separately. First, the HPF and the LPF are exploited for each row data, and then are down-sampled by two to get high- and low-frequency components of the row. Next, the high- and the low-pass filters are applied again for each high- and low-frequency components of the column, and down-sampled by two. By way of the above processing, the four sub-band images are generated: $\boldsymbol{HH}_{(m+1)}$, $\boldsymbol{HL}_{(m+1)}$, $\boldsymbol{LH}_{(m+1)}$, and $\boldsymbol{LL}_{(m+1)}$, with a resolution level equals to $(m+1)$ due to the down-sampling (note that we used the round brackets for emphasize that we were passed from m to (m+1) resolution by applying the DWT approach). Each sub-band image has its own feature, such as the low-frequency information is preserved in the $\boldsymbol{LL}_{(m+1)}$-band (named *context image* also) and the high-frequency information is almost preserved in the $\boldsymbol{HH}_{(m+1)}$-, $\boldsymbol{HL}_{(m+1)}$-, and $\boldsymbol{LH}_{(m+1)}$-bands.

The $\boldsymbol{LL}_{(m+1)}$-subband image can be further decomposed in the same way (in recursive manner) for the second level sub-band image. By using 2D DWT, an image can be decomposed into any level sub-band images, as shown in Figure 8.

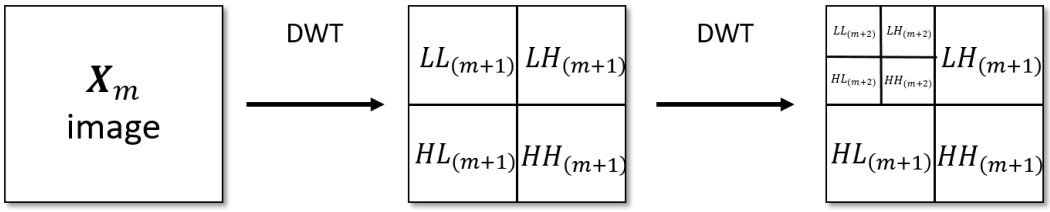


Figure 8. Diagrams of DWT image decomposition: the 2-Level 2D analysis DWT subband.

For carrying out the data fusion between two images having different resolution levels, $\boldsymbol{X}_i$ and $\boldsymbol{X}_j$, with $i < j$, for example, we could be use the DWT in addition to the standard Histogram Matching (HM) method. The HM is a transformation used for generating an image that is harmonized from a statistical point of view, since its probability distribution function matches a specified histogram [27].

In fact, the DWT is recursively applied on the finer image $\boldsymbol{X}_i$ until achieving the desired level j, and the $\boldsymbol{LL}_{(j)}$ subimage is hence calculated. Then, the HM is applied on the coarser image $\boldsymbol{X}_j$ in order to obtain the HM image version, ${\boldsymbol{X}_j}^{HM}$(calculated respect the target image $\boldsymbol{LL}_{(j)}$) for resampling it on a common (much finer) grid. The next step provides to substitute $\boldsymbol{LL}_{(j)}$ with the derived image given by the mean $\boldsymbol{LL}_{(j)}$ and ${\boldsymbol{X}_j}^{HM}$, i.e. $(\boldsymbol{LL}_{(j)} + {\boldsymbol{X}_j}^{HM})/2$. Finally, we derive the data fusion result at $i - th$ resolution level by applying the inverse DWT, i.e. the reconstruction process, opposite to the decomposition one, is formed by synthesis filters and up-samplers [28] going back until the finer scale.

The method can be implemented for the whole multiscale dataset $\boldsymbol{X}$, starting from two images with coarsest resolution. The DWT procedure is hence iteratively repeated by using the fused result and the image with the finest resolution (among all those still unused for the fusion) as input. The output is a unique final fused image.

As a backup, in the procedure has been implemented a second method for multi-scale SAR merging, the Multiscale Kalman Filter (MKF) can be considered. MKF is a pyramidal approach where the spatial resolution is assumed as an independent variable as the time. As described in [29], the MKF algorithm can be applied following two different modes respect the DWT one, since the fusion data with different scales might be carrying out starting both from finer resolution data to coarser resolution (upward step).

5 Multi-sensor geolocation

Given the outputs of the optical and SAR processing chains, a further pre-processing stage, prior to their joint use for land cover mapping, is generally necessary to make them spatially aligned. In general terms, the process of aligning different sets of image data and of referencing them into a common coordinate system (Figure 9) is named image registration. Input data for registration may be multiple photographs, data from different sensors, times, or viewpoints [30]. One image is taken as the “reference image”, and all other images are registered to the reference image are called “sensed (or input) images”. Besides remote sensing, it is used in computer vision, medical imaging, military automatic target recognition, etc. Registration is necessary in order to be able to compare or integrate the data corresponding to the same scene but obtained from different measurements. Here, the focus is put on multi-sensor geolocation, which corresponds to the case where image registration is applied to data gathered by different sensors, namely optical and SAR sensors in the CCI+ HRLC pipeline.

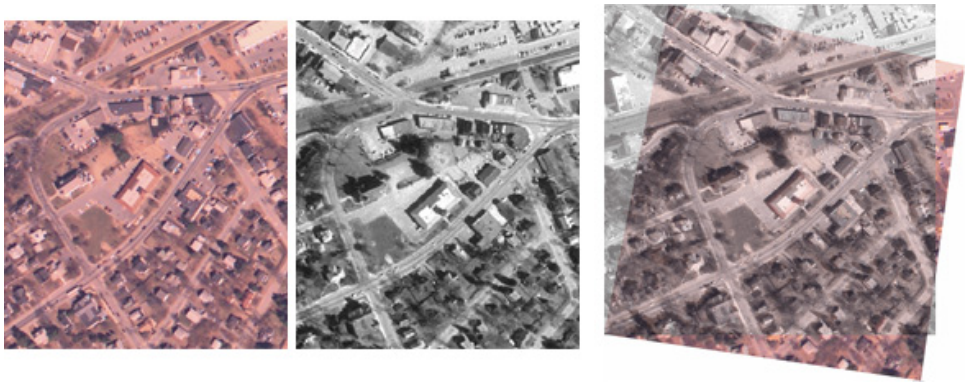


Figure 9: Image registration example from aerial photos.

By definition, multi-sensor geolocation enables the integration of complementary information from different sensors. A registration method is broadly composed of different elements, i.e.: (i) the geometric transformation used to warp the input image; (ii) the similarity measure used to compare the reference and input images during the registration process; and (iii) the optimization strategy used to minimize or maximize the similarity measure, depending on the semantic of the metric.

Figure 10: Building blocks of multi-sensor geolocation.

The following subsections cover each one of such aspects, focusing on the choices related to the processing steps of the multi-sensor geolocation block in the CCI+ HRLC pipeline. Hence, Section 5.1 describes all the geometric transformations utilized within multi-sensor geolocation. Section 5.2 details the similarity measures, while Section 5.3 deals with the minimization strategies. Finally, Section 5.4 introduces the possibilities of using deep learning methods for geolocation purposes.

5.1 Geometric Transformations

Image registration assumes a consistent geometric transformation between the sensed and reference images. Suppose that the sensed (or input) image $In(x, y)$ is defined over an (x, y) coordinate system, while the reference image $Ref(X, Y)$ is defined over an (X, Y) coordinate system. The goal of image registration is to find the transformation $T: (X, Y) \mapsto (x, y)$ that modifies the input image so as to be referenced in the same coordinate system as the reference image:

$$Ref(X, Y) \simeq In\{T(X, Y)\}$$

Within the CCI+ HRLC pipeline, the focus is put on global transformations, i.e., transformations operating on the entire image or on an image patch of non-negligible size. A rather general case is represented by the affine transformation. Affine transformations are identified by a vector of six parameters, i.e. translation over the x axis T_x, translation over the y axis T_y, rotation angle θ, scale factor on the x axis s_x, scale factor on the y axis s_y, and shear angle ϕ_{sh}. Particular cases of affine transformations are represented by rotation-scale-translation (RST) transformations (similarity transformations), where the shear angle is zero ($\phi_{sh} = 0$) and the scale factor is equal in the two dimensions ($s_x = s_y = s$); rigid transformations, a particular case of similarity transformation where there is no effect on the scale ($s = 1$); and shift transformations, characterized by a simple translation of the image ($\theta = 0$) [31].

In more details, the transformation $T: (X, Y) \mapsto (x, y)$ can be formulated as in the following equations, for each of the aforementioned cases, starting from the simpler shift transformation and moving to the more complex affine transformation. It is worth noting that, with the following convention, all the rotations are intended to be counter-clockwise.

- Shift transformations
$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -T_x \\ 0 & 1 & -T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
- Rigid transformations
$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -T_x \\ 0 & 1 & -T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
- Similarity transformations (RST)
$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -T_x \\ 0 & 1 & -T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$
- Affine transformation
$$\begin{bmatrix} X \\ Y \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & -T_x \\ 0 & 1 & -T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin(\theta + \phi_{sh}) & 0 \\ \sin\theta & \cos(\theta + \phi_{sh}) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Together with a given geometric transformation, to complete the mapping between the reference and the input image $Ref(X, Y) \simeq In\{T(X, Y)\}$, it is also necessary to define a resampling strategy [31]. In the case of the CCI+ HRLC processing chain, the chosen resampling strategy is the nearest neighbour (NN) interpolation. More in detail, considering again the transformation $T: (X, Y) \mapsto (x, y)$, the value of the output pixel (X, Y) is chosen equal to that of the input pixel (x', y') whose location is closest to the reverse sampled position (x, y) (whose components are generally non-integer). The advantage of nearest neighbour resampling is that the output image only contains intensity values present in the original image.

5.2 Similarity Measures

Image registration is aimed at aligning two images, the input and the reference. The reference image is fixed, and the input image is transformed to match the reference image. The matching strategies may be feature-based (e.g., speeded-up robust features (SURF) [32], Harris corner detection [33], maximally stable extremal regions (MSER) [34], etc.), area-based (cross-correlation, information theoretic measures [35], etc.), or hybrid. Within the CCI+ HRLC pipeline we focus on area-based methods and in particular on mutual information [36], [37], [38] and cross correlation. Additional details on such strategy are reported in Section 5.2.3.

5.2.1 Area-based Methods

Area-based strategies [39], [40] rely on similarity and information-theoretic measures. In general, area-based methods are computationally heavier than the feature-based strategies because of the necessity to compute the similarity measure taking into consideration the whole image or generally large image regions. Nevertheless, the accuracy achievable by such techniques is generally higher than that achieved by feature-based methods [39].

As anticipated above, within the HRLC pipeline, two similarity measures are taken into consideration. On one hand, mutual information, an information-theoretic measure based on comparing local intensity distributions rather than individual pixel values, is particularly suited for multi-sensor geolocation where the images to be registered have different statistics and acquisition geometries. The main drawback is that, even though it is more robust and less sensitive to noise than the correlation-based measures, statistical distributions are heavier to be estimated on large-scale imagery and, in the end, they result in long computation time. On the other hand, similarity measures like the cross correlation are faster to compute but less suited for multi-sensor data, as they are based on the pixel-wise comparisons of intensity values.

5.2.2 Feature-based Methods

Feature-based methods are generally faster but less accurate than area-based methods, and the accuracy of the registration result depends on the accuracy of the feature extraction method that is being used. There exist different strategies for the extraction of informative features. In particular, feature-point registration algorithms [39] extract a set of distinctive and highly informative individual points from both images, and then find the geometric transformation that matches them. Feature points are named in different ways, including control points, tie-points, and landmarks.

Well-known approaches in this area are those based on scale-invariant feature transforms (SIFT) [41], speeded-up robust features (SURF) [32], maximally stable extremal regions (MSER) [34], and Harris point detectors [33]. Other features of interest may be curvilinear and could be extracted by using edge detection algorithms [39], generalized Hough transforms [42], or stochastic geometry (e.g., marked point processes) methods [43].

Within the CCI+ HRLC processing chain, the possible role of feature-based methods within the multi-sensor geolocation stage of the pipeline might possibly be involved in the second Round Robin exercise.

5.2.3 The CCI+ HRLC strategy

Within the CCI+ HRLC processing chain, where the reference and the input images are the optical and the SAR images, the choice is the usage of area-based methods based on the estimation of the mutual information between the two images.

Another common possibility is the usage of cross-correlation as similarity measure; however, such option is particularly critical in the multi-sensor case of the CCI+ HRLC chain. The computation of the cross-correlation, especially using the strategy based on the fast Fourier transform (FFT) [44], is usually faster and hence more convenient in an iterative process like image registration. However, the different statistics of the optical and SAR images, together with the different acquisition geometries, prevent the usage of cross-correlation within the CCI+ HRLC pipeline. Nevertheless, the usage of cross-correlation and the fast computation through FFT will be dealt with in Section 5.4 and Section 5.4.1, where the focus will be put on the possibility of using generative adversarial networks (GANs) to perform domain adaptation as a pre-processing step of registration.

With respect to mutual information $\mathrm{MI}(Ref, In)$ between the reference and the input images, let $In(\cdot)$ and $Ref(\cdot)$ indicate the input and reference images (which are both assumed composed of $M \times N$ pixels), respectively. Let also $p_{Ref,In}$ be their joint distribution, and p_{Ref} and p_{In} be their marginal distributions. The mutual information is thus computed according to:

$$\mathrm{MI}(Ref, In) = \sum_{r} \sum_{i} p_{Ref,In}(r, i) \log \frac{p_{Ref,In}(r, i)}{p_{Ref}(r)\, p_{In}(i)}$$

There are different methods to compute such quantity. Within the CCI+ HRLC the mutual information is estimated by approximating the probability distributions through the normalized histograms. Another option, which is computationally heavier, is to estimate such distributions using kernel-based methods like Parzen window density estimation [45]. Unfortunately, due to the large scale of the project and the iterative optimization process, using heavy estimators is unfeasible because of the registration process requiring multiple sequential estimations.

5.3 Optimization Strategies

As anticipated in the introduction to this chapter, the registration task is viewed as the combination of the following sub-processes [46]:

1. Selecting a transformation model and a resampling strategy.
2. Selecting a similarity metric to decide if a transformed input image closely matches the reference image.
3. Selecting a search strategy, which is used to match the images based on maximizing or minimizing the similarity metric.

We already discussed points 1 and 2 in the Sections above; here the focus is put on the optimization strategy that has been chosen for the CCI+ HRLC multi-sensor geolocation step. The optimization strategies that are integrated in the pipeline are the unconstrained Powell's algorithm and constrained optimization by linear approximation (COBYLA) method. On one hand, the unconstrained Powell's algorithm uses Powell's formulation of an approximate conjugate direction method. The objective function does not need to be differentiable, and no derivatives are required (differently from the standard conjugate gradient algorithm). The method minimizes the function using a bi-directional search along a set of search vectors [47]. Moreover, the bi-directional line search is done by Golden-section search and Brent's method [48].

On the other hand, COBYLA addresses constrained optimization by a linear approximation. It works by iteratively approximating the actual constrained optimization problem with linear programming problems. At each iteration, the resulting linear programming problem is solved to obtain a candidate for the optimal solution. The candidate solution is evaluated using the original objective and constraint functions, yielding a new data point in the optimization space. This information is used to improve the approximating linear programming problem used for the next iteration of the algorithm. When no improvement is possible, the step size is reduced, refining the search. When the step size becomes sufficiently small, the algorithm stops [49].

It is worth noting that the Powell's algorithm performs well in case of transformations where the input image and the reference image are not "very distant," i.e., when the optimal solution is in the neighborhood of the starting point. Conversely, the COBYLA algorithm allows the user to choose the starting search radius. The tuning of such parameter allows the registration process to explore regions of the search space that a simple conjugate-gradient method would never reach. Within the project pipeline, a modified COBYLA method is able to perform a grid search for the radius parameter and choose the one that allows best fitting the two images.

5.4 Multi-sensor Geolocation using Deep Learning Architectures

Another approach that is taken into consideration within the CCI+ HRLC processing chain is the usage of deep learning architectures [50] for multi-sensor geolocation. Deep learning solutions for the registration of multi-sensor data is becoming of great interest for the remote sensing community.

In the context of the CCI+ HRLC processing chain, a deep learning solution will be investigated. Such strategy uses auto encoders [50] and adversarial networks [51] with the goal of developing a domain adaptation [52] method and transform optical images into SAR-like data or vice versa. With such a domain adaptation, the application of the aforementioned area-based techniques is significantly favoured because the optical and SAR data are brought together in a common homogeneous domain in which they are more directly comparable. The adversarial network considered here will be based on the interconnection of convolutional neural networks (CNNs), which have been proven highly effective in the application to the semantic segmentation of remote sensing images for land cover mapping purposes [53].

In particular, as anticipated before, the cross-correlation similarity measure, together with the fast computation through FFT, will take the place of the heavier-to-estimate mutual information. Therefore, in the following section, the details of such computation through the fast Fourier transform are presented.

5.4.1 Cross-correlation via Fast Fourier Transform

Let again $In(\cdot)$ and $Ref(\cdot)$ indicate the input and reference images (which are both assumed composed of $M \times N$ pixels), their cross correlation can be computed according to:

$$\mathrm{CC}(x,y) = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} In(m,n)\, Ref(m-x, n-y)$$

There exists a formulation of such quantity computed using the fast Fourier transform [44]. Such process takes advantage of the relation between the convolution operation in the spatial or time domain and the product operation in the frequency domain. Let $\mathcal{F}(\cdot)$ denote the Fourier transform operator and let f and g be two signals defined in the spatial or time domain. It is straightforward to write the cross-correlation in terms of a

convolution operator, which allows taking benefit of the computational efficiency of the FFT [44] and derive the cross-correlation by combining transformation, products, and inverse transformations (up to introducing the appropriate zero padding):

$$\mathcal{F}(f * g) = \mathcal{F}(f) \cdot \mathcal{F}(g) \quad \rightarrow \quad f * g = \mathcal{F}^{-1}\big(\mathcal{F}(f) \cdot \mathcal{F}(g)\big)$$

In more details, to compute the cross-correlation between two images it is necessary to: (i) compute the FFT of each image (up to zero padding) to pass from the spatial domain to the frequency domain; (ii) compute the complex conjugate of one of the two resulting signals in the frequency domain because of the mirroring operation performed during convolution and not during correlation; (iii) multiply the images in the frequency domain; and (iv) compute the inverse FFT transform of the product to obtain the cross-correlation of the two images in the spatial domain. The flowchart of such computation is provided in Figure 11.

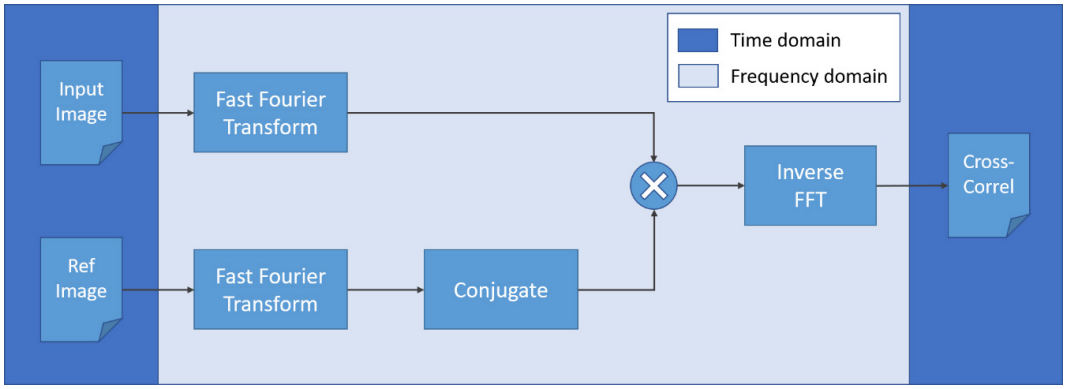


Figure 11: Computation of the cross-correlation via FFT.

6 Classification algorithms for HR land cover

Classification is the process that converts multitemporal imagery (both optical and SAR) into land cover maps, see the workflow in Figure 12. The selected classification algorithm must achieve the best trade-off between classification accuracy and computational burden due to the need of processing a huge amount of data. By analysing the recent literature, the team have identified and tested different successful core approaches: (1) Support Vector Machine (SVM) classifier, (2) Random Forest (RF), (3) Maximum Likelihood (ML), (4) Multilayer Perceptron (MLP) Artificial Neural Network (ANN). According to the classification results obtained by the above-mentioned classification algorithms, the team selected the SVM as classifier (see PVASR v1.0). However, the team is also evaluating the possibility of using sophisticated deep learning technique such as Long Short Term Memory classifier [19] or Convolutional Neural Networks (CNN) trained on multitemporal data, to extensively exploit the spectral information provided by the long time series of Sentinel 2 images (see ADP v1.0).

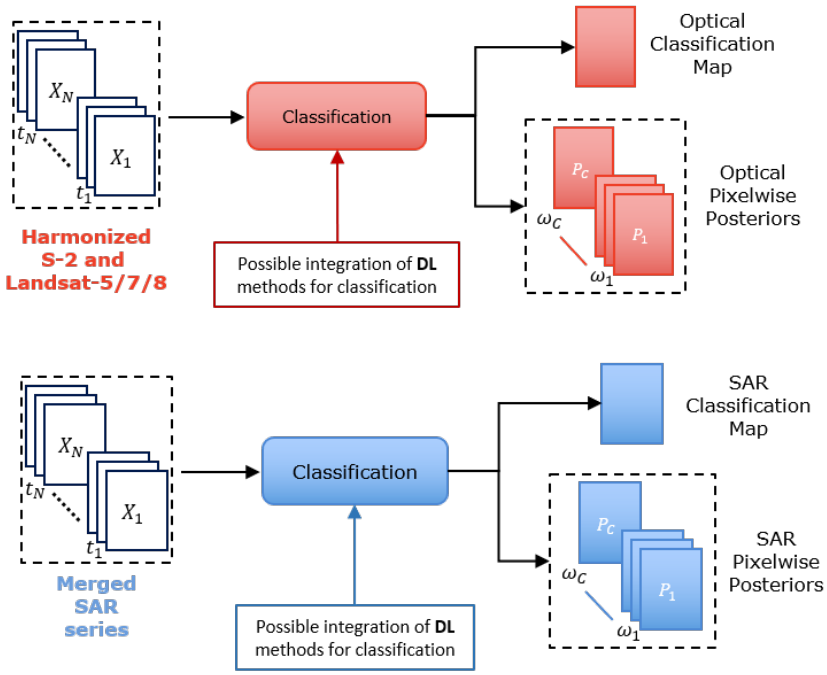


Figure 12. Workflow of the classification process for optical and SAR time series of images.

6.1 Random Forest classifier

The Random Forest (RF) algorithm is a supervised classification method that creates a set of decision trees from randomly selected subset of training set. It then aggregates the votes from different decision trees to decide the final class of the test object. The bagging method developed in [54] is used for each feature/feature combination selected. The idea is to use multiple versions of a predictor or classifier to make an ultimate decision by taking a plurality vote among the predictors. Hence, any pixel is classified by taking the most popular voted class from all the tree predictors in the forest. In bagging, it has been proved that the accuracy increases with increasing of number of trees, i.e. the predictors number [55].

The effectiveness of decision tree classifier for land cover classification has been assessed in [56].

The training algorithm for random forests starts with a training set $X = x_1, \dots, x_n$ with responses $Y = y_1, \dots, y_n$ and it iterates B times over repeated random sampling with replacement of the training set while fitting trees to these samples. For each $b = 1, \dots, B$, let us denote these samples (from X and Y) as X_b and Y_b, then a classification tree f_b is trained over these samples. The number of features used at each node to generate a tree and the number of trees to be grown are two user-defined parameters required to generate a random forest classifier.

At each node, only selected features are searched for the best split. Thus, the random forest classifier consists of B trees, where B is the number of trees to be grown which can be any value defined by the user. To classify a new data set, each case of the data sets is passed down to each of the B trees. The forest chooses a class having the most out of B votes, for that case. Finding the optimal number of predictors to generate will yield the highest accuracy.

This bagging procedure leads to better model performance because it decreases the variance of the model, without increasing the bias. This means that while the predictions of a single tree are highly sensitive to noise in its training set, the average of many trees is not, as long as the trees are not correlated. Simply training many trees on a single training set would give strongly correlated trees (or even the same tree many times, if the training algorithm is deterministic); bagging sampling is a way of de-correlating the trees by showing them different training sets. Moreover, an estimate of the uncertainty of the prediction can be made as the standard deviation of the predictions from all the individual regression trees on the sample x as

$$\sigma = \sqrt{\frac{\sum_{b=1}^{B}\left(f_b(x) - \hat{f}\right)^2}{B-1}}$$

where prediction $\hat{f}$ is obtained as the sample mean over the f_b.

6.2 Support Vector Machine

As a classifier, the Support Vector Machine (SVM) is one of the most effective methods in pattern and texture classification to the land cover mapping [57]. Its fundamental idea is that the feature of input space is mapped into a high-dimensional feature space through nonlinear transformation. The nonlinear transformation is implemented by defining proper kernel function. SVM has two important features. Firstly, the upper bound on the generalization error does not depend on the dimension of the space. Secondly, the error bound is minimized by maximizing the margin, that is, the minimal distance between the hyperplane and the closest data points [58], [59]. SVMs are particularly appealing in remote sensing field due to their ability to successfully handle small training datasets, often producing higher classification accuracy than traditional methods, as well as to be the best algorithm when classes are separable [59]. In contrast, for larger dataset, it requires a large amount of time to process.

SVM implements a classification strategy that exploits a margin-based "geometrical" criterion rather than a purely "statistical" criterion. In other words, SVM does not require an estimation of the statistical distributions of classes to carry out the classification task. Instead, the classification model exploits the concept of margin maximization. The main properties that make SVM particularly attractive in the considered application are the following:

- their intrinsic effectiveness with respect to traditional classifiers thanks to the structural risk minimization principle, which results in high classification accuracies and very good generalization capabilities;
- the possibility to exploit the kernel trick to solve non-linear separable classification problems by projecting the data into a high dimensional feature space and separating the data with a simple linear function;
- the convexity of the objective function used in the learning of the classifier, which results in the possibility to solve the learning process according to linearly constrained quadratic programming (QP) characterized from a unique solution (i.e., the system cannot fall into sub-optimal solutions associated with local minima);
- the possibility of representing the convex optimization problem in a dual formulation, where only non-zero Lagrange multipliers are necessary for defining the separation hyperplane (which is a very important advantage in the case of large datasets). This is related to property of sparseness of the solution.

Let us assume that a training set is given $D = \{(x_i, y_i)\}_{i=1}^{N}$, where $x_i = (x_i^1, \ldots, x_i^j)$ is the i-th primitive feature and $\mathcal{Y} = \{y_i\}_{i=1}^{N}$ is the corresponding set of labels. Accordingly, let us assume that $y_i \in \{+1; -1\}$ is the binary label of the sample x_i. The goal of the binary SVM is to divide the d-dimensional feature space in two subspaces, one for each class, through a separating hyperplane $H: y = \langle w \cdot x \rangle + b = 0$. The final decision rule used to find the membership of a test sample is based on the sign of the discrimination function $f(x) = \langle w \cdot x \rangle + b$ associated to the hyperplane. Therefore, a generic sample x will be labelled according to the following rule: $y = sgn\, f(x)$.

The training of an SVM consists in finding the position of the hyperplane H, estimating the values of the vector w and the scalar b, according to the solution of an optimization problem. From a geometrical point of view, w is a vector perpendicular to the hyperplane H and thus defines its orientation. The distance of the H to the origin is $b/\|\mathbf{w}\|$, while the distance of a sample x to the hyperplane is $f(\mathbf{x})/\|\mathbf{w}\|$. Let us define the functional margin $F = \min\{y_i f(\mathbf{x}_i)\}$, $i = 1, \ldots, N$ and the geometric margin $G = F/\|\mathbf{w}\|$. The geometric margin represents the minimum Euclidean distance between the available training samples and the hyperplane.
In the case of a linearly separable problems, the learning of an SVM can be performed with the maximal margin algorithm, which consists in finding the hyperplane H that maximizes the geometric margin G. However, the

maximum margin-training algorithm cannot be used in case the available training samples are not linearly separable because of noisy samples and outliers. In these cases, the soft margin algorithm is used in order to handle nonlinear separable data. This is done by defining the so-called slack variables as:

$$\xi\left[(x_i, y_i), (w, b)\right] = \xi_i = \max\left[0, 1 - y_i(\langle w \cdot x_i \rangle + b)\right]$$

Slack variables allow one to control the penalty associated with misclassified samples. In this way the learning algorithm is robust to both noise and outliers present in the training set, thus resulting in high generalization capability. The optimization problem can be formulated as follows:

$$\begin{cases} \min\limits_{\mathbf{w},b}\left\{\frac{1}{2}\|\mathbf{w}\|^2 + C\sum\limits_{i=1}^{N}\xi_i\right\} \\ y_i(\langle \mathbf{w}\cdot\mathbf{x}_i\rangle + b) \geq 1-\xi_i,\ \ \xi_i \geq 0,\ \ \forall i = 1,\ldots,N \end{cases}$$

where $C \geq 0$ is the regularization parameter that allows one to control the penalty associated to errors (if $C = \infty$ we come back to the maximal margin algorithm), and thus to control the trade-off between the number of allowed mislabelled training samples and the width of the margin. If the value of *C* is too small, many errors are permitted and the resulting discriminant function will poorly fit with the data; on the opposite, if *C* is too large, the classifier may overfit the data instances, thus resulting in low generalization ability. A precise definition of the value of the *C* parameter is crucial for the accuracy that can be obtained in the classification step and should be derived through an accurate model selection phase. Similarly to the case of the maximal margin algorithm, the optimization problem can be rewritten in an equivalent dual form:

$$\begin{cases} \max\limits_{\alpha}: \left\{\sum\limits_{i=1}^{N}\alpha_i - \frac{1}{2}\sum\limits_{i=1}^{N}\sum\limits_{j=1}^{N} y_i y_j \alpha_i \alpha_j \left\langle \mathbf{x}_i, \mathbf{x}_j \right\rangle\right\} \\ \sum_{i=1}^{N} y_i \alpha_i = 0,\ \ 0 \leq \alpha_i \leq C,\ \ \ 1 \leq i \leq N \end{cases}$$

Because of the constraint introduced by the multipliers $\{\alpha_i\}_{i=1}^{N}$ that for the soft margin algorithm are bounded by the parameter *C*, the problem is also known as box constrained problem. The Karush–Kuhn–Tucker (KKT) complementarity conditions provide useful information about the structure of the solution. They state that the optimal solution should satisfy:

$$\begin{cases} \alpha_i\left[y_i(\langle \mathbf{w}\cdot\mathbf{x}_i\rangle + b) - 1 + \xi_i\right] = 0, & i = 1,\ldots,l \\ \xi_i(\alpha_i - C) = 0, & i = 1,\ldots,l \end{cases}$$

Varying the values of the multipliers $\{\alpha_i\}_{i=1}^{N}$ three cases can be distinguished:

$$\begin{cases} \alpha_i = 0 & \Rightarrow y_i f(\mathbf{x}_i) \geq 1 \\ 0 < \alpha_i < C_i & \Rightarrow y_i f(\mathbf{x}_i) = 1 \\ \alpha_i = C_i & \Rightarrow y_i f(\mathbf{x}_i) \leq 1 \end{cases}$$

The support vectors with multiplier $\alpha_i = C$ are called bound support vectors (BSV) and are associated to slack variables $\xi_i \geq 0$; the ones with $0 < \alpha_i < C_i$ are called non-bound support vectors (NBSV) and lie on the margin hyperplane H_1 or H_2 ($y_i f(\mathbf{x}_i) = 1$).

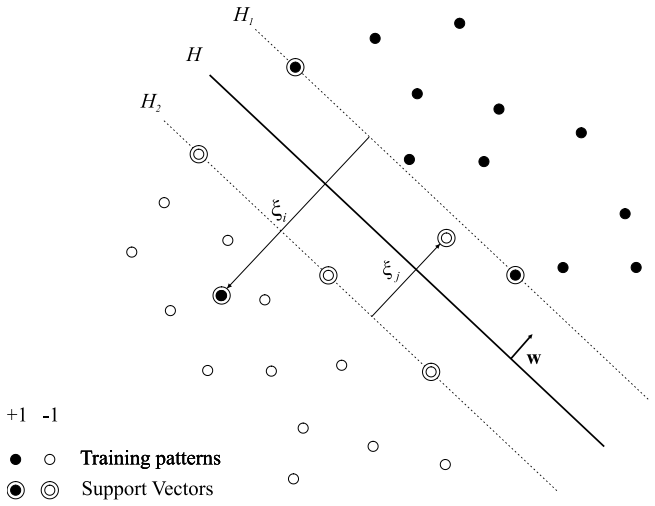


Figure 13: Qualitative example of a separating hyperplane in the case of a non-linear separable classification problem.

An important improvement to the above-described methods consists in considering nonlinear discriminant functions for separating the two information classes. This can be obtained by transforming the input data into a high dimension (Hilbert) feature space $\Phi(\mathbf{x}) \in \Re^{d'}$ ($d' > d$) where the transformed samples can be better separated by a hyperplane (Figure 14). The main problem is to explicitly choose and calculate the function $\Phi(\mathbf{x}) \in \Re^{d'}$ for each training samples. Given that the input points in dual formulation appear in the form of inner products, we can do this mapping in an implicit way by exploiting the so-called kernel trick. Kernel methods provide an elegant and effective way of dealing with this problem by replacing the inner product in the input space with a kernel function such that:

$$K(x_i, x_j) = \langle (\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \rangle \qquad i, j = 1, \ldots, N \tag{7}$$

implicitly calculating the inner product in the transformed space. The soft margin algorithm for nonlinear function can be represented by the following optimization problem:

$$\begin{cases} \max\limits_{\alpha} \left\{ \sum\limits_{i=1}^{N} \alpha_i - \frac{1}{2} \sum\limits_{i=1}^{N} \sum\limits_{j=1}^{N} y_i y_j \alpha_i \alpha_j k(\mathbf{x}_i, \mathbf{x}_j) \right\} \\ \sum_{i=1}^{N} y_i \alpha_i = 0, \;\; 0 \leq \alpha_i \leq C \text{ and } \; 1 \leq i \leq N \end{cases} \tag{8}$$

And the discrimination function becomes:

$$f(\mathbf{x}) = \sum_{i \in SV} y_i \alpha_i^* k(\mathbf{x}_i \cdot \mathbf{x}) + b \tag{9}$$

The condition for a function to be a valid kernel is given by the Mercer's theorem. The most widely used non-linear kernel functions are the following:

- homogeneous polynomial function: $k(\mathbf{x}_i, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^p, \quad p \in Z$
- inhomogeneous polynomial function: $k(\mathbf{x}_i, \mathbf{x}_j) = (c + (\mathbf{x}_i \cdot \mathbf{x}_j))^p, \quad p \in Z, c \geq 0$

- Gaussian function: $k(\mathbf{x}_i, \mathbf{x}_j) = e^{\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}}, \quad \sigma \in \Re$

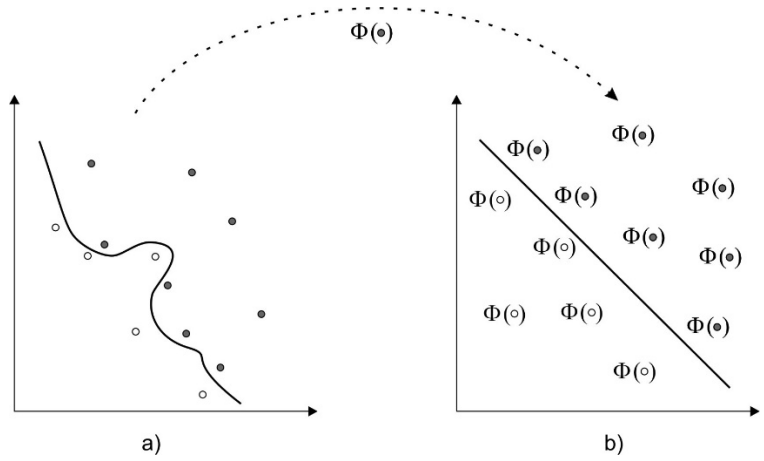


Figure 14: Transformation of the input samples by means of a kernel function into a high dimension feature space: a) Input feature space; b) kernel induced high dimensional feature space.

From an operational perspective, a possible implementation would use the Gaussian kernel since linear and polynomial kernels are less time consuming but provide in general less accuracy. The Sigma σ parameter is a positive parameter whose behavior regulates the fitting property: if its value increases the model gets overfits, while decreasing the model underfits. In our implementation, the default value for gamma is initially set equals to 1 over the number of features [60], optimal choice can be made in proper training stage.

6.3 Deep Convolutional Neural Network

Learning efficient image representations is at the core of the scene classification task of remote sensing imagery. The existing methods for solving the scene classification task, based on either feature coding approaches with low-level hand-engineered features or unsupervised feature learning, can only generate mid-level image features with limited representative ability, which essentially prevents them from achieving better performance. Recently, the deep convolutional neural networks (CNNs), which are hierarchical architectures trained on large-scale datasets, have shown astounding performance in object recognition and detection.

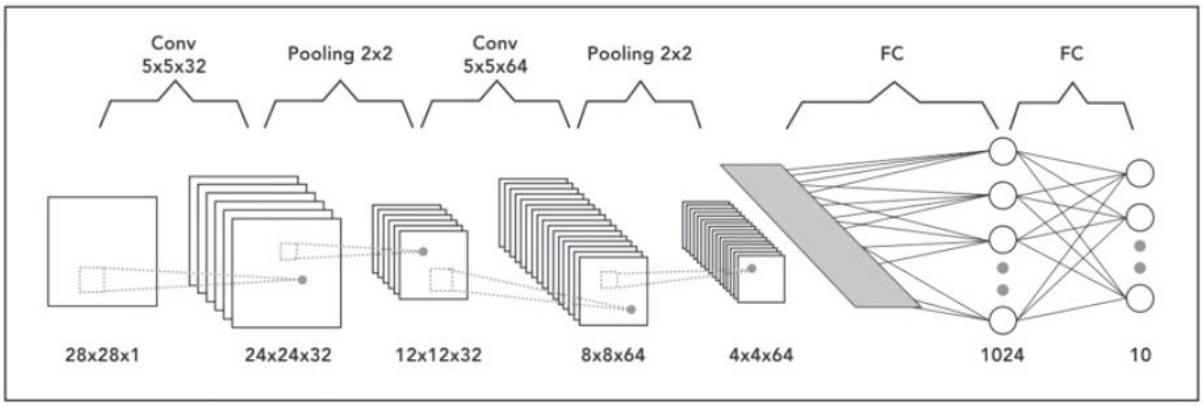


Figure 15. General architecture of a Deep Convolutional Neural Network (CNN) made of subsequent convolutional and pooling layers arranged in a cascade.

The typical architecture of a CNN is composed of multiple cascaded stages. The convolutional (conv) layers and pooling layers construct the first few stages, and a typical stage is shown in Figure 15. The convolutional layers output feature maps, each element of which is obtained by computing a dot product between the local region

(receptive field) it is connected to in the input feature maps and a set of weights (also called filters or kernels). In general, an elementwise non-linear activation function is applied to these feature maps. The pooling layers perform a down-sampling operation along the spatial dimensions of feature maps via computing the maximum on a local region. The fully-connected (FC) layers finally follow several stacked convolutional and pooling layers, and the last fully-connected layer is a Softmax layer that computes the scores for each defined class. CNNs transform the input image from original pixel values to the final class scores through the network in a feedforward manner. The parameters of CNNs (i.e., the weights in convolutional and FC layers) are trained with classic stochastic gradient descent based on the backpropagation algorithm [61].

6.3.1 CNNs on HR remote sensing imagery

In contrast to the popularity of the CNN features from FC layers, the features from intermediate convolutional layers appear to lack practical use. Although the features of FC layers capture global spatial layout information, they are still sensitive to global rotation and scaling, making them less suitable for HR images that greatly differ in orientation and scales. Therefore, feature maps produced by convolutional layers should be regarded as dense features and aggregated via other coding approaches.

By removing all FC layers, feature maps all come from the last convolutional layer. Each entity along the feature maps can be considered as a “local” feature, and the length of the feature equals the number of feature maps.

Let the $F_s^{(m)}$ be the set of dense convolutional features extracted from image X_m at scale index s. We then obtain a complete feature set by combining all $F_s^{(m)}$ at different scales, which is denoted as $F^{(m)} = [x_1, x_2, \dots, x_N] \in R^{DxN}$ consisting of N separate D-dimensional features. Three conventional feature coding methods can be considered, locality-constrained linear coding (LLC) [62], improved Fisher kernel (IFK) [63] and vector of locally aggregated descriptors (VLAD) [64] to encode the feature set $F^{(m)}$ into a global feature representation for each image X_m. Note that the LLC and VLAD encode features based on a codebook constructed via K-means, whereas the IFK encodes features with a probability density distribution described by the Gaussian mixture model (GMM).

7 Optical imagery classification

From analysis of the recent literature related to large-scale land cover mapping problems the following crucial aspects must be considered in order to achieve efficient and robust classification of optical high-resolution images [65]:

- automation for efficiency and timeliness;
- spatial continuity of the maps;
- temporal coherence between updates of the product;
- reproducibility of the results;
- support of changes of nomenclature without changing the system.

The CCI HRLC project addresses each of these points. To maximize the outcome, the following strategies must be implemented and assessed in an operational context of land cover map production at the large-scale:

- all available images acquired during the reference period are used regardless of the amount of cloud cover;
- the procedure is fully automatic without need for manual operations;
- the processing chain is implemented using a massively parallel work-flow which achieves a reduced computation time allowing timely map production and data reprocessing for ensuring continuity across reference years in the case of updating the product specification.

Project activity for optical imagery classification is so far oriented in two directions: on the one side, internal benchmarking activities are entirely devoted to the selection of the best performing classification algorithms; on

the other side, a selection of reference/ancillary data to empower and enrich the feature extraction and training processes is undergoing.

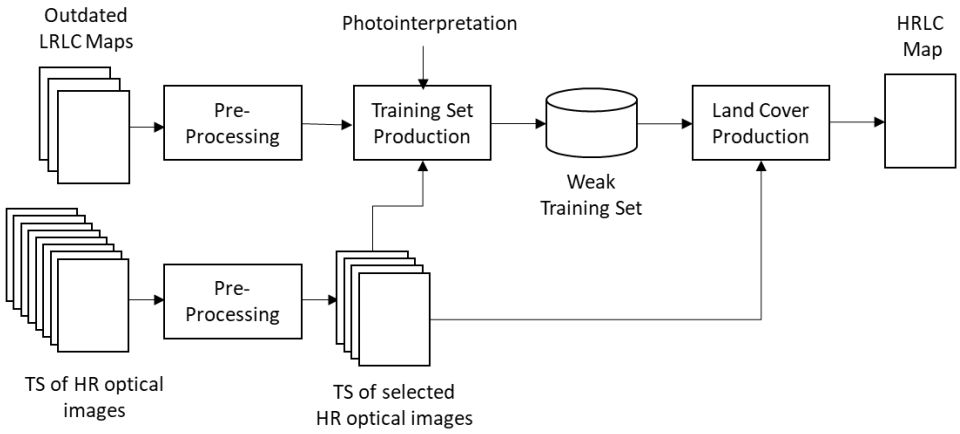


Figure 16. Optical data processing chain for the prototype production of the HR LC map obtained by classifying the time series of Sentinel 2 data.

Figure 16 depicts the optical data processing chain for the prototype production of the HR LC maps obtained by classifying the time series of Sentinel 2 data. The images are first pre-processed in order to perform the atmospheric correction and detect the clouds. Then, the best time series of images used to generate the HR static LC map is defined by automatically selected the 10 images having the lowest cloud coverage. Due to the missed availability of training data, a training set production step is performed to extract the labeled samples necessary to train the supervised classification system. Moderate resolution global LC maps are used to create a database of weak training samples. Note that at global scale the thematic products available are characterized by medium/coarse spatial resolution (e.g., 100m, 300m and 1 km), much coarser than the desired geometrical detail (10 m). The maps are analyzed and processed in an unsupervised way to detect and extract the most reliable samples which are included in the weak training set. Moreover, the team is planning to add samples by photo-interpretation to: (1) integrate the missing information on classes which require HR labeled pixels, and (2) increase the reliability of the training set.

7.1 Classification

Regardless of what specific type of classifier will be used, the general approach to classification can be summarized as depicted in Figure 17. Here, a sequence of temporally adjacent images is loaded to the classification module as input of the classification chain.

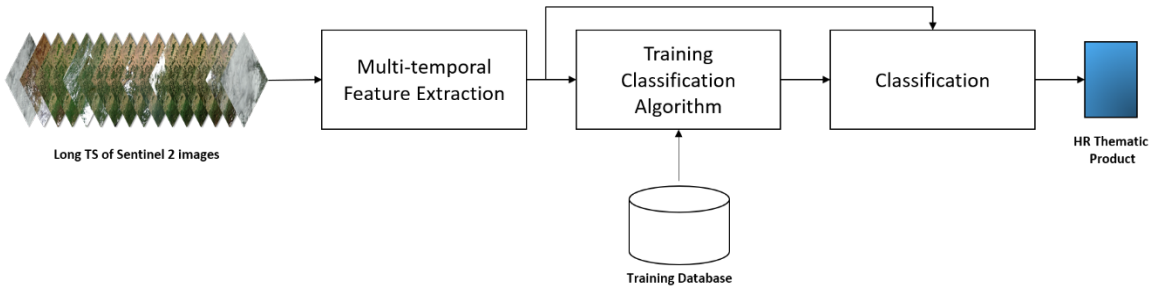


Figure 17. General approach to multitemporal multispectral image classification.

Here it follows a description of the classification module steps to be implemented for the classification:

- First, multitemporal feature extraction is performed. In this stage, a set of salient features and/or spectral bands are chosen as primitive characteristics to represent target classes. The aim of a good feature extraction phase is reaching a good discriminability of target classes in the resulting feature space, so that classification can be performed with low uncertainty. The output of the feature extraction

step is a set of pseudo-images X_t, with $t = 1, \dots, T$, each one having as many bands as the extracted features.

- The second stage is devoted to the training of the classifier. Here, labelled samples Y associated with pseudo-images X are extracted from the training database. The training step is the most intensive one. Indeed, this is where the numerical minimization of the classification mathematical problem takes place. In practice, the solution of the problem is a partition of the feature space in which every connected region represents the feature sub-region each class belongs to.
- The third and last stage is the classification. Here, each pixel in the time series of images is associated to its class according to its membership to a sub-region of the feature space as defined in the training step.

7.1.1 Training sets from medium resolution map

Reference data for the land cover classes are used for supervised classifier training. Operational land cover map production over large areas cannot rely on field campaigns because huge amounts of costly data would need to be collected, most importantly jeopardising the timeliness of the land cover map. The choice is therefore to rely on existing databases to build the reference data sets needed for the supervised classification and the final products. These data sources will contain fatal errors due to changes in the landscape (intrinsic) and the different data generation streams (systematic). Thus, reference data will undergo a preparation step to mitigate for possible inconsistencies, while other residual errors are assumed to be handled by the classifier itself. Moreover, the team is planning to enlarge the training database by photointerpretation. Although the labelling relies on the analyst's experience, these samples may sharply increase the quality of the training database.

Table 3 shows the properties of the medium resolution maps available global level. To generate a reliable training set the team exploited the 2015 Copernicus Global Land Cover (CGLC) map. This map was selected as the best candidate due to its: i) good spatial resolution (100 m), ii) detailed hierarchical legend that includes many classes that are in common with CCI+ legend, and iii) relatively recent temporal coverage (2015). Moreover, the CGLC provides the fractional cover layers together with the discrete LC map. These layers report the presence of each class estimated at pixel level in percentage. For more details please refer to DARD v2.0.

Figure 18 shows an example of fractional cover layer products provided by the CGLC maps. For each pixel of the discrete map (see Figure 18b), the percentage of the grassland and shrubland are available (see Figure 18c and Figure 18d, respectively). This ancillary information is used to increase the probability of extracting reliable labeled samples from the map. Thus, only samples associated with fractional covers higher than 60% were considered. Figure 19b and Figure 19d show examples of pixels discarded (in black) from the CGLC map for tiles 21KXT and 37PCP, respectively.

Table 3 Properties of the medium resolution maps available global level.

Land-Cover Maps Available	Temporal Coverage	Spatial Resolution	Spatial Coverage	# Classes
IGBP-DISCover	1993	1 km	Global	17
University of Maryland Land Cover	1998	1 km	Global	13
GLC-SHARE 2014	2014	1 km	Global	11
MODIS GLC	2012	500 m	Global	16
GlobCover	2010	300 m	Global	21
ESA CCI Land Cover	2015	300 m	Global	21
Copernicus Global Land Cover	2015	100 m	Global	23
GlobeLand30	2010	30 m	Global	10

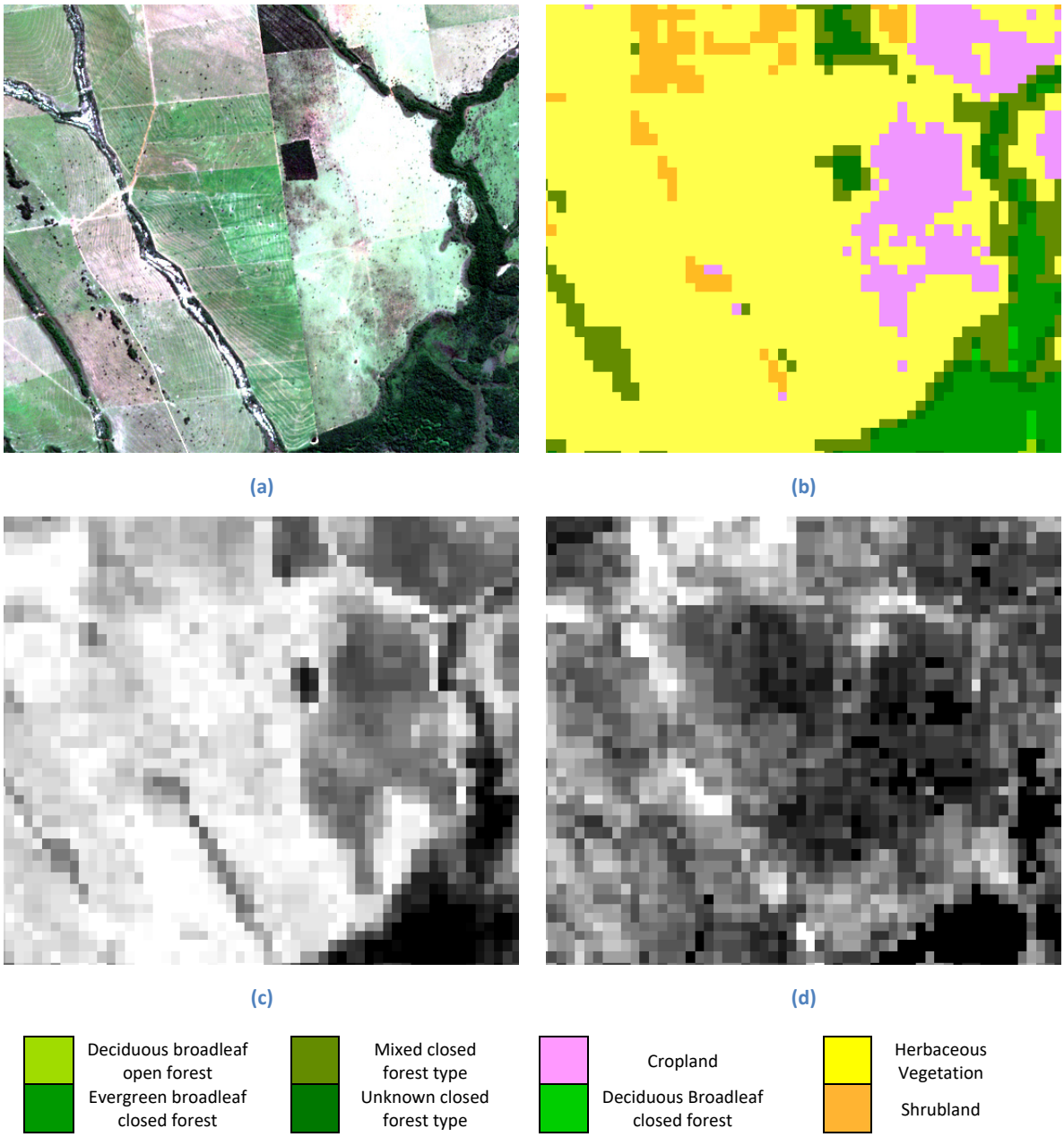


Figure 18. Example of fractional cover layer products provided by the CGLC maps: (a) the true color composition of the Sentinel 2 image acquired on the 23rd June 2018, (b) the CGLC map, (c) the Grass Fractional Cover, and (d) the Shrubland Fractional Cover.

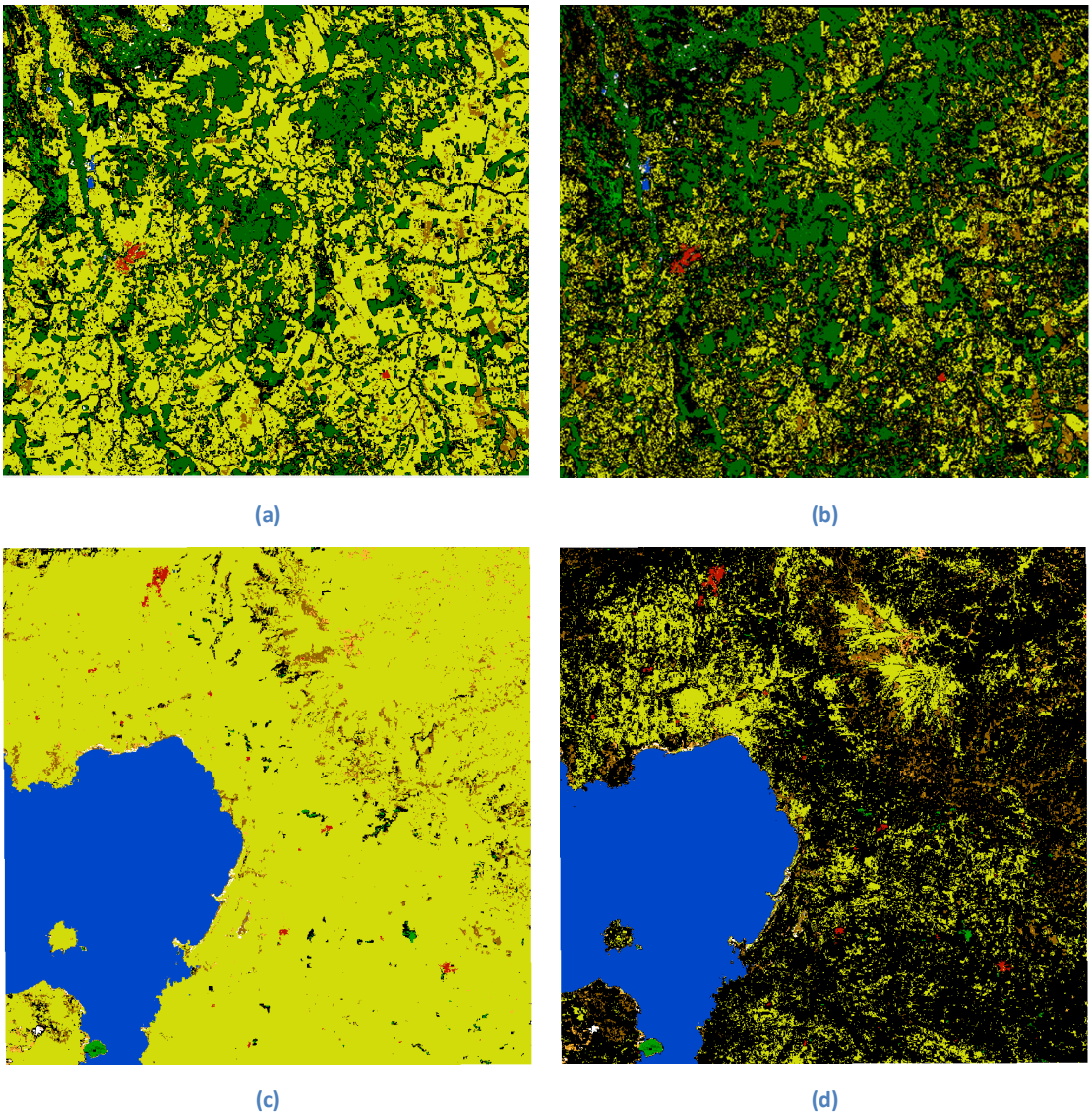


Figure 19. Example of CGLC map uncertain pixel removal based on the information provided by the fractional layers: (a) the CGLC map (tile KXT), (b) the CGLC map after uncertain pixel removal (tile KXT), (c) the CGLC map (tile 37PCP), and (d) the CGLC map after uncertain pixel removal (tile 37PCP).

To generate the training set, the map is first rescaled at 10 m spatial resolution and its legend converted to the required one according to the Land Cover Classification System (LCCS), i.e., the standard common LC language for translating and comparing existing legends. Table 4 presents the translation of the 2015 CGLC into the new HRLC legend. In order to increase the probability of selecting the most reliable samples for the tree cover classes, the proposed procedure gives the priority to the closed forest classes, which are associated with pixels having the higher presence of the forest. The open forest classes are considered only if needed, i.e., few samples of closed forest classes present in the scene.

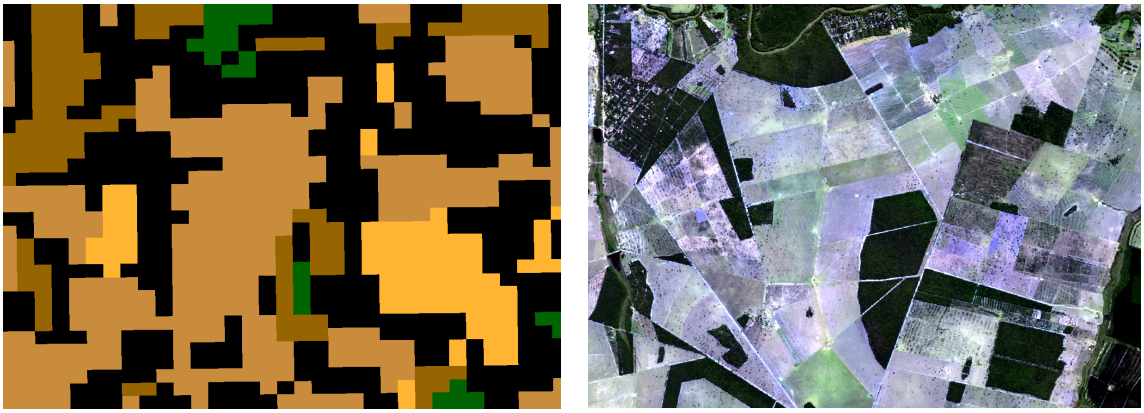
Table 4. Training Set Production: the translation of the CLGC into the desired map legend is reported.

	CCI-HRLC	CGLC (1st choice)	CGLC (2nd choice)
	Tree cover evergreen broadleaf	Closed forest, evergreen, broad leaf	Open forest, evergreen broad leaf
	Tree cover evergreen needleleaf	Closed forest, evergreen needle leaf	Open forest, evergreen needle leaf
	Tree cover deciduous broadleaf	Closed forest, deciduous broad leaf	Open forest, deciduous broad leaf
	Tree cover deciduous needleleaf	Closed forest, deciduous needle leaf	Open forest, deciduous needle lea
	Shrub cover evergreen	Shrubs	
	Shrub cover deciduous	Shrubs	
	Grasslands	-	
	Croplands	Cultivated and managed vegetation/agriculture (cropland)	Herbaceous vegetation
	Vegetation aquatic or regularly flooded	Herbaceous wetland	
	Lichen and Mosses	Moss and lichen	
	Bare areas	Bare / sparse vegetation	
	Built-up	Urban / built up	
	Open Water seasonal	-	
	Open Water permanent	Permanent water bodies	
	Snow and/or Ice	Snow and Ice	

7.1.2 Training sets: last version of HRLC map legend analysis

From Table 4 one can notice that few classes cannot be extracted from the CGLC: (1) grassland, (2) open water seasonal and (3) the distinction between evergreen and deciduous shrubland. While the open water seasonal class is not present in the CGLC, the grassland class should be linked to the Herbaceous vegetation one. However, from the qualitative example reported in Figure 14, the herbaceous vegetation class is mainly associated with pixels belonging to crops (see Figure 14a). This is true for most of the pixels of the considered study areas (Sentinel 2 tiles 21KXT, 21KUQ, 37PCP, and 42WXS). For this reason, the Herbaceous vegetation is included in the cropland class as 2nd choice. The team investigated also the possibility of extracting the grassland samples from the 2015 ESA CCI land cover product. Figure 20 shows two examples of ESA CCI land cover map of the KUQ tile. Also in this case, most of the pixels associated with the grassland class (i.e., in orange) correspond to cropland areas in the Sentinel 2 images.

Regarding the shrubland class, the team is investigating if there is the possibility to distinguish by photointerpretation evergreen and deciduous shrubland. Figure 22 reports an example of the study area where the shrubland class is present according to the CGLC map. The complex detection of this class by photo-interpretation is clearly visible in both the high-resolution product and the Sentinel 2 image.



(a) (b)

(c) (d)

	Evergreen broadleaf		Evergreen needleleaf		Deciduous broadleaf		Deciduous needle leaf
	Shrubland		Cropland		Permanent water bodies		Grassland

Figure 20. Example of grassland pixels present in the ESA CCI LC map: (a),(c) the 2015 ESA CCI LC map, (b),(d) the true color composition of the Sentinel 2 image acquired on the 23rd June 2018.

(a)

(b)

Figure 21. Example of the considered study area where the shrubland class is present according to the CGLC map (tile 21KXT): (a) high-resolution optical data, (b) Sentinel 2 image.

7.1.3 Training sets: proposed unsupervised automatic extraction

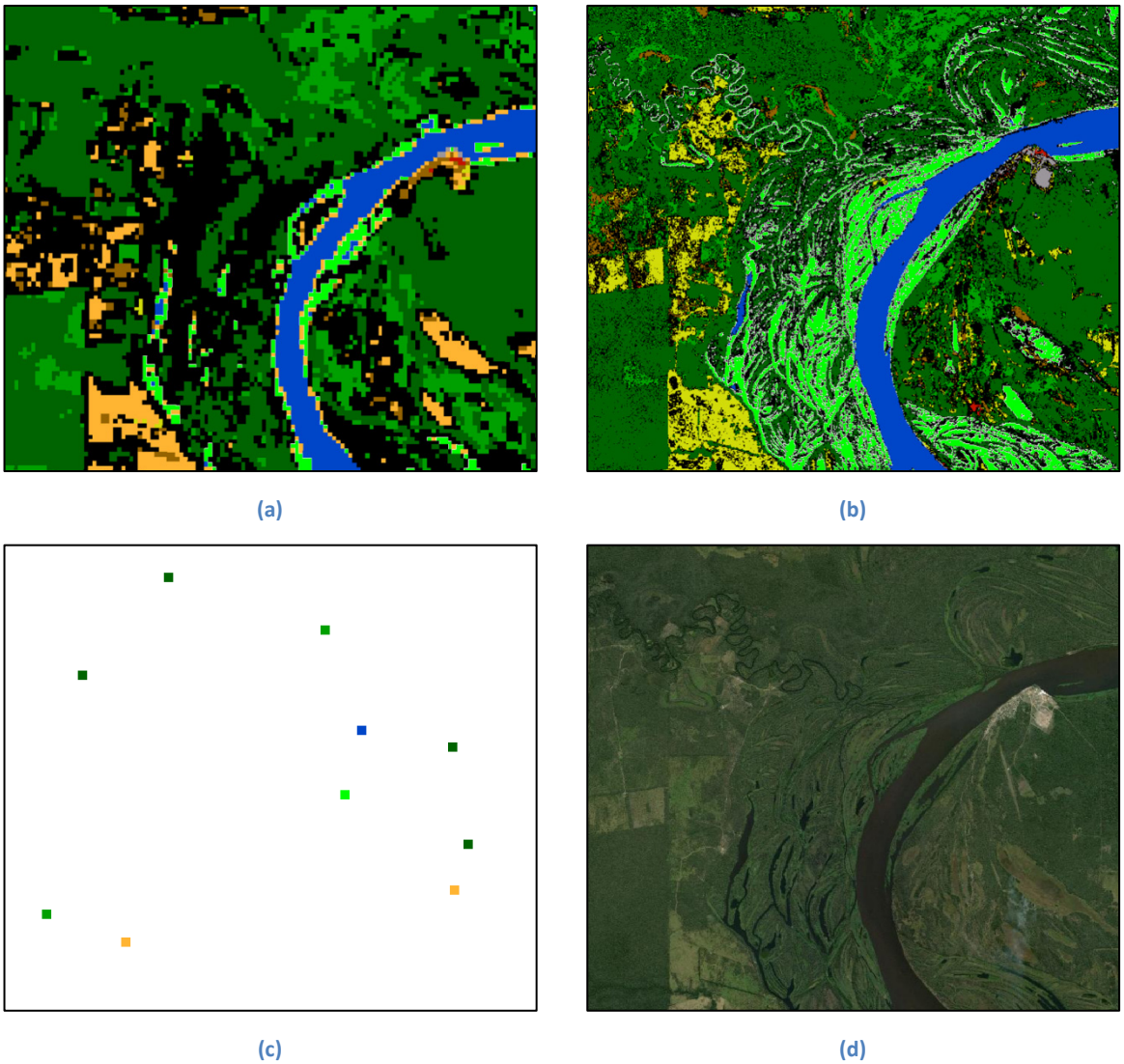


Figure 22. Unsupervised Training Set production: (a) CGLC map after the legend conversion, (b) intermediate map produced by the ensemble of five classifiers, (c) weak training samples selected, and (d) true color composition of the high-resolution optical image available on the considered study area.

Due to the need of generating a large training database, the weak training set production is completely unsupervised and automatic. The method aims to detect in the CGLC map those samples having the highest probability of belonging to areas correctly associated with their labels. Although existing thematic maps represent a valuable source of information, many difficulties arise when extracting labeled samples from them. Due to the medium spatial resolution, the label assigned to mixed pixels can be propagated to the pure pixels of Sentinel 2 images. Moreover, the considered maps are outdated and thus, they are not completely reliable.

To address all these issues, we perform an automatic and unsupervised analysis that extracts from the moderate resolution CGLC map, a weak but reliable training set. First, a random stratified sampling is performed by using the LC classes as strata. Five training sets are generated via bootstrap statistical method (e.g., without replacement) and used to train an ensemble of statistically independent classifiers. This condition allows us to

generate an intermediate thematic product obtained at 10 m spatial resolution by classifying the time series of Sentinel 2 images. Only the areas where the ensemble of classifiers agrees are kept. This increases the probability of selecting reliable samples to produce the final weak training database.

Figure 22 shows a qualitative comparison over a portion of the study area located in tile 21KUQ (Amazonia) between: (a) the coarse CGLC map after the legend conversion, (b) the intermediate product produced by the ensemble of five classifiers, (c) the weak training samples selected, and (d) a true color composition of the high-resolution optical image. The qualitative example demonstrates the importance of generating the intermediate product at 10 m spatial resolution. For instance, he geometrical detail of the aquatic vegetation present in the scene with respect to the original medium thematic product is sharply improved in the in the intermediate product with respect to the original one. This condition allows us to sharply increase the probability of selecting samples correctly associated with their labels with respect to the ones that can be directly selected from the CGLC map.

7.1.4 Eco-climatic products

Ecoregions, in the simplest definition, are ecosystems of regional extent. Specifically, ecoregions represent distinct assemblages of biodiversity—all taxa, not just vegetation—whose boundaries include the space required to sustain ecological processes. Ecoregions provide a useful basemap for thematic mapping to several extents because they draw on natural boundaries, define distinct biogeographic assemblages and ecological habitats within biomes, and assist in representation of Earth biodiversity.

ECOREGIONS 2017 product [66]: The 846 terrestrial ecoregions are grouped into 14 biomes and 8 realms. Six of these biomes are forest biomes and remaining eight are non-forest biomes. For the forest biomes, the geographic boundaries of the ecoregions and protected areas (UNEP-WCMC 2016) were intersected with the Global Forest Change data for the years 2000 to 2015, to calculate percent of habitat in protected areas and percent of remaining habitat outside protected areas. Likewise, the boundaries of the non-forest ecoregions and protected areas (UNEP-WCMC 2016) were intersected with Anthropogenic Biomes data (Anthromes v2) for the year 2000 to identify remaining habitats inside and outside the protected areas. Each ecoregion has a unique ID, area (sq. degrees), and NNH (Nature Needs Half) categories 1-4. NNH categories are based on percent of habitat in protected areas and percent of remaining habitat outside protected areas.

7.1.5 Remarks

In order to train a single classifier over a whole region, a huge amount of RAM would be needed. This holds true for any of the classifiers described in Section 6, i.e., RF and SVM. Therefore, the training/classification tasks need to be split according to well established spatial strategies. In general, classification methods are robust to strategies of this kind, especially RF and SVM. Since RF decision function is the majority vote of a committee of decorrelated decision trees, one can train several RF over different subsets of the training set and combine the results of each individual RF without much modification of the results. For SVM a similar approach is not straightforward. However, including a voting strategy on the outcomes of different subsets of training/classification stages is a consolidated approach in literature and gives satisfactory results.

We aim at defining a processing chain able to perform the training of several classifiers using data coming from a subset of the tiles. These subsets can be disjoint, comprise common tiles or contain all the tiles, which would result in the trivial case of a single classifier. Once the classifiers are trained, the classification step can implement two spatial strategies:

- each tile is classified using the classifier trained with itself;
- each tile is classified using several classifiers and the class labels are assigned according to a majority vote on their decisions.

The choice of the subset of tiles used for training is important. A different classifier per tile could be used or a classifier per group of adjacent tiles. The choice of disjoint classifiers trained over randomly selected tiles, followed by a majority voting of all classifiers gives a good trade-off between complexity and accuracy.

One of the main difficulties in the production of land cover maps over very large areas is the intra-class variability. Indeed, the same semantic category can have different spectral and temporal signatures depending on the areas. On the other hand, some classes are present in certain areas only, which means they are a minority over the complete area and they can therefore be difficult to account for by a classifier that is designed to maximise the overall accuracy, and therefore giving more weight to majority classes. To mitigate for these issues, supplying additional information to the classifier that identify climatic areas can be useful. Eco-climatic maps can be used to divide the area to be mapped into different strata that will be processed independently. In terms of implementation of the classification, there is no major difference with respect to the tiling approach. The intersection of the tile footprints and the eco-climatic areas is used to define the disjoint regions on which independent classifiers are trained. Then each region is classified using only the classifier trained over this region. In the case of very large regions the training data set is split into smaller subsets and the majority vote strategy of the tile-based approach is used.

8 SAR imagery classification

8.1 Feature extraction

To carry out the land cover classification using Sentinel-1 dual-Pol data sets based on the defined classes, reported in Table 1, the feature extraction will be based on the polarimetric information of data [67], [68].
To improve the ability of classifier to recognize and discriminate the different environment textures and morphological structures (e.g. urban areas, agricultural crops, forests, etc.), the amplitude of VH and VV channels and their combinations have been assumed.

Although the S1 data are not fully polarized, we can exploit the polarimetric information arising from the intensities of the VH and VV channels by means analysis on single channel (by choosing VH or VH) or on their combination (their mean or ratio, for instance). These features contain essential polarimetric information provided by the dual-Pol data since the polarimetry combination distinguishes specular scattering from diffuse scattering.

8.1.1 Texture analysis on single polarization

To analyze and exploring the spatial information contained in a single S1 image (VH or VV), a docker application has been developed in order to provide a set of filters that operate especially in spatial domain. The rationale for selecting these algorithms is the velocity of the execution. Although they might not be the most accurate ones, the possibility to apply them quickly to the SAR images in a large stack in a reasonable amount of time is an invaluable asset for wide area processing. The implemented techniques are summarized in the following list:

- *Mean filter* is one of the most widely used low-pass filters (LPF). It substitutes the pixel value with the average of all the values in the local neighborhood (filter kernel).
- *Median filter*, a non-adaptive filter and replaces each pixel value with the median of the pixel values in the local neighborhood.
- *Minimum (maximum) filter* is a non-linear filter that is located the darkest (brightest) point in an image. It is based on median filter since it is defined as his 0th (100th) percentile, i.e. by considering the minimum (maximum) of all pixels within a local region of an image.
- *Max-Min filter*, blurs the image by replacing each pixel with the difference of the highest pixel and the lowest pixel (with respect to intensity) within the specified window-size.

8.1.1.1 Mean filter

The Mean filter is a low-pass filter (LPF) and represents the simplest and easiest method of smoothing images, in addition to being very easy to implement. Mean filtering is usually thought of as a convolution filter. Like other convolutions it is based around a kernel, which represents the shape and size of the neighborhood to be sampled when calculating the mean. The idea of mean filtering is simply to replace each pixel value in an image with the mean ("average") of values belonging to neighborhood, including itself. Then, the filter window will be moved pixel-by-pixel until to scanner the whole image.

It does not remove the speckle from the image but averages it into one. In fact, the noise becomes less apparent, but the image looks "softened". Theoretically, dark and bright speckle pixels within the filter window can cancel each other out. The probability of such situations increases with the size of the filter window, 7×7 or 9x9 for example. However, it produces image blur, loss of details and, eventually, loss of spatial resolution, giving an image with less noise but less high frequency detail. For this reasons, 3x3 or 5x5 size filter are recommended. Note that the mean filtering is not suitable in case of pulse and spike noise since the shot noise pixel values are often very different from the surrounding values, they tend to significantly distort the pixel average calculated by the mean filter. The median filter is successful at removing pulse and spike noise while retaining step and ramp functions [69].

8.1.1.2 Median filter

The median filter is normally used to reduce noise in an image, somewhat like the mean filter. However, it often does a better job than the mean filter of preserving useful detail in the image. Like the mean filter, the median filter considers each pixel in the image in turn and looks at its nearby neighbors to decide whether it is representative of its surroundings or not. Instead of simply replacing the pixel value with the mean of neighboring pixel values, it replaces it with the median of those values. The median is calculated by first sorting all the pixel values from the surrounding neighborhood into numerical order and then replacing the pixel being considered with the middle pixel value.

By calculating the median value of a neighborhood rather than the mean filter, the median filter has two main advantages over the mean filter:

- The median is a more robust average than the mean and so a single very unrepresentative pixel in a neighborhood will not affect the median value significantly.
- Since the median value must be the value of one of the pixels in the neighborhood, the median filter does not create new unrealistic pixel values when the filter straddles an edge. For this reason, the median filter is much better at preserving sharp edges than the mean filter.

Hence, the median filter is edge preserving [70] although it may lead to the removal (or suppression) of small (also linear) objects from the image, exactly in the same way as it removes (or suppresses) speckle noise.

Applying a 3×3 median filter produces a noise reduction at the expense of a slight degradation in image quality. If we smooth the noisy image with a larger median filter, e.g. 7×7, all the noisy pixels disappear, but the image looks a bit "blotchy".

A good solution is to use 3×3 or 5x5 median filter [71] and passing it over the image more times in order to remove all the noise with less loss of detail, alternatively.

The mean and median filters meet with only limited success when applied to SAR data. One reason for this is the multiplicative nature of speckle noise, which relates the amount of noise to the signal intensity. The other reason is that they are not adaptive filters in the sense that they do not account for the speckle properties of the image. Adaptive filters, such as the Lee filter, assume that the mean and variance of the pixel of interest are equal to the local mean and variance of all pixels within the user-selected moving window.

8.1.1.3 Minimum and maximum filters

Minimum and maximum filters, also known as erosion and dilation filters, respectively, are morphological filters that work by considering a neighborhood (running window) around each pixel. The running window is an image area around a current pixel with a defined radius. For example, if we specify the radius = 1, the running window will be a 3x3 square around the target pixel, which is the smallest box size. The maximum and minimum filters are shift-invariant. Whereas the minimum filter replaces the central pixel with the darkest one in the running window, the maximum filter replaces it with the lightest one. In other words, the minimum filter extends object boundaries, whereas the maximum filter erodes shapes on the image. The odd size of the neighborhood considered for each pixel. Also in this case, the recommended size are 5x5 or 7x7 in order not to incur in issues have been addressed previously, see Section 8.1.1.1.

The docker offers also the possibility for each user to choose the kernel filter size adapted to its needs, but the default dimension is 9x9 because the implemented filter has shown satisfactory results both in terms of computational complexity and the quality of output image, due its ability in details preservation, edges definition.

8.1.1.4 Max-Min filter

The output image is given by the difference between dilation and erosion filters (described in previous section 8.1.1.3).

Hiring X as input image, the max-min filtered image is given taking into the account to the following simple expression:

$$Y = X_{max} - X_{min}$$

where Y is the resulting gray level image, whereas X_{max} and X_{min} are the maximum and minimum filtered version of input image X, respectively. The Max-Min filter blurs the image by replacing each pixel with the difference of the highest pixel and the lowest pixel (with respect to intensity) within a specified window-size (for example, the grayscale 3x3 or 5x5 pixel neighborhoods). To preserve much more spatial details and texture structures, we have set up window size to 9 by default, also according to the evaluations explained above.

8.1.2 Texture analysis on dual- polarization

SAR polarimetry allows for the retrieve of shape, orientation, and dielectric property information of scatterers [72],[73]. Since there are multiple polarimetric channels, it provides more information than single-pol SAR data. However, the richness of polarimetry is achieved by sacrificing the spatial resolution. To balance the trade-off, instead of a fully polarized SAR, Sentinel-1 mission provides partially polarized SAR data, known as dual-Pol data, with the VV and VH channels. To extract the polarimetric information of Sentinel-1 data, we used the signal acquired from VH and VV channels, and several composite images given by:

- *Ratio*, VV/VH
- *Sum*, $VH + VV$
- *Mean*, $(VH + VV)/2$
- *Difference*, $VV - VH$

These four features contain essential polarimetric information provided by the dual-Pol data. This polarimetry combination is able to distinguish specular scattering from diffuse scattering [74]. For the purpose of classification, these features are highly beneficial to differ classes with different surface roughnesses, such as water, plant, building, and soil. The aim is basically exploit the dual-polarization capability of S1 for providing as many ground surface information as possible [75].

8.1.3 Texture analysis by statistics

To increase the feature space it is also possible to add texture features by applying the *Grey Level Co-occurrence* matrix (GLCM), in order to retrieve second order textures [76]–[78].

This operation is done before applying the speckle filtering, since the despeckling destroys most of the image texture. For example, the classification accuracy related to perennial agroforestry land cover can be improved by using less correlated GLCM texture measures: Contrast, Entropy, Correlation, and Variance. The GLCM texture can be measured using a 5×5 moving window, one-pixel displacement, for example. In [79] it is shown that the GLCM texture measures are appropriated to discriminate vegetation types, and less sensitive to no vegetation cover. Is shown that the more informative variables are the VH Variance and Correlation of SAR images acquired in a dry season and, and VV Contrast of images in a wet season.

Instead, [80] highlights the importance of VH image that is the best band for differentiating agricultural land from other land cover types. The major differences in vegetation, their vertical structure, are captured in co-polarized (VV) band.

Another way for land-cover classification is to use multi-temporal SAR data (i.e. SAR data time series) analysis and extract features by considering the temporal variation of backscattering coefficients and information from interferometric data processing. The work [81] exploits the combination of the average backscattering coefficient and temporal variability. The average backscattering coefficient permits to classify water and urban areas, since they present very low and high signatures, respectively. The temporal variability, which is a main feature in multitemporal analysis, can be used to distinguish cultivated areas and water from the forest and urban classes.

The behavior of VH and VV backscatter signal is different over vegetated areas. Over vegetation land covers, there is much volume scattering of the radar signal. And volume scattering tends to cause a depolarization of the return signals, which then corresponds to a high backscatter in cross-polarization (VH or HV) bands. Thus, VH bands show a higher sensitivity to vegetation.

For the purposes of classification, these features are highly beneficial to diversify classes with different surface roughness, such as water, plants, buildings, and soil. In this manner, the classification maps may achieve high classification accuracy values. Specifically, the feature extraction step is preliminary to the classification step in the sense that only specific features for peculiar classes may be extracted and used each time. In addition, according to the technical literature, we also identified several works describing most performing classification methods able to classify different classes (water, urban areas, snow, for example) with a proper combination of features set. A preliminary list is reported in the following Table 5.

Table 5. Preliminary list of SAR features for subsets of classes.

Class	Feature(s)	Reference
Urban	Occurrence range, DEM slope	[G. Lisini, A. Salentinig, P. Du, P. Gamba, "SAR-based urban extents extraction: from ENVISAT to Sentinel-1", IEEE J. of Selected Topics in Applied Earth Observation and Remote Sensing, doi: 10.1109/JSTARS.2017.2782180, vol. 11, no. 8, pp. 2683-2691, Aug. 2018.]
Water	Average backscatter, the minimum backscatter of a time series and standard deviation of the backscatter	[Santoro, Maurizio, and Urs Wegmüller. "Multi-temporal SAR metrics applied to map water bodies." 2012 IEEE International Geoscience and Remote Sensing Symposium. IEEE, 2012.
Snow	σ^0VV band; backscattering ratio	[Tsai, Ya-Lun S., et al. "Wet and Dry Snow Detection Using Sentinel-1 SAR Data for Mountainous Areas with a Machine Learning Technique." Remote Sensing 11.8 (2019): 895.]
Crop	Occurrence variance; co-occurrence contrast	[Fontanelli, Giacomo, et al. "Agricultural crop mapping using optical and SAR multi-temporal seasonal data: A case study in Lombardy region, Italy." 2014 IEEE Geoscience and Remote Sensing Symposium. IEEE, 2014.]
Deciduous vegetation	Temporal signature	[Rüetschi, Marius, Michael Schaepman, and David Small. "Using multitemporal Sentinel-1 C-band backscatter to monitor phenology

		and classify deciduous and coniferous forests in northern Switzerland." *Remote Sensing* 10.1 (2017): 55.]
Evergreen vegetation	VV and VH channels	[Abdikan, Saygin, et al. "Land cover mapping using sentinel-1 SAR data." The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences 41 (2016): 757]
Soil	VV and VH channels	[Hu, Jingliang, Pedram Ghamisi, and Xiao Zhu. "Feature Extraction and Selection of Sentinel-1 Dual-Pol Data for Global-Scale Local Climate Zone Classification." ISPRS International Journal of Geo-Information 7.9 (2018): 379.]

At this point, features are computed according to the following steps:

a) Initially, the SAR time series is properly pre-processed by means of the methods and filters previously introduced in Section 4.
b) Then, all the de-speckled images in one year are first divided according to the season and then merged into one image per season by means of a temporal average. This step is performed as a thread-off between the need to keep multitemporal information and the one to reduce the computational load of the classification procedure.
c) Finally, the features useful for the extraction of the classes reported in the table above are computed for the final multitemporal sequence.

8.2 Classification

The classification procedure implemented in this work is based on a hierarchical extraction of specific classes followed by a general classification applied to the rest of the scene. Specifically:

- First, some of the classes that are recognizable using a specific subset of features are extracted from the data by means of unsupervised techniques. This is currently performed for the urban class, but plans are to implement a similar approach for water surfaces and snow cover.
- Then, supervised classifiers, namely Random Forest (RF) and Support Vector Machines with Radial Basis Function (RBF) kernel, are applied to the set of features highlighted in Table 3.

For the latter step, suitable training data are necessary. To avoid the unbearable cost of a manual extraction of high-resolution samples, in the following a procedure able to extract samples for many of the desired classes from existing maps is highlighted. This procedure must be complemented for specific areas and classes by more performing sample extraction methods (manual selection, for example), but it helps to reduce the cost of that procedure to a level which is manageable in the context of a global mapping methodology.

8.2.1 Training sets from medium resolution maps

To automatically carry out a classification based on a training set extracted from the medium resolution products, we start from the assumption to classify in high resolution only pure classes that were recognized in medium resolution maps.

Specifically, the medium resolution maps that were considered are:

- *ESA CCI-LC 2018 (300m)*: The annual ESA CCI LC maps cover a period of 24 years from 1992 to 2018 at a spatial resolution of 300m. These maps describe the Earth's terrestrial surface in 37 original LC classes based on the United Nations Land Cover Classification System (UN-LCCS) [82]. The product that covers the 2015 year have been assumed as baseline.
- *GLCNMO (1km)*: The Global Land Cover by National Mapping Organizations (GLCNMO) is geospatial information in raster format which classifies the status of land cover of the whole globe into 20 classes at a spatial resolution of 1 km [83]–[85]. The classification is based on LCCS developed by Food and Agriculture Organization of the United Nations (FAO).

The proposed strategy aims to exploit the information associated only to those MR classes that present a good correlation with the high-resolution legend, excluding for instance the MR mixed-classes of ESA CCI-LC product reported in Table 4.

Table 6. Mixed classes list of the ESA CCI-LC 2015 (330m) product

Values	ESA CCI-LC 2015 (300m) labels
30	Mosaic cropland (>50%)/natural vegetation (tree, shrub, herbaceous cover) (<50%)
40	Mosaic natural vegetation (tree, shrub, herbaceous cover) (>50%)/cropland (<50%)
100	Mosaic tree and shrub (>50%)/herbaceous cover (<50%)
110	Mosaic herbaceous cover (>50%) tree and shrub (<50%)
180	Shrub and herbaceous cover, flooded, fresh/ saline/brakish water

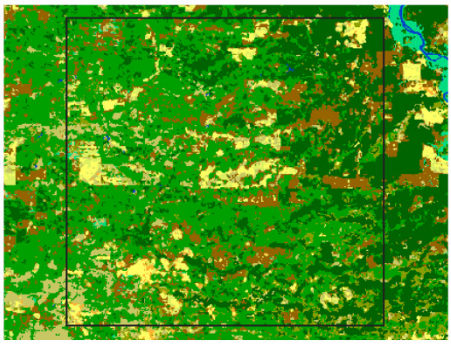
A comparison of the above-mentioned medium resolution map and the desired classes for HR mapping in this project has eventually brought to the results summarized in the following table:

Table 7. List of several medium resolution classes dealing to a training set for a high resolution classes subset

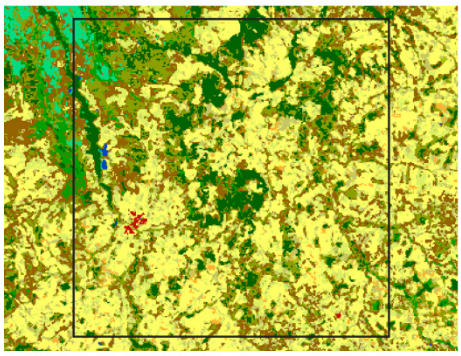
Value	CCI-HR LC classes	ESA CCI LC 2018 (300m) values	GLCNMO (1km) values
10	*Tree cover evergreen broadleaf*	50	
20	*Tree cover evergreen needleleaf*	70, 71, 72	
30	*Tree cover deciduous broadleaf*	60, 61, 62	
40	*Tree cover deciduous needleleaf*	80, 81, 82	
50	*Shrub cover evergreen*	121	
60	*Shrub cover deciduous*	122	
70	*Grassland*	130	
80	*Cropland*		11,13
90	*Vegetation acquatic or regularly flood*	160,170,180	
100	*Lichens and mosses*	140	
110	*Bare areas*	200,201,202	
111	*Sands*		17
112	*Rocks*		16
120	*Built-up areas*	Urban extraction methodology [22]	
130	*Water permanent*	210	
140	*Snow and/or ice*	220	

In the second and third columns of Table 7, several values of ESA and GLCNMO legends, respectively, are selected in a way to provide a redundant yet meaningful set of training points for the corresponding high-resolution

classes reported in the first column. For an easier reading of Table 7, the legends of ESA CCI-LC 2015 and GLCNMO maps have been shown in Figure 23 and Figure 24, respectively.



(a)



Legend of the global CCI-LC maps, based on LCCS

Value	Label	Color
0	No Data	
10	Cropland, rainfed	
11	Herbaceous cover	
12	Tree or shrub cover	
20	Cropland, irrigated or post-flooding	
30	Mosaic cropland (>50%) / natural vegetation (tree, shrub, herbaceous cover) (<50%)	
40	Mosaic natural vegetation (tree, shrub, herbaceous cover) (>50%) / cropland (<50%)	
50	Tree cover, broadleaved, evergreen, closed to open (>15%)	
60	Tree cover, broadleaved, deciduous, closed to open (>15%)	
61	Tree cover, broadleaved, deciduous, closed (>40%)	
62	Tree cover, broadleaved, deciduous, open (15-40%)	
70	Tree cover, needleleaved, evergreen, closed to open (>15%)	
71	Tree cover, needleleaved, evergreen, closed (>40%)	
72	Tree cover, needleleaved, evergreen, open (15-40%)	
80	Tree cover, needleleaved, deciduous, closed to open (>15%)	
81	Tree cover, needleleaved, deciduous, closed (>40%)	
82	Tree cover, needleleaved, deciduous, open (15-40%)	
90	Tree cover, mixed leaf type (broadleaved and needleleaved)	
100	Mosaic tree and shrub (>50%) / herbaceous cover (<50%)	
110	Mosaic herbaceous cover (>50%) / tree and shrub (<50%)	
120	Shrubland	
121	Evergreen shrubland	
122	Deciduous shrubland	
130	Grassland	
140	Lichens and mosses	
150	Sparse vegetation (tree, shrub, herbaceous cover) (<15%)	
151	Sparse tree (<15%)	
152	Sparse shrub (<15%)	
153	Sparse herbaceous cover (<15%)	
160	Tree cover, flooded, fresh or brakish water	
170	Tree cover, flooded, saline water	
180	Shrub or herbaceous cover, flooded, fresh/saline/brakish water	
190	Urban areas	
200	Bare areas	
201	Consolidated bare areas	
202	Unconsolidated bare areas	
210	Water bodies	
220	Permanent snow and ice	

Figure 23. The CCI-LC MR maps referred to Amazonian tile 21KUQ (q) Amazonian tile 21KXT (b) classified according to the legend of the global CCI-LC maps (c).

Value	Class Name
1	Broadleaf Evergreen Forest
2	Broadleaf Deciduous Forest
3	Needleleaf Evergreen Forest
4	Needleleaf Deciduous Forest
5	Mixed Forest
6	Tree Open
7	Shrub
8	Herbaceous
9	Herbaceous with Sparse Tree/Shrub
10	Sparse vegetation
11	Cropland
12	Paddy field
13	Cropland / Other Vegetation Mosaic
14	Mangrove
15	Wetland
16	Bare area,consolidated(gravel,rock)
17	Bare area,unconsolidated (sand)
18	Urban
19	Snow / Ice
20	Water bodies

Figure 24. Legend associated to GLCNMO medium resolution map.

The final step for training point selection is performed using random sampling. This is performed by first selecting the points belonging (for each HR class) to the corresponding classes in the MR maps into binary maps. To avoid inaccuracies and collect more reliable samples, a morphological erosion step is applied to this binary map, and only its "internal" area is considered. Then, random sampling is applied. The procedure is repeated for each class, and a consistent set of training samples is extracted.

This approach does not reduce the resolution of the final HR map, which is obtained considering the original resolution of S-1 data. Moreover, by selecting only classes that are not mixed, it allows to obtain reasonably good training samples at a very limited cost. Of course, these samples are as accurate as (in average) the maps from which they are taken, and this is the reason why robust classifiers, such as RF and SVM has been selected.

9 Decision fusion

Data fusion methodologies, and specifically the sub-class of decision fusion, allow making a common decision in case of multiple actors and opinions. Within the CCI+ HRLC pipeline, decision fusion combines the posterior probabilities associated with the outputs of the single classifiers that are applied to optical and SAR data separately. Therefore, multiple decisions are combined into a final result by taking into account the level of uncertainty associated with each source. This uncertainty is expressed precisely by the probabilistic characterization provided on a pixelwise basis by the aforementioned posteriors.

The sets of classes that can be accurately discriminated by using optical and SAR data separately do not coincide in general. While optical data are generally expected to be useful to the discrimination of all considered land cover classes, SAR data are expected to well discriminate especially built-up classes and water bodies. Accordingly, SAR and optical classification algorithms generally work on different, although obviously non-disjoint, sets of classes. Decision fusion methodologies are aimed at fusing posterior probabilities related to the classes in common across the two sets. Hence, a class-specific combination rule has been devised to take this into account and, correspondingly, integrate this fusion result on the common classes with the results obtained using only optical or SAR data for the remaining classes.

Specifically, the whole class legend Ω is divided into three disjoint subsets of thematic classes: Ω_O, the set of classes that are distinguished only by using optical data ("optical-exclusive"); Ω_S, the set of classes that are distinguished only by using SAR data ("SAR-exclusive"); and Ω_C, the set of classes which are discriminated by the classifiers operating with both data modalities (common classes). Accordingly, $\Omega = \Omega_O \cup \Omega_S \cup \Omega_C$. While the optical classifier works on the set of classes $\Omega_O \cup \Omega_C$, the SAR classifier outputs posterior probabilities for the set of classes $\Omega_S \cup \Omega_C$.

As a trade-off between computational complexity and expected accuracy, in the context of the CCI+ HRLC processing chain the following families of decision fusion methods are developed: weighted voting and consensus theoretic methods, and fusion strategies based on Markovian modelling (i.e., Markov and conditional random fields). Both families are combined with class-specific combination rules that take into account the aforementioned rationale. Details can be found in the following subsections.

9.1 Consensus Theory and Class-Specific Combination Rule

Consensus theory [87], [88] involves general procedures with the goal of combining multiple probability distributions to summarize their estimates. The problem can be formulated as the combination of different opinions. This is represented as the fusion of posterior probabilities coming from different classifiers, each associated with a particular data source.

Under the assumption that all the classifiers can be made into generating Bayesian outputs and that, accordingly, their predictions are endowed with a probabilistic characterization, i.e., pixelwise posteriors are available, the goal is to produce a single probability distribution that summarizes their estimates. The study of such combination procedures is called consensus theory.

A first well-known consensus rule is the linear opinion pool (LOP). Focusing on the specific case of optical and SAR classifiers as sources generating the posterior probabilities and keeping in mind that the two classifiers generally work on different sets of classes, let $\underline{x} = [\underline{O}, \underline{S}]$ be the input data vector on a generic pixel, resulting from the stacking of optical ($\underline{O}$) and SAR ($\underline{S}$) individual feature vectors, and let ω_j be the jth information class ($\omega_j \in \Omega$). The LOP functional can be expressed as:

$$\mathcal{C}(\omega_j|\underline{x}, \Omega_C) = \alpha\, P(\omega_j|\underline{O}, \Omega_C) + \beta\, P(\omega_j|\underline{S}, \Omega_C),$$

where $P(\omega_j|\underline{O}, \Omega_C)$ is the optical posterior probability of ω_j conditioned to the common subset of classes Ω_C and $P(\omega_j|\underline{S}, \Omega_C)$ is the SAR posterior probability conditioned to the same subset Ω_C. α and β are optical and SAR source-specific weights, respectively, and control the relative influence of the two sources. We note that the pixelwise outputs of the optical-based and SAR-based classification chains are $P(\omega_j|\underline{O}, \Omega_O \cup \Omega_C)$ and $P(\omega_j|\underline{S}, \Omega_S \cup \Omega_C)$, respectively, i.e., the pixelwise posteriors associated with the corresponding sets of classes. Deriving $P(\omega_j|\underline{O}, \Omega_C)$ and $P(\omega_j|\underline{S}, \Omega_C)$ (as well as $P(\omega_j|\underline{O}, \Omega_O)$ and $P(\omega_j|\underline{S}, \Omega_S)$) is straightforward.

LOP has several good properties: it is simple, it yields a probabilistic formulation, and the weights α and β can reflect the relative expertise of the optical and SAR classifiers, respectively. Moreover, if the data sources have absolutely continuous probability distributions, LOP may be related to an absolutely continuous distribution [88]. LOP also assumes that all the experts (classifiers) observe the input vector $\underline{x}$. Therefore, LOP can be viewed as a weighted average of the probability distributions from the experts that results in a combined probability distribution. Yet, LOP is a simple method and, besides the aforementioned advantages, has also several weaknesses [89]. An example is a possible dictatorship when Bayes' theorem is applied (i.e., a specific data source dominates in making a decision). Moreover, not deriving from the joint probabilities using Bayes' rule, it is also not externally Bayesian (does not obey Bayes' rule).

Another well-known and usually effective consensus rule, the logarithmic opinion pool (LOGP), has been proposed to overcome some of the problems of LOP. In the optical-SAR case addressed here, the LOGP functional can be defined as:

$$\mathcal{L}(\omega_j|\underline{x}, \Omega_C) = \alpha \log P(\omega_j|\underline{O}, \Omega_C) + \beta \log P(\omega_j|\underline{S}, \Omega_C)$$

LOGP differs from the linear version in that it is usually unimodal and less dispersed. Zeros are considered vetoes: if any of the two sources assigns a zero posterior (i.e. $P(\omega_j|\underline{O},\Omega_C)=0$ or $P(\omega_j|\underline{S},\Omega_C)=0$), then by definition $\mathcal{L}(\omega_j|\underline{x},\Omega_C)=0$. This dramatic behaviour is a drawback when the single-source predictions are very inaccurate and can be generated even by roundoff error. In order to prevent this, all posterior values are increased by the machine epsilon (the minimum number that can possibly be represented given a certain data type).

$\mathcal{C}(\cdot)$ and $\mathcal{L}(\cdot)$ provide probabilistic fusion results associated with the classes in common between the two single-sensor outputs, although they generally do not take values in the interval [0, 1]. Either can be mapped to proper posteriors by suitably transforming to a probabilistic output, which represents a fused posterior probability $P_{\mathcal{F}}(\omega_j|\underline{x},\Omega_C)$. In the case of LOP, $P_{\mathcal{F}}(\omega_j|\underline{x},\Omega_C)$ is computed from $\mathcal{C}(\omega_j|\underline{x},\Omega_C)$ by just re-normalizing so that the sum over all $\omega_j \in \Omega_C$ is unity. In the case of LOGP, the following softmax operator is appropriate to take into account the logarithmic relation between the $\mathcal{L}(\cdot)$ functional and the original probabilities:

$$P_{\mathcal{F}}(\omega_j|\underline{x},\Omega_C)=\frac{\exp\mathcal{L}(\omega_j|\underline{x},\Omega_C)}{\sum_{\omega_k\in\Omega_C}\exp\mathcal{L}(\omega_k|\underline{x},\Omega_C)}$$

This probabilistic fusion output $P_{\mathcal{F}}(\cdot)$ covers the subset of classes in common across the two single-sensor classifications. To extend it to the whole set of classes, the posterior probability (unconditional with respect to Ω_C) can be defined according to the total probability theorem:

$$P_{\mathcal{F}}(\omega_j|\underline{x})=P(\omega_j|\underline{x},\Omega_C)P(\Omega_C|\underline{x})+P(\omega_j|\underline{x},\Omega_O)P(\Omega_O|\underline{x})+P(\omega_j|\underline{x},\Omega_S)P(\Omega_S|\underline{x})=$$
$$=P_{\mathcal{F}}(\omega_j|\underline{x},\Omega_C)P(\Omega_C|\underline{x})+P(\omega_j|\underline{O},\Omega_O)P(\Omega_O|\underline{x})+P(\omega_j|\underline{S},\Omega_S)P(\Omega_S|\underline{x}),$$

where the aforementioned probabilistic fusion result $P_{\mathcal{F}}(\omega_j|\underline{x},\Omega_C)$ is used for the common classes, whereas the optical-based and SAR-based posteriors $P(\omega_j|\underline{O},\Omega_O)$ and $P(\omega_j|\underline{S},\Omega_S)$ are used for the two sets of exclusive classes. The aggregated posteriors of the three subsets of thematic classes Ω_O,Ω_S and Ω_C are modelled as follows:

$$P(\Omega_O|\underline{x})=\lambda P(\Omega_O|\underline{O},\Omega_O\cup\Omega_C),\qquad P(\Omega_S|\underline{x})=(1-\lambda)P(\Omega_S|\underline{S},\Omega_S\cup\Omega_C),$$
$$P(\Omega_C|\underline{x})=\lambda\,P(\Omega_C|\underline{O},\Omega_O\cup\Omega_C)+(1-\lambda)P(\Omega_C|\underline{S},\Omega_S\cup\Omega_C),$$

where $0\le\lambda\le1$. This choice makes sure that the resulting terms correctly sum to unity (for all $\lambda\in[0,1]$), combines the optical- and SAR-specific probabilistic outputs using a LOP-like formulation on the common classes, and expresses the items associated with the exclusive classes as functions of the output of one of the two single-sensor processing chains. To determine an appropriate value for λ, we note that, in the limit case $\Omega_S=\emptyset$ (i.e., if the set of classes discriminated using SAR is a subset of the set of classes discriminated using optical data), $\lambda=1$ is a desired choice. Vice versa, in the limit case $\Omega_O=\emptyset$, a desired value is $\lambda=0$. A suitable weight that covers both limit cases is:

$$\lambda=\frac{P(\Omega_O)}{P(\Omega_O)+P(\Omega_S)},$$

where the prior probabilities $P(\Omega_O)$ and $P(\Omega_S)$ are estimated on the training set. Therefore:

$$P_{\mathcal{F}}(\omega_j|\underline{x})=P_{\mathcal{F}}(\omega_j|\underline{x},\Omega_C)\left[\lambda\,P(\Omega_C|\underline{O},\Omega_O\cup\Omega_C)+(1-\lambda)P(\Omega_C|\underline{S},\Omega_S\cup\Omega_C)\right]$$
$$+P(\omega_j|\underline{O},\Omega_O)\lambda P(\Omega_O|\underline{O},\Omega_O\cup\Omega_C)+P(\omega_j|\underline{S},\Omega_S)(1-\lambda)P(\Omega_S|\underline{S},\Omega_S\cup\Omega_C).$$

This combination rule is applicable to all cases, independently on the set of classes on which the two classifiers works. It is worth noting that, in the fusion of optical and SAR data, a frequent scenario is that one of the two sources discriminates among a larger set of classes than the other source. In particular, SAR-based classifiers typically work on a set of classes which is a proper subset of the set of classes considered by optical classifiers. In this case (consistent with the areas considered for the round robin experiments in PVASR-v1), we have $\Omega_S=\emptyset$ and then $\Omega_S\cup\Omega_C=\Omega_C,\Omega_O\cup\Omega_C=\Omega$, and $\lambda=1$. Therefore the previous formulation simplifies as follows:

$$P_{\mathcal{F}}(\omega_j|\underline{x})=P_{\mathcal{F}}(\omega_j|\underline{x},\Omega_C)P(\Omega_C|\underline{O},\Omega)+P(\omega_j|\underline{O},\Omega_O)P(\Omega_O|\underline{O},\Omega),$$

where it is possible to remove the conditioning on the whole set of classes:

$$P_{\mathcal{F}}(\omega_j|\underline{x}) = P_{\mathcal{F}}(\omega_j|\underline{x},\Omega_C)P(\Omega_C|\underline{O}) + P(\omega_j|O,\Omega_O)P(\Omega_O|\underline{O}),$$

which is the formulation used in PVASR-v1 to fuse posterior probabilities in the experiments on the round robin areas.

Within the HRLC pipeline, special focus is given to the definition of the weights α and β. Several approaches are being explored. The first is the use of uniform weights, which formalizes the case in which the decision maker has no knowledge on which source is more reliable. On one hand, this is straightforward; on the other hand, it does not benefit from the aforementioned properties of optical and SAR data in terms of the capability to discriminate the various classes. More accurately, it is possible to assign the weights proportionally to a score that is set according to the "goodness" of each source, where a higher score indicates a better (i.e., more reliable) source. This scoring may be accomplished by assessing the accuracy of the land-cover predictions coming from the optical and the SAR sources. Finally, another solution that is considered is to compute the weights through linear or nonlinear optimization [90] [87]. In particular, the method in [17] which is based on the expectation-maximization (EM) algorithm, can be incorporated into the HRLC pipeline. It regards a LOGP-type model in the framework of unsupervised change detection and will be generalized here to the case of supervised land-cover classification.

9.2 Markov Random Fields

Markov random fields (MRFs) are probabilistic graphical models able to include contextual information in the form of class interactions between neighbouring pixels. An MRF is determined by an energy function, whose minimization with respect to the labels is equivalent to the application of a maximum a-posteriori criterion [91]. Considering an MRF model in which only up to pairwise clique potentials are non-zero (comparing items one couple of nodes at a time), this energy is composed of two main terms: one characterizing class likelihood at the pixel level (depending on per-class scores obtained from any method able to estimate posterior or class-conditional probability density functions), and another promoting label smoothness in a local neighbourhood [91]. This means that the model encourages two neighbouring pixels to be labelled with the same class.

Let Ω be again the set of thematic classes. Define the regular pixel lattice as I, and let y_i be the class label of the i-th pixel ($y_i \in \Omega, i \in I$). The MRF considers y_i as sample of the random field $Y = \{y_i\}_{i\in I}$ of class labels, which is discrete-valued. A neighbourhood system $\{\partial i\}_{i\in I}$, which provides each i-th pixel with a set $\partial i \subset I$ of neighbouring pixels, is defined [92]. It is possible to choose different kinds of adjacency systems: the ones that have being used the most include the first- and second-order connectivity [92]. In the former, ∂i is made of the four pixels adjacent to the i-th pixel (four-connected) while in the latter the eight pixels surrounding it are considered.

Considering the aforementioned frequently used family of the MRF models in which only up to pairwise clique potentials are non-zero, the energy can be written as:

$$U(Y|X) = -\sum_{i\in I} \alpha \log P(y_i|\underline{x}_i) - \gamma \sum_{\substack{i\in I \\ j\in\partial i}} \delta(y_i, y_j).$$

where α and γ are positive weights and $\delta(\cdot)$ is the Kronecker impulse. In the multi-sensor case, a different unary term is added for each sensor, so that it is possible to fuse the different posterior probabilities while enforcing contextual relationships. The overall equation is:

$$U(Y|X) = -\sum_{i\in I}\sum_{s=1}^{S} \alpha_s \log P(y_i|\underline{x}_{is}) - \gamma \sum_{\substack{i\in I \\ j\in\partial i}} \delta(y_i, y_j),$$

where the notation x_{is} indicates the dependence of image data on both the pixel location i and the sensor s ($s = 1,2,\dots,S$), S is the number of sensors, and $\{\alpha_s\}_{s=1}^{S}$ is a set of positive weights.

Within the HRLC pipeline, in order to ensure consistency with the aforementioned pixelwise formulation and inspired by the similarity between the unary term and LOGP, the MRF approach is applied to the posterior probabilities resulting from the pixelwise fusion of the outputs of the optical and SAR classifiers. Therefore, in our specific setting, the overall equation becomes:

$$U(Y|X) = -\sum_{i \in I} \alpha \log P_{\mathcal{F}}(y_i|\underline{x}_i) - \gamma \sum_{\substack{i \in I \\ j \in \partial i}} \delta(y_i, y_j),$$

As compared to the previous fusion approaches, the strategy based on MRFs incorporates spatial information, an important contribution in the application to high-resolution remote sensing imagery, which is intrinsic in the HRLC project. The weights α and γ that tune the tradeoff among the various contributions to the energy function U are optimized by extending to the MRF fusion formulation the approaches previously described, with regard to the consensus formulation.

In the application of MRF-based methods to decision fusion, special focus is devoted to the minimization of the energy function U with respect to the random field Y of the class labels. First, as an efficient tradeoff between accuracy and computational burden, the iterated conditional mode (ICM) algorithm is adopted. It ensures short execution times, yet, it converges to a local minimum of the energy, which can be possibly suboptimal [93]. We shall investigate, either methodologically or experimentally, the opportunity to make use of global (or near-global) energy minimization methods based on graph-theoretic concepts (namely, graph cuts [94] and belief propagation methods [93]). On one hand, they ensure convergence to minima with stronger optimality properties than ICM. On the other hand, their computational burden is significantly higher and needs to be properly evaluated according to the data size involved in the HRLC project.

9.3 Deep Learning Solution

As an alternative to the aforementioned approaches to decision fusion, the multisensor fusion stage of the HRLC processing chain can also benefit from deep learning architectures. In this case, multi-sensor classification and fusion are dealt with by a deep convolutional neural network [50], [95], [96], [97] rather than by the specific aforementioned formulations. This is promising from the viewpoint of classification performance as confirmed by the accuracy gain observed in several recent international contests, in which deep learning solutions have overcome previous methods (e.g., recent IEEE GRSS Data Fusion Contests [98], [99] or ISPRS 2D Semantic Labeling competitions [100], [101]). On the other hand, the implementation, training, and computational complexity of the deep formulation will be significantly higher than those involved by the previous, more traditional, approaches.

In the specific case of the decision fusion block of the HRLC processing chain, an effective deep learning formulation would be based on the aforementioned CNN, autoencoder, and adversarial components that have been mentioned in previous sections. Adversarial networks are especially promising in this case thanks to their domain adaptation capabilities and to the opportunity to use them to map optical and SAR products into a homogeneous domain [52] (see also Section 5).

10 Multitemporal change detection and trend analysis

In accordance with the SoW [AD2] and as per the lessons learnt from the CCI MRLC, the scheme shown in Figure 25 is used for the generation of HRLC change products. In particular, the multitemporal change detection (CD) and trend analysis processing chain, assumes to have the entire data time series (both optical and SAR) from 1990-2015 already pre-processed. Additional to this information, this processing chain requires as input the HRLC static map (10m) and the 5 years regional HRLC maps (30m). As output from the processing chain, there will be the change information at 30 m and yearly time scale.

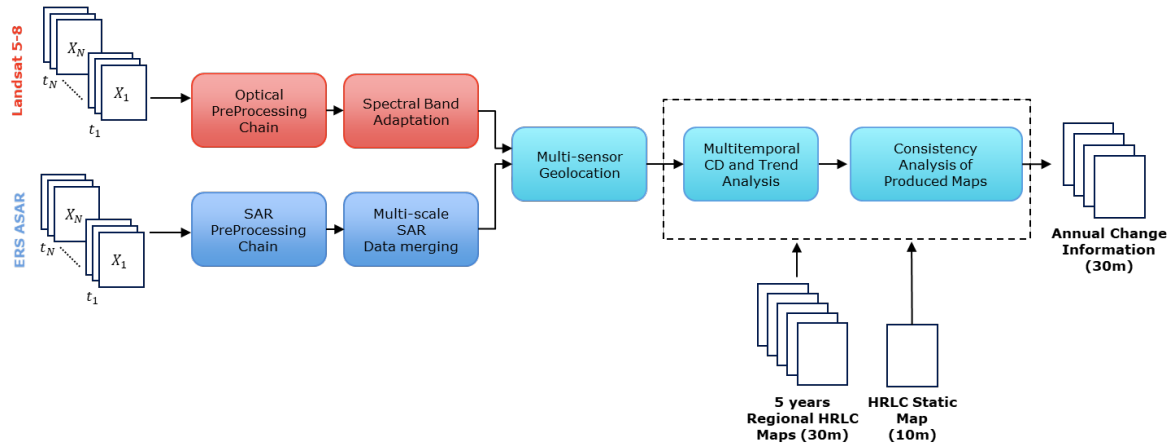


Figure 25. Block-based representation of the processing chain for the multitemporal change detection and trend analysis.

Changes can be divided into three classes [102]: (1) seasonal changes, impacting plant phenology or proportional cover of LC types with different plant phenology; (2) gradual changes such as inter-annual climate variability (e.g., trends in mean NDVI) or gradual change inland management or land degradation; and (3) abrupt (or permanent) changes, caused by disturbances such as deforestation, urbanization, floods, and fires.

The CCI HRLC change products will therefore be developed with an emphasis on quantification of the variability/changes at: seasonal, inter-annual and longer time scales (abrupt or permanent changes). The analysis is performed over the products derived from the multi-sensor geolocation step, plus the HRLC static and 5 years regional maps. For seasonal and longer time scales cases, methods that allow for the analysis of time series trends are considered. In the inter-annual case (which can be understood as 1-5 years), standard CD algorithms available in the literature are to be considered.

The analysis will be performed in a top-down time scale direction. In other words, abrupt/permanent changes occurring at longer time scales will be first identified in an unsupervised way. Knowing between which 5-years period this change has occurred, we will be further analysed and searched for inter-annual changes in a supervised and/or unsupervised matter. Finally, for plant/vegetation LC types, we will further analyse seasonal changes (supervised way).

10.1 Abrupt/permanent change and trend detection

A limited number of methods have been developed in the literature that allow the analysis of long time series (with 16 days acquisitions) and can be considered as scalable to the spatial resolutions of the available sensors in this project. Possible adaptation/combination is foreseen, given the fact that most of state-of-the-art methods: (1) have been developed for medium and/or low spatial resolution applications; (2) make use of a single spectral value per each evaluated year; and (3) focus on single LC only (e.g., forest and/or vegetation). In order to map the abrupt/permanent changes, combination of two main methods are to be considered:

- Breaks For Additive Seasonal and Trend (BFAST) [81];
- A Bayesian Estimator of Abrupt change, Seasonal change, and Trend (BEAST) [103].

BFAST is a generic CD approach for time series, involving the detection and characterization of BFAST. BFAST integrates the iterative decomposition of time series into trend, seasonal and noise components with methods for detecting changes, without the need to select a reference period, set a threshold, or define a change trajectory. The main limitation of this method is that it has been developed for MODIS data and tested mainly in NDVI index, and a few vegetation indices, and in particular for forest change detection. Adaptation to both HR data and other spectral information is thus required.

BEAST is an ensemble algorithm, and as such it quantifies the relative usefulness of individual decomposition models, leveraging all the models via Bayesian model averaging. It has been developed for Landsat and MODIS data. BEAST is able to detect change-points, seasonality, and trends in the data reliably; it derives realistic nonlinear trends and credible uncertainty measures (e.g., occurrence probability of change-points over time)—

some information difficult to derive by conventional single-best-model algorithms but critical for interpretation of ecosystem dynamics and detection of low-magnitude disturbances. The combination of many models enabled BEAST to alleviate model misspecification, address algorithmic uncertainty, and reduce overfitting. BEAST is generically applicable to time-series data of all kinds (offering an advantage for the CCI+ problem, where both optical and SAR images are to be considered). It offers a new analytical option for robust change-point detection and nonlinear trend analysis and helps exploit environmental time-series data for probing patterns and drivers of ecosystem dynamics.

The combination of these two methods [104], plus further improvements, will allow us to: i) detect multiple abrupt/permanent changes in the seasonal and trend components of the time series, ii) characterize the gradual and abrupt ecosystem changes by deriving the time, magnitude, and direction of change within the trend component of the time series; and iii) generate color-coded maps where different colours represent the year in which a given change has occurred. The optical and SAR data will be studied in separate ways, focusing in each case in particular LCs (e.g., vegetation in optical and cities in SAR), in accordance to the HRLC static map. Figure 26 shows the general block scheme followed in this case, where features are first extracted from both optical and SAR TS (relying on the LC type). As second step, feature TS are regularized to compensate for further errors from pre-processing steps. As third step, combination of adapted BFAST and BEAST algorithms will be used to generate the color-coded change map. Additional information about the time and duration of the change (if found) will be provided as well.

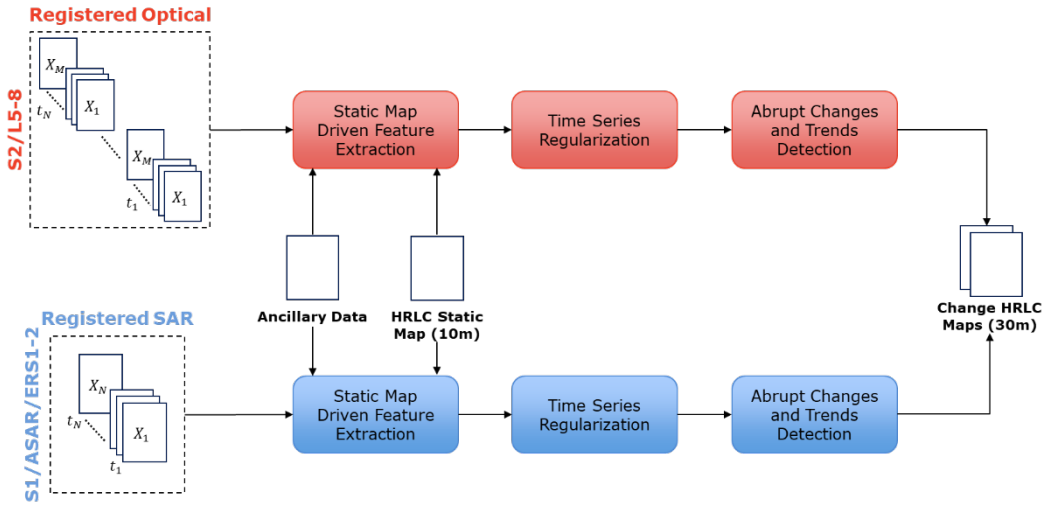


Figure 26. Block-based representation of the processing chain for the abrupt/permanent change and trend detection.

10.2 Inter-annual change detection

Opposite to the long time series analysis case, plenty of methods exist in literature that deal with the bi-temporal CD problem [105] both from supervised and unsupervised perspectives, in optical and SAR data. Any of these methods can be used for the bi-temporal CD part.

At this stage of the multitemporal CD step, we are aware of when the change occurred, but we are not aware of the type of change. As shown in Figure 27, we take advantage of the change HRLC maps (30m), the HRLC static map (10m) and the 5 years regional HRLC maps (30m), to map the type of change. To this end, we will focus the attention on changes of interest happening over a 5 years span (e.g., between 2010 and 2015). Two situations can be presented in here: (1) only one type of change has occurred in the 5-years span or (2) more than one type of change has occurred. In the first situation, the solution is straight forward (and supervised): we compare the initial and ending HRLC maps to understand the type of change occurred. In the second situation, semi-supervised CD approaches must be considered, and original optical and SAR data used as extra information. The reasons to pick a semi-supervised approach over a supervised or unsupervised one are related to the complexity of the problem and the need to know the type of change occurred, respectively. As a result, inter-annual CD

maps (30m) showing when and what type of change has occurred will be produced. There remains the limitation as per number of images available per year to perform the corresponding analysis.

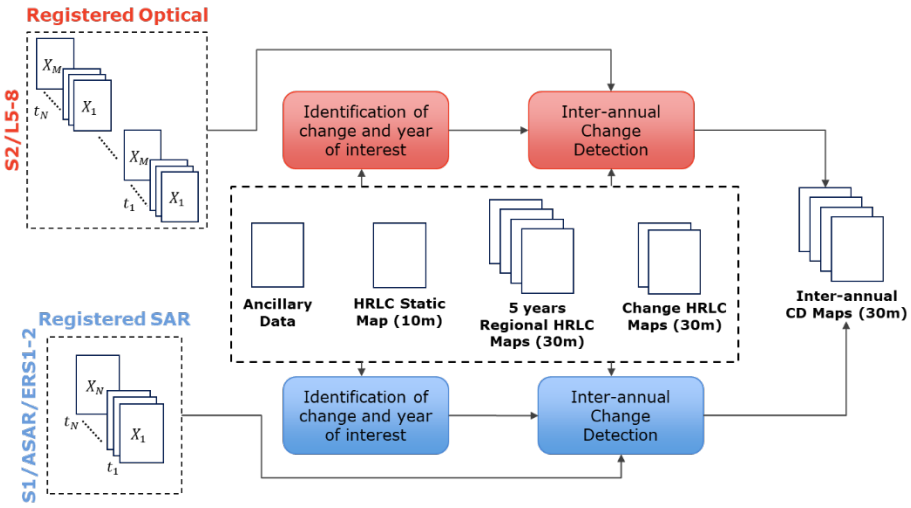


Figure 27. Block-based representation of the processing chain for the inter-annual change detection.

10.2.1 A multi-dimensional Dynamic Time Warping (DTW) strategy for change detection in long and dense optical time series.

We propose a semi-supervised procedure for prototyping/modelling the signatures linked to a set of pre-defined classes (by the HRLC Static Map for the first year and then by the 5 year Regional HRLC Maps). If the same LC class is been compared, temporal signatures and temporal evolution/profiles are expected to be similar, and therefore comparable. If the LC class is different, we expect for it to be a change or transition from one class to another. The proposed method creates models of the temporal evolution of different LC classes trends in high spatial multi-spectral Satellite Image Time Series (SITS) to generate multi-temporal class similarity profiles and detect possible changes from a class to another. The similarity between the built prototypes and a given pixel temporal signature is exploited by means of a multi-dimensional DTW.

Figure 28 presents the general block-scheme of the proposed method, where it is possible to identify three main steps: (1) time series reconstruction, (2) LC training and (3) LC classification and CD. There are two inputs for all these steps, being the first one the HR multi-spectral dense SITS, pre-processed in the previous phases. The pre-processing step is crucial to ensure all the images composing the SITS to be comparable, mitigating all the possible inconsistencies between images acquired at different times. The second input is the training set, an ensemble of spatial samples associated with a label (HRLC static map), stable for a sufficiently long period (over a year). The training set fidelity is a crucial requirement for the correct execution of the algorithm. The information carried by the training set about the stability of a class in a defined period ensures the possibility to extract the temporal signatures associated with a label and use them to generate Class Prototypes (CPs).

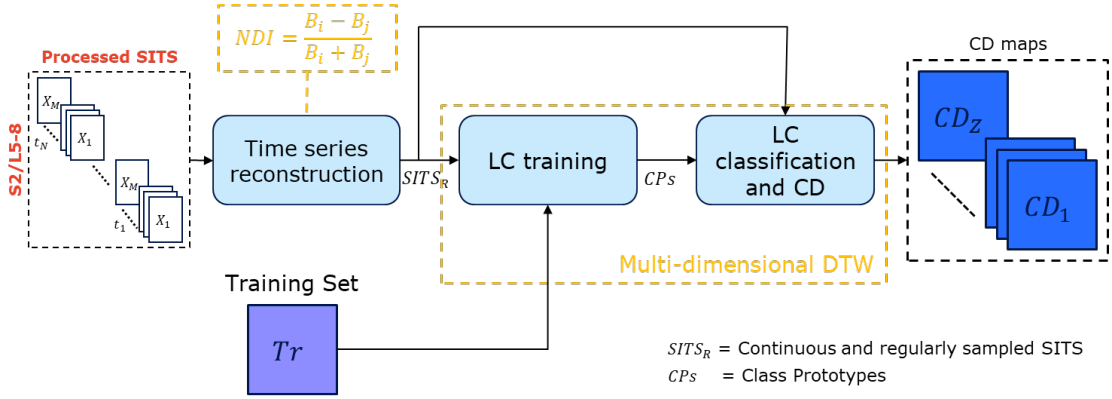


Figure 28. Block-based representation of the proposed approach for CD based on multi-dimensional DTW.

Let $SITS = \{X_1, X_2, \dots, X_N\}$ be a pre-processed Satellite Image Time Series acquired over the same geographical area in the period $[t_1, t_N]$. Assume the SITS have non-uniform time sampling, and each image has a total number of P pixels. Given an image X_n, each pixel value represents the surface reflectance value in a given spatial position and a temporal instant t_n. Considering all the pixel values in the SITS, it is possible to retrieve the temporal signature of a pixel TS_p (with $1 < p < P$) in the interval $[t_1, t_N]$. Let $B = \{b_1, b_2, \dots, b_K\}$ be the set of bands that compose the images taken into consideration and K the total number of bands, involving in K temporal signatures for each pixel. The temporal signature associated with a spatial position inside the geographical area is strictly related to the LC and LU inside the spatial portion represented by the pixel. Assume the LC classes discriminable inside the investigated area are $C = \{c_1, c_2, \dots, c_L\}$, and the total number of classes is equal to L. Finally, let $Tr = \{Tr^1, Tr^2, \dots, Tr^L\}$ be the training set representing the L classes composed by $Tr^c = \{Tr_1^c, Tr_2^c, \dots, Tr_{V_c}^c\}$. All the training samples linked to a spatial position are associated with a LC class and characterized by class stability inside a sufficiently long period T_{stab}. V_c is the total number of training samples for the class c, and V is the total number of training samples in the whole Tr. The T_{stab} in here is set to 1 year, ensuring the maintenance of natural cycle characteristics of spectral temporal signatures.

10.2.1.1 Time series reconstruction.

The time series reconstruction stage allows generating a sequence of values, denser than the source signal. The temporal signature is expected to be a truthful behaviour smooth and continuous. Moreover, this stage combines the original satellite spectral bands in higher Feature Space (FS) to recognize the temporal class evolution in a precise way. The spatial position, stability period and class label information carried by Tr allow the extraction of the temporal signatures of the training set TS_{Tr}. The temporal signatures contained in TS_{Tr} are finite-length discrete sequences that describe the behaviour of a particular class. To understand how the TS_{Tr} are correlated, a measure to compare the temporal sequences is needed.

At this stage, the temporal signature is a raw signal characterized by non-equally distributed temporal sampling and non-continuous trend, also affected by noisy oscillation not corrected in the pre-processing step. The state of the art literature mainly compares vegetative profiles between inner class temporal signatures. The behaviours are modelled, taking into account vegetation cycles and cycling harmonics models. The usage of those strategies does not fit the case of multiple class trends and fails in the presence of abrupt changes or number of cycles different from the pre-established values. The development of an ad-hoc non-parametric strategy to reconstruct the temporal signature is needed. The time series reconstruction stage consists of transforming the original temporal signal into a harmonious and plausible sampling sequence, proper to perform the successive analysis.

Taking inspiration from [106], a non-parametric regression is used and adapted to produce continuous and regularly sampled temporal signatures. To do so, four steps are followed: (1) Computation of Normalized Difference Indices (NDI), (2) uniform sampling interpolation, (3) low pass filtering and; (4) non-parametric regression through a Multi-Layer Perceptron Neural Network (MLP-NN). First, the spectral temporal signatures are combined, generating NDI arrays (FS). The combination of the source signals in the K bands produces an increased number of features. The NDI temporal signatures are then interpolated, taking into account the density and the shape maintenance requirement. A low pass filter reduces the intensity of high-frequency oscillations not usual in the LC temporal signatures, achieving a more smooth behaviour. Last, a non-parametric regression captures the temporal signatures trend reducing the profile complexity and arithmetic dependency.

The choice of a suitable FS is one of the fundamental elements to be able to distinguish the spectral trends belonging to the set of classes C labeled in the Tr. All possible couples of the available original sensor bands are combined to compute the NDIs. This stage transforms the B-dimensional FS into an F-dimensional FS (see equation (1)). The employment of NDIs, reduces the undesirable oscillations that mark the spectral bands. The ratio between various bands is valuable in the analysis between different classes, and the NDI values, included in the $[-1, 1]$ interval, are suitable for reliability comparison in a successive step.

$$F = \frac{1}{2}(K-1) \times K \quad (1)$$

10.2.1.2 LC training.

The temporal signatures associated with the spatial position of the Tr coming from the HRLC static map (or the 5 year HRLC maps), is used to build a set of CPs that allow to understand the evolution of different pixels in the study area. Three challenges are faced in order to generate the CPs:

1. Understand the similarity between the temporal signatures belonging to the same/different class;
2. Identify suitable features for the comparison of temporal signatures in a multiple feature (MF) space;
3. Deal with possible mislabelled classes in Tr as well as with classes presenting multiple temporal behaviours (e.g., cropland).

The LC training step manages those challenges employing a similarity measure based on the DTW and Multi-Dimension DTW (MDDTW). First, the DTW similarity measure is computed between all possible couples of TS_{Tr} in all the NDI features. The output of this operation is a set of SF DTW Similarity Matrices $\{SM^1, SM^2, \dots, SM^F\}$. The analysis of the obtained SF SM^f is required to choose the most suitable features to continue the study and fix the Class Clusters Parameters P_{CC} required for the CPs generation. The TS_{Tr} are then reduced by a feature selection step and used to compute the MF DTW Similarity Matrix SM^{MF}. The SM^{MF} is the main ingredient of the CPs generation driven by the parameters P_{CC}. The final output of the LC training step is the set of CPs depicting the various behaviours of the classes in a MF space. Usually, the set of classes C labeled in the training set is not a sufficient categorization for the analysis of temporal signatures. The LC training aims to prototype different trends for the same class c by forming groups (clusters) of similar temporal signatures belonging to TS_c, where each temporal signature is linked to a training sample. The clusters are generated following a set of rules and must respect boundaries imposed by P_{CC}. Also, the similarity analysis of SM^{MF} allows to find of a set of meaningful class similarity thresholds Ω convenient to understand the CPs generation results.

10.2.1.2.1 Dynamic Time Warping (DTW)

The measure selected for the comparison of the temporal signatures is the DTW [107] similarity measure. The DTW can handle the temporal distortion and shifting that characterize the temporal signatures, by exploiting the opportunity to find the optimal alignment between sequences. DTW also provides a similarity measure (SM) to quantify the similarity of the compared profiles. The introduction of the DTW SM solves the challenges linked to possible distortions or shifts of the same behaviour conducted by various temporal signatures. DTW stands out from the Euclidean distance thanks to the ability to capture flexible similarities. The aligning procedure of the coordinates inside both sequences consists in linking each element of the first sequence with at least one element of the second sequence.

10.2.1.2.2 Feature space analysis and reduction

The feature space analysis aims at searching the most suitable FS for the comparison of the temporal signatures, as well as the best parameters for the generation of the CPs. The strategy models the DTW similarity distributions between groups of similarity measures and computes a Dissimilarity Measure (DM) for each couple of classes in C. In this step, the SF SMs are employed separately, and jointly with SM^f. Two input parameters are required to carry out the task: P_A and P_B. P_A imposes the minimum number of samples per cluster, whereas P_B imposes the minimum number of samples per class. In the FS analysis stage, P_A assures the comparison of a number of SM bigger than a minimum threshold. P_B guarantees an amount of similarity distributions comparisons higher than a minimum threshold. The designed strategy allows simulating the comparison between SM distributions of different classes and considers the parameters that are used in successive stages to identify the most suitable ones. The two parameters are unknown and need to be found by exploring larger parameter space, computing the DMs for a set of possible choices.

10.2.1.2.3 Multi-feature DTW (MF-DTW) similarity matrix

Ones the proper FS is defined, it is time to further analyse the problem. To do so, we assume feature dependence, which seems to fit better the problem. While the DTW SM solves the issues linked to possible distortions or shifts of the same behaviour conducted by various temporal signatures, it needs to be further studied while considering a MF space. One solution would be to define a MF SM as the summation of SMs computed independently in each feature. Nevertheless, the MFs composing the temporal signatures are not independent. The idea is to arrange the DTW SM to a multi-dimensional space exploiting the features' dependence. To do so, the summation of the absolute difference between couples of samples is considered.

10.2.1.2.4 Similarity Analysis

The similarity analysis consists of computing a similarity threshold (Ω) for each class, which allows to understand what is the class similarity value for separating the CPs without incurring in incorrect classification of temporal signatures. The strategy models the MDDTW similarity distributions between groups of similarity measures and computes a Ω for each couple of classes in C. Jointly with SM^{MF}, two input parameters are required to carry out the task, likewise in the strategy presented in the subsection (10.2.1.2.2). However, in this stage, the two parameters $P_{CC} = [P_A, P_B]$ are known, since they have been calculated in the feature space analysis step.

The main ingredient here is the SM^{MF}, which is calculated in the reduced MF space R. The similarity distribution comparison is translated into an MF space and $\Omega_v^{(i,j)}$ are computed. The thresholds point the ideal similarity value to avoid confusion between the classes in the CPs generation step.

10.2.1.2.5 Class Prototypes (CP) generation

The class prototypes generation stage deals with class training samples clustering and the generation of CPs that depict the behaviour of a group of temporal signatures. The classes are handled individually, considering only the similarities between same class training temporal signatures. Starting from the SM^{MF}, containing the similarity measures between all the possible couples of training samples in the MF space, it is possible to extract the similarity matrix SM_i^{MF} belonging to the generic class c_i. To do so, a hierarchical clustering strategy is followed [108]. The clustering algorithm is dependent on the insertion order of samples inside the groups. A sampling order is defined based on simple rules, driven by the similarity index S_{idx}. S_{idx} is the dynamic value that starts as the maximum similarity value and decreases until convergence. Each class sample v (where $1 \leq v \leq V_i$) determines a column vector in SM_i^{MF}. The number of elements of the vector greater than S_{idx} indicates how many correspondences Y_v the sample v has with respect to other samples. The mean value of the elements that find correspondence for the sample v is μ_v. The samples are descending ordered for Y_v and μ_v, resulting in an ordered vector of samples. This sorting allows inserting first the samples that are most representative for the class and then the less representative or possible outliers.

The clustering process takes as input SM_i^{MF} and follows the prearranged order, inserting the samples in the clusters. The first sample intuitively enters and creates a new group. The following samples are examined with respect to the available clusters Θ_i, evaluating the number of cluster elements more similar than S_{idx}. Let us assume that the present clusters are $\Theta_i = \{\Theta_1^i, \Theta_2^i, \dots, \Theta_Q^i\}$. The mean value of similarity between the new sample and all the elements of the cluster q is μ_v^q (with $1 \leq q \leq Q$). A sample can enter the cluster q if Y_v^q is larger than a fixed threshold (Y_{TH}). In the case of multiple suitable clusters, the one with the higher μ_v^q is chosen. If no clusters allow the new sample entrance, the sample forms a new cluster. The operation is repeated until all the samples are clustered.

The result is a set of CP^i that depict the behaviours of the clusters Θ_i. The CP generation is performed for all the classes. This way of clustering ensures the generation of multiple prototypes dealing with multiple inner-class behaviours, and ensuring the recognition of different LC trends belonging to the same class.

10.2.1.3 LC classification and change detection (CD).

The goal here is to evaluate the similarity between known class trends (CPs found in previous step) and unknown temporal signatures. This is done in the reduced MF space R. Figure 29 shows the block scheme of the LC classification and CD approach, where a time division strategy is defined in order to have a controlled way to compare temporal sequences with minor length temporal signatures shifted in time. The MF-DTW compares the temporal sequences associated with time-order intervals and the available CPs in an MF space. The objective of the comparison is to derive informative similarity trends (STs) that describe the evolution of the pixel LC to highlight the similarity between unknown pixel behaviour and known classes behaviours. The STs drive the identification of imposing class in the defined time intervals and a stability correction strategy produces a sequence of LC classes, which depict the pixel LC evolution. The CD step identifies the LC class variation to derive a CD sequence that highlights the change between classes.

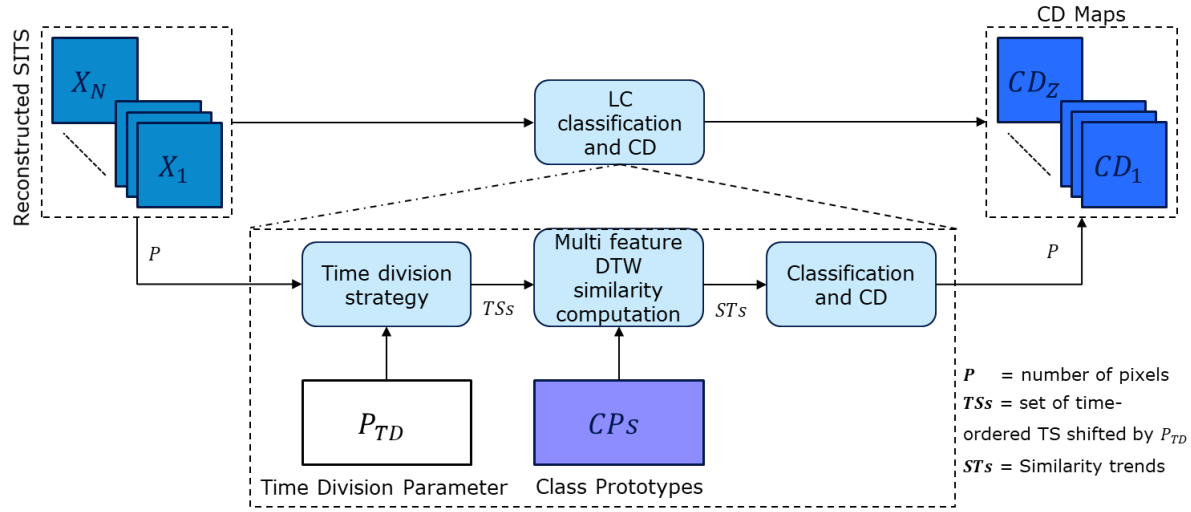


Figure 29. Block-based representation of LC classification and CD strategy.

10.2.1.3.1 Time division strategy

The time-division strategy step (see Figure 30) implements an approach for the analysis of pixel temporal signatures that aims at understanding the behaviour of the land response associated with a pixel in the period $[t_1, t_N]$. It is convenient to discretize the long time interval into periods with shorter duration to compare the temporal signatures with the known CPs. The inputs are the temporal signatures linked to a known spatial position in the interval $[t_1, t_N]$ and the time division parameter P_{TD} that drives the time-division scheme required to discretize the period $[t_1, t_N]$ under analysis. The temporal signatures linked to successive time intervals are extracted and time-shifted to produce a set of ordered temporal signatures $\{TS_p^1, TS_p^2, \dots, TS_p^Z\}$. Every TS_p^z ($1 \leq z \leq Z$) has a fixed length T_{stab} and a time starting point uniform with the CPs. The amount of ordered temporal signatures is equal to:

$$Z = \frac{time(t_N - t_1) - T_{stab} + P_{TD}}{P_{TD}} \quad (2)$$

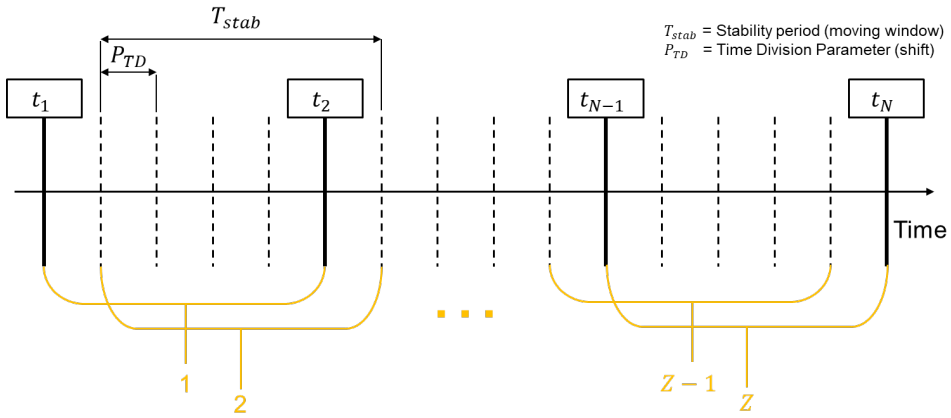


Figure 30. Block-based representation of time division strategy.

10.2.1.3.2 Multiple-feature DTW similarity computation

The obtained temporal signatures (from previous time division strategy) are compared with all the available CPs in order to obtain a SM for each couple (TS_p^z, CP). Given a temporal signature TS_p^z and a CP_q^i, it is possible to compute the MF-DTW similarity measure $S_{p^z}^{i_q}$.

$$S_{p^z}^{i_q} = MFDTW(TS_p^z, CP_q^i) \tag{3}$$

The comparison between TS_p^z and the Q CPs of the i class results in a vector of similarity measures $S_{p^z}^i$. The maximum value of the $S_{p^z}^i$ is considered as the similarity measure between TS_p^z and c_i. The operation is repeated for all the L classes to generate a similarity vector S_p^z that contains the similarity measures between the temporal signature and each of the L classes. The STs describe how the pixel response is related to the set of classes defined by the prototypes originated by the training set temporal signatures.

10.2.1.3.3 LC classification and CD

The analysis of the similarity between pixel temporal signatures and classes arises concerns about correct classification and detection of stable changes. The STs of the pixel p is a matrix where the rows are the L STs evolving in the Z (time intervals) columns:

$$ST_p = \begin{bmatrix} S_{p^1}^1 & S_{p^2}^1 & \cdots & S_{p^Z}^1 \\ S_{p^1}^2 & S_{p^2}^2 & & S_{p^Z}^2 \\ \vdots & & \ddots & \vdots \\ S_{p^1}^L & S_{p^2}^L & \cdots & S_{p^Z}^L \end{bmatrix} \tag{4}$$

A strategy identifies the imposing class and possible transitions between different classes. The highest value in a generic column ST_{p^z}, determines the most similar class to the pixel p temporal signature in the time interval z. The identification of the largest elements for each column builds a sequence of classes. The vector of classes could be affected by "false" transitions. The temporal signatures characterized by a class alteration may be influenced by the presence of a variable-length time interval where the class with the highest similarity is not the "true" LC. It is important to correct potential non-stable changes and produce a sequence of stable classes to detect the variations. A simple rule adjusts the classes sequence. A generic class is considered stable if it persists as the imposing class for a fixed number of time intervals TI_s. The rule allows to discard the presence of "false" transitions and imposes a stability condition for the emerging classes. The output is a sequence of classes LC_p dependent on the STs and reviewed by a stability rule. The fixed number TI_s drives the stability condition and is dependent on the ratio T_{stab}/P_{TD}. The last step of the method provides the generation of a sequence of CD maps, highlighting the LC changes from a class to another. The algorithm searches for class variation in LC_p

and reports it as a sequence of stability (0 value) or variation (h value). The h value is determined by the order dependent couple of classes in the transition. The set $\{g_1, g_2, \dots, g_H\}$ are the possible values, where $H = L \times (L-1)$ is the number of possible LC changes. The actual number of relevant changes might differ from H according to the user/climate requirements.

10.3 Seasonal changes detection

When required (according to the type of change) and possible (enough images available in one year), composite maps will be built by adapting function fitting methods such as the one used in the TIMESAT program [105] in order to derive seasonal changes. If the type of change is related to plant/crop phenology, other approaches to build continuous information based on vegetation indices can be used [109].

In this case, the analysis will be fully supervised, where the user will request information for the LC of interest. The analysis will be carried out at the level of vegetation indices. Therefore, possible setup of a set of thresholds as per known ranges of change/disturbance might be required from the user. As an alternative, such thresholds can be defined by us in an unsupervised way, allowing the user to know if there has been some disturbance/change, but not the type of disturbance/change.

10.4 A deep learning perspective

The processing chain for the multitemporal CD and trend analysis could be also analyzed from a Deep Learning (DL) perspective (Figure 31). In particular, some works [110]–[114] can be found nowadays in literature that deal with rather longer time scale changes or inter-annual changes. The main problem for deep learning approaches remains the lack of enough training samples to train the algorithms. This problem is even bigger when we talk about CD and long time series. When training samples are available, the potential in terms of accuracy is quite remarkable. Some examples of works carried out in literature, rather in Landsat like data or long time series, are provided in the next in order to show the potential of DL. Inspiration could be taken from these works in order to be applied on the CCI HRLC with some extra work for training data collection.

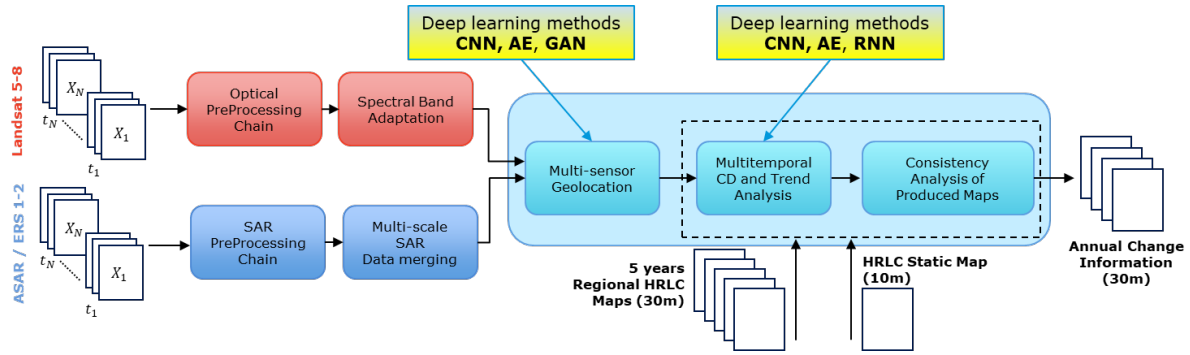


Figure 31. New deep learning block-based representation of the processing chain for the multitemporal change detection and trend analysis.

10.4.1 Learning a Transferable Change Rule from a Recurrent Neural Network (RNN) for Land Cover Change Detection (REFEREE).

The goal of this work is to design an efficient transferable change rule for binary and multi-class CD. To do so, the method relies on an improved Long Short-Term Memory (LSTM) model, in a RNN learning framework, that acquires and records the change information of long-term sequences of remote sensing data. Experiments were carried out in three different datasets/cities (Taizhou, Kunshan and Yancheng in China), with different types of changes. The results of REFEREE were compared with non-deep learning approaches such as Change Vector Analysis (CVA), Principal Component Analysis (PCA), Iteratively-Reweighted Multivariate Alteration Detection (IRMAD) and Supervised Slow Feature Analysis (SSFA). The results, summarized in Table 8, show the high

potential of REFEREE over standard methods with an increase of accuracy over 10-30% for the binary CD case and over 10-25% for the multiple CD case.

Table 8. Kappa coefficient and Overall Accuracy (OA) for the three datasets in (a) binary and (b) multiple change detection cases.

	TaiZhou		KunShan		Yancheng	
	KAPPA	OA	KAPPA	OA	KAPPA	OA
CVA	0.3755	0.6982	0.4011	0.7160	0.7907	0.8722
PCA	0.5413	0.7419	0.633	0.7741	0.8174	0.9025
IRMAD	0.7942	0.9133	0.87	0.9397	0.6973	0.8352
SSFA	0.8229	0.9454	0.9361	0.9763	0.9032	0.9516
REFEREE	**0.9477**	**0.9777**	**0.9573**	**0.9837**	**0.9563**	**0.9828**

(a)

		OA	Kappa	F-score				
				Unchanged	City (C)	Water (C)	Soil (C)	Farmland (C)
Taizhou	REFEREE	0.95	0.8689	0.9788	0.7887	0.8749	0.7524	/
	CNN	0.9235	0.8063	0.9675	0.6679	0.8721	0.5521	/
	SVM	0.8391	0.6758	0.8717	0.5203	0.8326	0.3558	/
	Decision tree	0.7113	0.5221	0.8701	0.6403	0.7496	0.3558	/
Kunshan	REFEREE	0.9587	0.8988	0.9432	0.9735	/	/	0.8750
	CNN	0.9336	0.8413	0.8844	0.9559	/	/	0.8491
	SVM	0.8024	0.6654	0.6830	0.8762	/	/	0.3743
	Decision tree	0.6979	0.4844	0.6642	0.7913	/	/	0.1542

(b)

10.4.2 Forest Change Detection in Incomplete Satellite Images with Deep Neural Network.

The goal of this work is to detect forest cover changes (deforestation and fire) over a period of 29 years (1987-2015). The study area is located in Australia and Landsat images are used. This is the closest example to what we will face in the CCI HRLC project, both in time span and data type. Given the well-known problem of incomplete and contaminated Landsat data, this approach includes the pre-processing steps as well, which are not addressed with deep learning approaches. The CD problem is addressed as a classification problem itself, where features are learnt using a deep neural network in a data-driven fashion. Based on these highly discriminative representations, it is possible to determine forest changes and predict their onset and offset timings. Results are compared to state-of-the-art approaches such Support Vector Machines (SVM), Random Forest (RF), Bag of visual Words (BoW) and Scale Invariant Feature Transform (SIFT). The proposed approach in this paper showed an improvement of about 16-24% for the forest changes (see Table 9) and a mean onset/offset prediction error of 4.9months (an error reduction of five months – see Table 9 and Figure 32).

Table 9. Example of classification/change detection and onset/offset detection. Accuracies are given in percentage, whereas the error units are months.

Method	Accuracy	Avg. Recall	Onset Error (Mn)	Offset Error (Mn)
SIFT+l-SVM	68.1	57.3	8.7 ± 4.1	15.1 ± 7.5
SIFT+k-SVM	71.3	61.4	8.3 ± 4.1	14.9 ± 7.2
SIFT+RF	69.7	58.8	8.9 ± 4.3	15.9 ± 7.7
BoW+l-SVM	72.6	63.1	7.4 ± 3.6	13.5 ± 6.9
BoW+k-SVM	74.1	64.9	7.1 ± 3.4	12.6 ± 6.8
BoW + RF	71.7	64.0	7.4 ± 3.7	13.8 ± 7.1
This paper	**92.0**	**84.6**	**3.2 ± 2.3**	**5.5 ± 5.5**

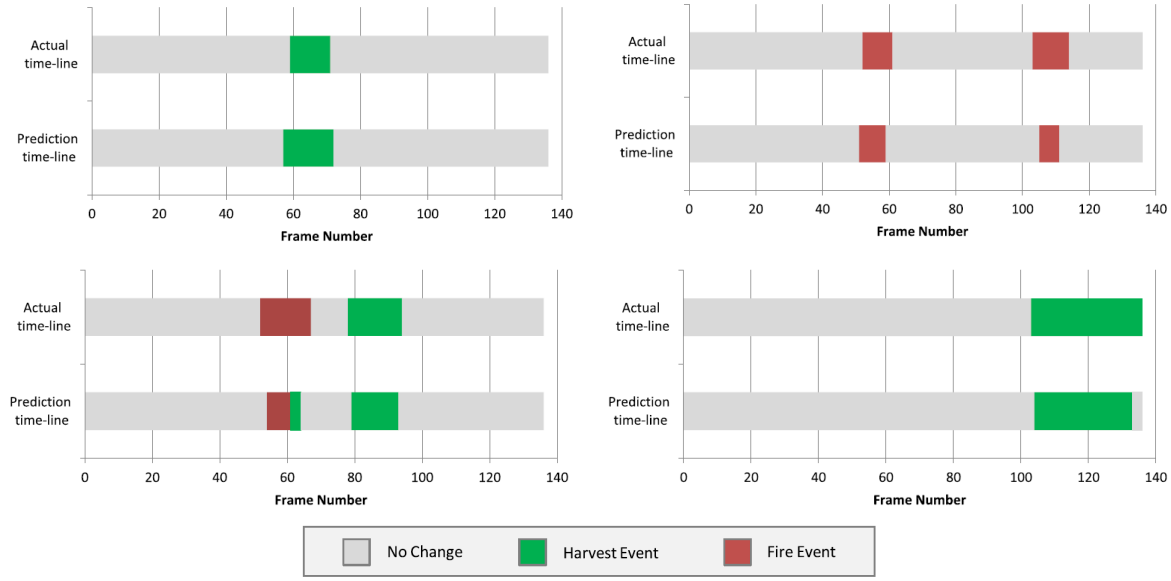


Figure 32. Sample result of the ground truth onset/offset events. In each plot, the top bar shows ground truth, and the bottom bar shows prediction from the proposed approach.

10.4.3 Long-Term Annual Mapping of Four Cities on Different Continents by Applying a Deep Information Learning Method to Landsat Data

The goal of [113] is to detect long-term urban changes by addressing temporal spectral variance and a scarcity of training samples in Landsat images from 1984-2016. Once again, we are in a similar situation to the CCI HRLC project. This time the focus is on urban changes, and not on vegetation LC like, which is indeed complementary to the paper presented in section 10.4.2. The method is applied to Landsat observations over urban areas in four cities in the temperate zone (Beijing, New York, Melbourne, and Munich). The method is trained using observations of Beijing collected in 1999, and then used to map urban areas in all target cities for the entire 1984–2016 period. The method uses two main steps: (1) use of RNN to minimize seasonal urban spectral variance; and (2) introduce an automated transfer strategy to maximize information gain from limited training samples when applied to new target cities in similar climate zones. The method is compared to other state-of-the-art methods (SVM, RF and RNN-LSTM), achieving comparable or even better accuracies (see Table 10). The overall accuracy of single-year urban maps is approximately 96±3% among the four target cities.

Table 10. Detection results from state-of-the-art methods and proposed method with OA and run-time.

		SVM-RBF (%)	RF (%)	RNN-LSTM (%)	Proposed Framework (%)
Temporal transfer	Beijing	68.63	71.38	76.25	81.87
Spatial transfer	New York	69.13	72.75	80.63	82.08
	Melbourne	71.25	67.63	85.88	84.75
	Munich	79.25	78.2	86.87	90.63
Run-Time (min)	-	7.53	0.37	0.78	0.82

11 References

[1] R. Richter and D. Schlapfer, “Atmospheric / Topographic Correction for Satellite Imagery (ATCOR-2/3 UserGuide, Version 9.0.2, March 2016).”

[2] B. Mayer and A. Kylling, “The libRadtran software package for radiative transfer calculations - description and examples of use,” *Atmos. Chem. Phys.*, vol. 5, pp. 1855–1877, 2005.

[3] J. G. Masek *et al.*, “A Landsat surface reflectance dataset for North America, 1990-2000,” *IEEE Geoscience and Remote Sensing Letters*, vol. 3, no. 1, pp. 68–72, Jan. 2006, doi: 10.1109/LGRS.2005.857030.

[4] E. Vermote, C. Justice, M. Claverie, and B. Franch, “Preliminary analysis of the performance of the Landsat 8/OLI land surface reflectance product,” *Remote Sensing of Environment*, vol. 185, pp. 46–56, Nov. 2016, doi: 10.1016/j.rse.2016.04.008.

[5] Z. Zhu and C. E. Woodcock, “Object-based cloud and cloud shadow detection in Landsat imagery,” *Remote Sensing of Environment*, vol. 118, pp. 83–94, Mar. 2012, doi: 10.1016/j.rse.2011.10.028.

[6] Z. Zhu, S. Wang, and C. E. Woodcock, “Improvement and expansion of the Fmask algorithm: cloud, cloud shadow, and snow detection for Landsats 4–7, 8, and Sentinel 2 images,” *Remote Sensing of Environment*, vol. 159, pp. 269–277, Mar. 2015, doi: 10.1016/j.rse.2014.12.014.

[7] X. Zhu, F. Gao, D. Liu, and J. Chen, “A Modified Neighborhood Similar Pixel Interpolator Approach for Removing Thick Clouds in Landsat Images,” *IEEE Geoscience and Remote Sensing Letters*, vol. 9, no. 3, pp. 521–525, May 2012, doi: 10.1109/LGRS.2011.2173290.

[8] M. Xu, X. Jia, M. Pickering, and A. J. Plaza, “Cloud Removal Based on Sparse Representation via Multitemporal Dictionary Learning,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 5, pp. 2998–3006, May 2016, doi: 10.1109/TGRS.2015.2509860.

[9] J. Chen, X. Zhu, J. E. Vogelmann, F. Gao, and S. Jin, “A simple and effective method for filling gaps in Landsat ETM+ SLC-off images,” *Remote Sensing of Environment*, vol. 115, no. 4, pp. 1053–1064, Apr. 2011, doi: 10.1016/j.rse.2010.12.010.

[10] J. Heo and T. W. FitzHugh, “A standardized radiometric normalization method for change detection using remotely sensed imagery,” *PHOTOGRAM ENG REMOTE SENS*, vol. 66, no. 2, pp. 173–181, Feb. 2000.

[11] B. Chen, B. Huang, L. Chen, and B. Xu, “Spatially and Temporally Weighted Regression: A Novel Method to Produce Continuous Cloud-Free Landsat Imagery,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 1, pp. 27–37, Jan. 2017, doi: 10.1109/TGRS.2016.2580576.

[12] A. V. Egorov, D. P. Roy, H. K. Zhang, M. C. Hansen, and A. Kommareddy, “Demonstration of Percent Tree Cover Mapping Using Landsat Analysis Ready Data (ARD) and Sensitivity with Respect to Landsat ARD Processing Level,” *Remote Sensing*, vol. 10, no. 2, p. 209, Feb. 2018, doi: 10.3390/rs10020209.

[13] “Generation of Homogeneous VHR Time Series by Nonparametric Regression of Multisensor Bitemporal Images - IEEE Journals & Magazine.” [Online]. Available: https://ieeexplore.ieee.org/document/8726352. [Accessed: 10-Dec-2019].

[14] M. Claverie *et al.*, “The Harmonized Landsat and Sentinel-2 surface reflectance data set,” *Remote Sensing of Environment*, vol. 219, pp. 145–161, Dec. 2018, doi: 10.1016/j.rse.2018.09.002.

[15] “Developer Guide - SNAP - SNAP Wiki.” [Online]. Available: https://senbox.atlassian.net/wiki/spaces/SNAP/pages/8847381/Developer+Guide. [Accessed: 10-Dec-2019].

[16] C. Oliver and S. Quegan, *Understanding synthetic aperture radar images*. Boston: Artech House, 1998.

[17] F. Argenti, A. Lapini, T. Bianchi, and L. Alparone, “A Tutorial on Speckle Reduction in Synthetic Aperture Radar Images,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 1, no. 3, pp. 6–35, Sep. 2013, doi: 10.1109/MGRS.2013.2277512.

[18] F. Argenti, A. Lapini, T. Bianchi, and L. Alparone, “A Tutorial on Speckle Reduction in Synthetic Aperture Radar Images,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 1, no. 3, pp. 6–35, Sep. 2013, doi: 10.1109/MGRS.2013.2277512.

[19] J.-S. Lee, “Speckle analysis and smoothing of synthetic aperture radar images,” *Computer Graphics and Image Processing*, vol. 17, no. 1, pp. 24–32, Sep. 1981, doi: 10.1016/S0146-664X(81)80005-6.

[20] P. Kupidura, “Comparison of Filters Dedicated to Speckle Suppression in SAR Images,” *ISPAr*, vol. 41B7, pp. 269–276, Jun. 2016, doi: 10.5194/isprs-archives-XLI-B7-269-2016.

[21] F. Qiu, J. Berglund, J. R. Jensen, P. Thakkar, and D. Ren, "Speckle Noise Reduction in SAR Imagery Using a Local Adaptive Median Filter," *GIScience & Remote Sensing*, vol. 41, no. 3, pp. 244–266, Sep. 2004, doi: 10.2747/1548-1603.41.3.244.

[22] W. Zhao, C.-A. Deledalle, L. Denis, H. Maître, J.-M. Nicolas, and F. Tupin, "Ratio-Based Multitemporal SAR Images Denoising: RABASAR," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 6, pp. 3552–3565, Jun. 2019, doi: 10.1109/TGRS.2018.2885683.

[23] G. Quin, B. Pinel-Puyssegur, J.-M. Nicolas, and P. Loreaux, "MIMOSA: An Automatic Change Detection Method for SAR Time Series," *IEEE Trans. Geosci. Remote Sensing*, vol. 52, no. 9, pp. 5349–5363, Sep. 2014, doi: 10.1109/TGRS.2013.2288271.

[24] A. Salentinig and P. Gamba, "A General Framework for Urban Area Extraction Exploiting Multiresolution SAR Data Fusion," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 5, pp. 2009–2018, May 2016, doi: 10.1109/JSTARS.2016.2546553.

[25] "Adapted Wavelet Analysis: From Theory to Software," *CRC Press*. [Online]. Available: https://www.crcpress.com/Adapted-Wavelet-Analysis-From-Theory-to-Software/Wickerhauser/p/book/9780367448608. [Accessed: 10-Dec-2019].

[26] S. G. Mallat, "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, no. 7, pp. 674–693, Jul. 1989, doi: 10.1109/34.192463.

[27] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 4 edition. New York, NY: Pearson, 2017.

[28] M. Vetterli and C. Herley, "Wavelets and filter banks: theory and design," *IEEE Transactions on Signal Processing*, vol. 40, no. 9, pp. 2207–2232, Sep. 1992, doi: 10.1109/78.157221.

[29] G. Simone, F. C. Morabito, and A. Farina, "Radar image fusion by multiscale Kalman filtering," in *Proceedings of the Third International Conference on Information Fusion*, 2000, vol. 2, p. WED3/10-WED3/17 vol.2, doi: 10.1109/IFIC.2000.859858.

[30] L. G. Brown and L. Gottesfeld, "A survey of image registration techniques," *ACM Computing Surveys*, vol. 24, no. 4, pp. 325–376, Dec. 1992, doi: 10.1145/146370.146374.

[31] A. A. Goshtasby, "Fusion of multi-exposure images," *Image and Vision Computing*, vol. 23, no. 6, pp. 611–618, Jun. 2005, doi: 10.1016/j.imavis.2005.02.004.

[32] H. Bay, T. Tuytelaars, and L. Van Gool, "SURF: Speeded Up Robust Features," in *Computer Vision – ECCV 2006*, Berlin, Heidelberg, 2006, pp. 404–417, doi: 10.1007/11744023_32.

[33] C. Harris and M. Stephens, "A combined corner and edge detector," in *In Proc. of Fourth Alvey Vision Conference*, 1988, pp. 147–151.

[34] M. Donoser and H. Bischof, "Efficient Maximally Stable Extremal Region (MSER) Tracking," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2006, vol. 1, pp. 553–560, doi: 10.1109/CVPR.2006.107.

[35] R. B. Ash, *Information Theory*. Courier Corporation, 1990.

[36] J. Le Moigne, N. S. Netanyahu, and R. D. Eastman, Eds., *Image Registration for Remote Sensing*. Cambridge: Cambridge University Press, 2011.

[37] A. A. Cole-Rhodes, K. L. Johnson, J. LeMoigne, and L. Zavorin, "Multiresolution registration of remote sensing imagery by optimization of mutual information using a stochastic gradient," *IEEE Transactions on Image Processing*, vol. 12, no. 12, pp. 1495–1510, 2003, doi: 10.1109/TIP.2003.819237.

[38] D. Solarna, A. Gotelli, J. Le Moigne, G. Moser, and S. B. Serpico, "Crater Detection and Registration of Planetary Images through Marked Point Processes, Multiscale Decomposition, and Region-Based Analysis," *TGRS*, Accepted up to minor modifications.

[39] J. Le Moigne, N. S. Netanyahu, and R. D. Eastman, Eds., *Image Registration for Remote Sensing*. Cambridge: Cambridge University Press, 2011.

[40] B. Zitová and J. Flusser, "Image registration methods: a survey," *Image and Vision Computing*, vol. 21, no. 11, pp. 977–1000, Oct. 2003, doi: 10.1016/S0262-8856(03)00137-9.

[41] D. G. Lowe, D. G. Lowe, and D. G. Lowe, "Object Recognition from Local Scale-Invariant Features," in *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, Washington, DC, USA, 1999, pp. 1150–.

[42] R. O. Duda and P. E. Hart, “Use of the Hough transformation to detect lines and curves in pictures,” *Communications of the ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972, doi: 10.1145/361237.361242.

[43] X. Descombes, R. Minlos, and E. Zhizhina, “Object Extraction Using a Stochastic Birth-and-Death Dynamics in Continuum,” *Journal of Mathematical Imaging and Vision*, vol. 33, no. 3, pp. 347–359, Mar. 2009, doi: 10.1007/s10851-008-0117-y.

[44] A. B. Carlson and P. Crilly, *Communication Systems*, 5 edizione. Boston: McGraw-Hill Education, 2009.

[45] E. Parzen, “On Estimation of a Probability Density Function and Mode,” *The Annals of Mathematical Statistics*, vol. 33, no. 3, pp. 1065–1076, 1962.

[46] J. M. Murphy, J. Le Moigne, and D. J. Harding, “Automatic Image Registration of Multimodal Remotely Sensed Data With Global Shearlet Features,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 3, pp. 1685–1704, Mar. 2016, doi: 10.1109/TGRS.2015.2487457.

[47] M. J. D. Powell, “An efficient method for finding the minimum of a function of several variables without calculating derivatives,” *Comput J*, vol. 7, no. 2, pp. 155–162, Jan. 1964, doi: 10.1093/comjnl/7.2.155.

[48] R. P. Brent, “An algorithm with guaranteed convergence for finding a zero of a function,” *Comput J*, vol. 14, no. 4, pp. 422–425, Jan. 1971, doi: 10.1093/comjnl/14.4.422.

[49] M. J. D. Powell, “A Direct Search Optimization Method That Models the Objective and Constraint Functions by Linear Interpolation,” in *Advances in Optimization and Numerical Analysis*, S. Gomez and J.-P. Hennart, Eds. Dordrecht: Springer Netherlands, 1994, pp. 51–67.

[50] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. .

[51] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, “Adversarial Autoencoders,” Nov. 2015.

[52] N. Merkle, S. Auer, R. Muller, and P. Reinartz, “Exploring the Potential of Conditional Adversarial Networks for Optical and SAR Image Matching,” *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 6, pp. 1811–1820, Jun. 2018, doi: 10.1109/JSTARS.2018.2803212.

[53] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, “Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 645–657, Feb. 2017, doi: 10.1109/TGRS.2016.2612821.

[54] L. Breiman, “Bagging Predictors,” *Machine Learning*, vol. 24, no. 2, pp. 123–140, Aug. 1996, doi: 10.1023/A:1018054314350.

[55] P. O. Gislason, J. A. Benediktsson, and J. R. Sveinsson, “Random Forests for Land Cover Classification,” *Pattern Recogn. Lett.*, vol. 27, no. 4, pp. 294–300, Mar. 2006, doi: 10.1016/j.patrec.2005.08.011.

[56] J. A. Benediktsson and I. Kanellopoulos, “Classification of multisource and hyperspectral data based on decision fusion,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 3, pp. 1367–1377, May 1999, doi: 10.1109/36.763301.

[57] S. Fukuda and H. Hirosawa, “Support vector machine classification of land cover: application to polarimetric SAR data,” in *IGARSS 2001. Scanning the Present and Resolving the Future. Proceedings. IEEE 2001 International Geoscience and Remote Sensing Symposium (Cat. No.01CH37217)*, 2001, vol. 1, pp. 187–189 vol.1, doi: 10.1109/IGARSS.2001.976097.

[58] R. S. Hosseini, I. Entezari, S. Homayouni, M. Motagh, and B. Mansouri, “Classification of polarimetric SAR images using Support Vector Machines,” *Canadian Journal of Remote Sensing*, vol. 37, no. 2, pp. 220–233, Nov. 2011, doi: 10.5589/m11-029.

[59] P. Mantero, G. Moser, and S. B. Serpico, “Partially Supervised classification of remote sensing images through SVM-based probability density estimation,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 559–570, Mar. 2005, doi: 10.1109/TGRS.2004.842022.

[60] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Nov. 2011.

[61] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, Oct. 1986, doi: 10.1038/323533a0.

[62] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, “Locality-constrained Linear Coding for image classification,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3360–3367, doi: 10.1109/CVPR.2010.5540018.

[63] F. Perronnin, J. Sánchez, and T. Mensink, “Improving the Fisher Kernel for Large-Scale Image Classification,” in *Computer Vision – ECCV 2010*, Berlin, Heidelberg, 2010, pp. 143–156, doi: 10.1007/978-3-642-15561-1_11.

[64] H. Jégou, M. Douze, C. Schmid, and P. Pérez, “Aggregating local descriptors into a compact image representation,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 3304–3311, doi: 10.1109/CVPR.2010.5540039.

[65] F. Hu, G.-S. Xia, J. Hu, and L. Zhang, “Transferring Deep Convolutional Neural Networks for the Scene Classification of High-Resolution Remote Sensing Imagery,” *Remote Sensing*, vol. 7, no. 11, pp. 14680–14707, Nov. 2015, doi: 10.3390/rs71114680.

[66] E. Dinerstein *et al.*, “An Ecoregion-Based Approach to Protecting Half the Terrestrial Realm,” *Bioscience*, vol. 67, no. 6, pp. 534–545, Jun. 2017, doi: 10.1093/biosci/bix014.

[67] S. Abdikan, F. B. Sanli, M. Ustuner, and F. Calò, “Land Cover Mapping Using SENTINEL-1 SAR Data,” *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 41B7, pp. 757–761, Jun. 2016, doi: 10.5194/isprs-archives-XLI-B7-757-2016.

[68] S. Niculescu, H. Talab Ou Ali, and A. Billey, “Random forest classification using Sentinel-1 and Sentinel-2 series for vegetation monitoring in the Pays de Brest (France),” in *Remote Sensing for Agriculture, Ecosystems, and Hydrology XX*, Berlin, Germany, 2018, p. 6, doi: 10.1117/12.2325546.

[69] “ERDAS Field Guide,” 03-Mar-2016. [Online]. Available: https://community.hexagongeospatial.com/t5/IMAGINE-Q-A/ERDAS-Field-Guide/ta-p/3179. [Accessed: 10-Dec-2019].

[70] W. K. Pratt, *Digital Image Processing: PIKS Scientific Inside*, 4 edition. Hoboken, N.J: Wiley-Interscience, 2007.

[71] J. S. Lee, L. Jurkevich, P. Dewaele, P. Wambacq, and A. Oosterlinck, “Speckle filtering of synthetic aperture radar images: A review,” *Remote Sensing Reviews*, vol. 8, no. 4, pp. 313–340, Feb. 1994, doi: 10.1080/02757259409532206.

[72] J. S. Lee and E. Pottier, *Polarimetric Radar Imaging: from Basics to Applications*, 1st ed. FL, USA: CRC Press: Boca Raton, 2009.

[73] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou, “A tutorial on synthetic aperture radar,” *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 1, pp. 6–43, Mar. 2013, doi: 10.1109/MGRS.2013.2248301.

[74] N. Yokoya, “Texture-Guided multisensor superresolution for remotely sensed images,” *Remote Sens.*, vol. 9, p. 316, 2017.

[75] H. Jingliang, P. Gamisi, and X. Zhu, “Feature extraction and selection of sentinel-1 dual-pol data for global-scale local climate zone classification,” *ISPRS International Journal of Geo-Information*, vol. 7, no. 9, p. 379, 2018.

[76] A. Braun and V. Hochschild, “Combined use of SAR and optical data for environmental assessments around refugee camps in semiarid landscapes,” *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 7, pp. 777–782, Apr. 2015, doi: 10.5194/isprsarchives-XL-7-W3-777-2015.

[77] P. Du, A. Samat, B. Waske, S. Liu, and Z. Li, “Random Forest and Rotation Forest for fully polarized SAR image classification using polarimetric and spatial features,” *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 105, pp. 38–53, Jul. 2015, doi: 10.1016/j.isprsjprs.2015.03.002.

[78] M. Wurm, H. Taubenböck, M. Weigand, and A. Schmitt, “Slum mapping in polarimetric SAR data using spatial features,” *Remote Sensing of Environment*, vol. 194, pp. 190–204, Jun. 2017, doi: 10.1016/j.rse.2017.03.030.

[79] F. N. Numbisi, F. V. Coillie, and R. D. Wulf, “MULTI-DATE SENTINEL1 SAR IMAGE TEXTURES DISCRIMINATE PERENNIAL AGROFORESTS IN A TROPICAL FOREST-SAVANNAH TRANSITION LANDSCAPE,” in *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 2018, vol. XLII–1, pp. 339–346, doi: https://doi.org/10.5194/isprs-archives-XLII-1-339-2018.

[80] O. Cartus, M. Santoro, C. Schmullius, P. Y. Yong, C. Er-xue, and L. Zeng-yuan, “CREATION OF LARGE AREA FOREST BIOMASS MAPS FOR NORTHEAST CHINA USING ERS-1 / 2 TANDEM COHERENCE,” 2007.

[81] N.-W. Park and K.-H. Chi, "Integration of multitemporal/polarization C-band SAR data sets for land-cover classification," *International Journal of Remote Sensing*, vol. 29, no. 16, pp. 4667–4688, Aug. 2008, doi: 10.1080/01431160801947341.

[82] "Land Cover Classification System - Classification concepts and user manual." [Online]. Available: http://www.fao.org/3/y7220e/y7220e00.htm. [Accessed: 10-Dec-2019].

[83] T. Kobayashi *et al.*, "Production of Global Land Cover Data – GLCNMO2013," *Journal of Geography and Geology*, vol. 9, no. 3, p. p1, Jun. 2017, doi: 10.5539/jgg.v9n3p1.

[84] R. Tateishi *et al.*, "Production of global land cover data – GLCNMO," *International Journal of Digital Earth*, vol. 4, no. 1, pp. 22–49, Jan. 2011, doi: 10.1080/17538941003777521.

[85] R. Tateishi, N. Hoan, T. Kobayashi, B. Alsaaideh, G. Tana, and D. Phong, "Production of Global Land Cover Data – GLCNMO2008," *Journal of Geography and Geology*, vol. 6, no. 3, p. p99, Jul. 2014, doi: 10.5539/jgg.v6n3p99.

[86] G. Lisini, A. Salentinig, P. Du, and P. Gamba, "SAR-Based Urban Extents Extraction: From ENVISAT to Sentinel-1," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 11, no. 8, pp. 2683–2691, Aug. 2018, doi: 10.1109/JSTARS.2017.2782180.

[87] J. A. Benediktsson, "Hybrid consensus theoretic classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 35, no. 4, pp. 833–843, Jul. 1997, doi: 10.1109/36.602526.

[88] J. A. Benediktsson and P. H. Swain, "Consensus theoretic classification methods," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 4, pp. 688–704, 1992, doi: 10.1109/21.156582.

[89] N. C. Dalkey, "An Impossibility Theorem for Group Probability Functions.," 1972.

[90] J. A. Benediktsson and I. Kanellopoulos, "Classification of multisource and hyperspectral data based on decision fusion," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 37, no. 3, pp. 1367–1377, May 1999, doi: 10.1109/36.763301.

[91] S. Z. Li, *Markov random field modeling in image analysis*. Springer, 2009.

[92] Z. Kato and J. Zerubia, "Markov Random Fields in Image Segmentation," *Foundations and Trends in Signal Processing*, vol. 5, no. 1–2, pp. 1–155, 2012, doi: 10.1561/2000000035.

[93] R. Szeliski *et al.*, "A Comparative Study of Energy Minimization Methods for Markov Random Fields with Smoothness-Based Priors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 6, pp. 1068–1080, Jun. 2008, doi: 10.1109/TPAMI.2007.70844.

[94] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1222–1239, 2001, doi: 10.1109/34.969114.

[95] N. Kussul, M. Lavreniuk, S. Skakun, and A. Shelestov, "Deep Learning Classification of Land Cover and Crop Types Using Remote Sensing Data," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 5, pp. 778–782, May 2017, doi: 10.1109/LGRS.2017.2681128.

[96] E. Maggiori, Y. Tarabalka, G. Charpiat, and P. Alliez, "Convolutional Neural Networks for Large-Scale Remote-Sensing Image Classification," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 2, pp. 645–657, Feb. 2017, doi: 10.1109/TGRS.2016.2612821.

[97] S. Ji *et al.*, "3D Convolutional Neural Networks for Crop Classification with Multi-Temporal Remote Sensing Images," *Remote Sensing*, vol. 10, no. 2, p. 75, Jan. 2018, doi: 10.3390/rs10010075.

[98] M. Campos-Taberner *et al.*, "Processing of Extremely High-Resolution LiDAR and RGB Data: Outcome of the 2015 IEEE GRSS Data Fusion Contest–Part A: 2-D Contest," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 9, no. 12, pp. 5547–5559, Dec. 2016, doi: 10.1109/JSTARS.2016.2569162.

[99] "2018 IEEE GRSS Data Fusion Contest Results - GRSS | IEEE | Geoscience & Remote Sensing Society." [Online]. Available: http://www.grss-ieee.org/community/technical-committees/data-fusion/2018-ieee-grss-data-fusion-contest-results/. [Accessed: 27-Jun-2019].

[100] "Potsdam 2D Semantic Labeling - ISPRS." [Online]. Available: http://www2.isprs.org/commissions/comm2/wg4/potsdam-2d-semantic-labeling.html. [Accessed: 27-Jun-2019].

[101] "Vaihingen 2D Semantic Labeling - ISPRS." [Online]. Available: http://www2.isprs.org/commissions/comm2/wg4/vaihingen-2d-semantic-labeling-contest.html. [Accessed: 27-Jun-2019].

[102] J. Verbesselt, R. Hyndman, G. Newnham, and D. Culvenor, "Detecting trend and seasonal changes in satellite image time series," *Remote Sensing of Environment*, vol. 114, no. 1, pp. 106–115, Jan. 2010, doi: 10.1016/j.rse.2009.08.014.

[103] K. Zhao *et al.*, "Detecting change-point, trend, and seasonality in satellite time series data to track abrupt changes and nonlinear dynamics: A Bayesian ensemble algorithm," *Remote Sensing of Environment*, vol. 232, p. 111181, Oct. 2019, doi: 10.1016/j.rse.2019.04.034.

[104] R. Saxena *et al.*, "Towards a polyalgorithm for land use change detection," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 144, pp. 217–234, Oct. 2018, doi: 10.1016/j.isprsjprs.2018.07.002.

[105] P. Jonsson and L. Eklundh, "Seasonality extraction by function fitting to time-series of satellite sensor data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 8, pp. 1824–1832, Aug. 2002, doi: 10.1109/TGRS.2002.802519.

[106] Y. T. Solano-Correa, F. Bovolo, L. Bruzzone, and D. Fernández-Prieto, "A Method for the Analysis of Small Crop Fields in Sentinel-2 Dense Time Series," *IEEE Transactions on Geoscience and Remote Sensing*, pp. 1–15, 2019, doi: 10.1109/TGRS.2019.2953652.

[107] M. Belgiu and O. Csillik, "Sentinel-2 cropland mapping using pixel-based and object-based time-weighted dynamic time warping analysis," *Remote Sensing of Environment*, vol. 204, pp. 509–523, Jan. 2018, doi: 10.1016/j.rse.2017.10.005.

[108] Y. T. Solano-Correa, F. Bovolo, and L. Bruzzone, "A Semi-Supervised Crop-Type Classification Based on Sentinel-2 NDVI Satellite Image Time Series And Phenological Parameters," in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, 2019, pp. 457–460, doi: 10.1109/IGARSS.2019.8897922.

[109] Y. T. Solano-Correa, F. Bovolo, L. Bruzzone, and D. Fernández-Prieto, "Automatic Derivation of Cropland Phenological Parameters by Adaptive Non-Parametric Regression of Sentinel-2 Ndvi Time Series," in *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium*, 2018, pp. 1946–1949, doi: 10.1109/IGARSS.2018.8519264.

[110] S. Saha, F. Bovolo, and L. Bruzzone, "Unsupervised Deep Change Vector Analysis for Multiple-Change Detection in VHR Images," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 57, no. 6, pp. 3677–3693, Jun. 2019, doi: 10.1109/TGRS.2018.2886643.

[111] S. H. Khan, X. He, F. Porikli, and M. Bennamoun, "Forest Change Detection in Incomplete Satellite Images With Deep Neural Networks," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 55, no. 9, pp. 5407–5423, Sep. 2017, doi: 10.1109/TGRS.2017.2707528.

[112] H. Lyu, H. Lu, and L. Mou, "Learning a Transferable Change Rule from a Recurrent Neural Network for Land Cover Change Detection," *Remote Sensing*, vol. 8, no. 6, p. 506, Jun. 2016, doi: 10.3390/rs8060506.

[113] H. Lyu *et al.*, "Long-Term Annual Mapping of Four Cities on Different Continents by Applying a Deep Information Learning Method to Landsat Data," *Remote Sensing*, vol. 10, no. 3, p. 471, Mar. 2018, doi: 10.3390/rs10030471.

[114] C. Pelletier, G. I. Webb, and F. Petitjean, "Temporal Convolutional Neural Network for the Classification of Satellite Image Time Series," *Remote Sensing*, vol. 11, no. 5, p. 523, Jan. 2019, doi: 10.3390/rs11050523.