Climate Change Initiative Extension (CCI+) Phase 2
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
LSCE
CREAF
University of Exeter
e-GEOS s.p.a.
Planetek Italia

Changelog

Issue	Changes	Date
1.0	First issue	25/10/2024
1.1	Revised according to RIDs	16/12/2025

Detailed Change Record

Issue	RID	Description of discrepancy	Sections	Change
1.0	ESA-01	"please reformulate, what do you mean by [...] to be addressed?	Page 5 / 1.1	The sentence has been updated.
1.0	ESA-02	Would [...] particularly due to the need to address the limited availability of the data back in time. [...] be better?"	Page 6 / 1.3	The last version of all the documents can be found at https://climate.esa.int/en/projects/high-resolution-land-cover/key-documents/ and they are also available to ESA by FTP and/or email
1.0	ESA-03	Consider adding where they are available, project website / useful documents? Are they still available?	Page 10 / 2	The sentence has been updated.
1.0	ESA-04	what about forward years as described above (2024)?	Page 10 / 2	The sentence has been updated.
1.0	ESA-05	Please replace "will be then"with "will then be"	Page 32 / Table 6	The values were typos. Table 6 has been updated with the correct labels.
1.0	ESA-06	TYPO: Maria Antonia Brovelli	Page 69 / Reference 38	The reference has been updated

Contents

1 Introduction

1.1 Executive summary

The algorithm development targets the specific technical requirements identified in Task 1 of the project and expands upon the work initiated during phase 1. In phase 1, the Algorithm Theoretical Basis Document (ATBD) [AD4] was first developed, and current document represents the continuation of the development. The current activity focuses on refining and testing the algorithm blocks in the processing chain identified by the Earth Observing Science (EOS) team which includes the optical processing chain, SAR processing chain, fusion chain, and change detection. The optical pre-processing chain follows the same logical steps established in phase 1, but with adjustments aimed at improving both the quality of the optical composites generation and classification together with reducing computational costs. In the SAR processing chain, gaps in data distribution are highlighted, particularly due to the need to address the limited availability of the data back in time. This phase incorporates a deep learning network applied to multitemporal SAR data analysis. The Decision fusion chain is being extended to reduce and mitigate residual artifacts observed in the phase 1 products, ensuring better spatial and temporal consistency. Lastly, the change detection process is being refined to enhance feature selection and reduce computational burden, further improving efficiency in detecting LC changes across the study areas. This version of the document highlights several promising algorithms, but more concrete insights will be provided after in the next versions of the document. The activities developed on different extensions of phase 1 activities in two cycles can be summarized as follows:

1. Re-processed Phase 1 historical products: generation of an improved version of historical products of Phase 1 through improvement of the enhanced sensor decision fusion, spatial and temporal harmonization modules of the processing chain. The starting point of the re-processing will be the intermediate products (from hereafter called meta products) consisting of the pixel-wise class-posterior probabilities generated by the SAR and Optical processing chains during Phase 1. Therefore, the SAR and Optical processing chain will not be run again on the already produced areas and years, minimizing costs, and allowing the team to better focus on the temporal consistency and change detection reliability, as well as on the new area and years that will be produced.
2. Historical production on a new selected area: improved SAR and Optical processing chains will be defined and run on the new selected area in addition to the previously mentioned enhanced sensor decision fusion, spatial and temporal harmonization modules. Note that the same sensor decision fusion and the spatial and temporal harmonization modules will be used for both re-processing Phase 1 products and for generating new Phase 2 products to generate compatible and consistent products.
3. Historical (forward) production of year 2024 for all the considered areas: Phase 2 will develop a different concept used for the historical production when considering the extension to years following 2019. In Phase 1, a backward approach was taken, producing static maps for 2019, and then proceeding backward for the historical production. Instead, Phase 2 will consider a forward approach to produce historical maps that extend forward in time after the static maps of 2019. The production of the year 2024 will be performed on the same historical areas considered in the backward approach. The backward and forward approaches will differ not only for the temporal direction, but also for the spatial resolution and data availability. Indeed, we can refer to the backward phase as the Landsat Era, characterized by the 30m spatial resolution and reduced data availability, and to the forward phase as the Sentinel Era, characterized by a 10m spatial resolution and a higher availability of satellite image data. This also leads to some differences in how the temporal correlation will be exploited, as in the Landsat Era more inconsistencies are expected in the meta products due to the use of different sensors and to lower data resolution, availability, and quality, requiring different levels of regularization between the two eras.

1.2 Purpose and scope

The ATBD outlines the algorithms used in the processing chain to generate the land cover products described in the PSD [AD2]. Its purpose is to provide a clear understanding of the processing workflow. ATBD version 1.0 presents refining and testing of the best algorithm candidates identified for implementation. Feedback from ongoing analysis will be incorporated in the next version of the document, ensuring that the selected algorithms meet the technical requirements.

The main blocks of computation can be identified as:

- Optical pre-processing.
- SAR pre-processing.
- Training dataset.
- 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
[AD2]	CCI_HRLC_Ph2-D1.2_PSD, latest version
[AD3]	CCI_HRLC_Ph2-D1.1_URD, latest version
[AD4]	CCI_HRLC_Ph1-D2.2_ATDB, latest version

available at https://climate.esa.int/en/projects/high-resolution-land-cover/key-documents/

1.4 Acronyms and abbreviations

3D-FCN	3-Dimensional - Fully Convolutional Network
6S	Second Simulation of a Satellite Signal in the Solar Spectrum
AC	Atmospheric correction
AMI	Active Microwave Instrument
AOT	Aerosol Optical Thickness
ARD	Analysis Ready Data
ASM	Angular Second Moment
ATBD	Algorithm Theoretical Basis Document
BEAST	Bayesian Estimator of Abrupt change, Seasonality & Trend
BFAST	Breaks For Additive Seasonal & Trend Bayesian Online Change Point Detection
BRDF	Bidirectional Reflectance Distribution Function
BOCPD	Bayesian Online Change Point Detection
CCI+	Climate Change Initiative Extension
CFmask	C version of Function of Mask
CGLS	Copernicus Global Land Service
CGLS-LC100	Copernicus Global Land Service Dynamic Land Cover map at 100 m resolution
CNN	Convolutional Neural Network
ConvLSTM	Convolutional Long Short-Term Memory
CRG	Climate Research Group
CSI	Cloud Shadow Index
DEM	Digital Elevation Model
DL	Deep Learning
DN	Digital Number
DEM	Digital Elevation Model
DSM	Digital Surface Model
DuPLO	DUal view Point deep Learning architecture for time series classificatiOn
ENL	Equivalent Number of Looks

ESA	European Space Agency
ETM	Enhanced Thematic Mapper
ETM+	Enhanced Thematic Mapper Plus
FCN	Fully Convolutional Networks
Fmask	Function of Mask
FORCE	Framework for Operational Radiometric Correction for Environmental monitoring
GLCM	Gray-Level Co-Occurrence Matrix
GRNN	General Regression Neural Network
GRD	Ground Range Detected
GRU	Gated Residual Unit
GSFC	NASA Goddard Space Flight Center
GPS	Global Positioning System
HLS	Harmonized Landsat Sentinel-2
HR	High Resolution
HRLC10	CCI High Resolution Land Cover Map at 10m resolution of 2019
HRLC30	CCI High Resolution Land Cover Map at 30m resolution from 1990 onwards every 5 years
HRLCC30	CCI High Resolution Land Cover Change Map at 30m resolution from 1990 onwards
IP	Image Patch
IW	Interferometric Wide Swath
k-NN	k-Nearest Neighbours
L-5/7/8/9	Landsat-5/7/8/9
L1C	Sentinel-2 Level 1C Top of Atmosphere product
L1	Landsat Level 1 Top of Atmosphere product
L2A	Sentinel-2 Level 2A Surface Reflectance product
L2	Landsat Level 2 Analysis Ready Data product
LaSRC	Landsat Surface Reflectance Code
LC	Land Cover
LCC	Land Cover Change
LEDAPS	Landsat Ecosystem Disturbance Adaptive Processing System
LDP	Local Directional Pattern
LPF	Low-Pass Filter
LSP	Land Surface Phenology
LSTM	Long Short Term Memory
LUT	Look-Up Table
MCMC	Markov Chain Monte Carlo
MEaSUREs	Making Earth Science Data Records for Use in Research Environments
MGRS	Military Grid Reference System
MHCVA	Multi-feature Hyper-temporal Change Vector Analysis
MMSE	Minimum Mean-Square Error
MODIS	Moderate Resolution Imaging Spectroradiometer
MOLCA	Map Of LC Agreement
MRLC	Medium Resolution Land Cover
MSI	MultiSpectral Instrument
MSS	Multispectral Scanner
MSSI	Mean Structural Similarity Index
MLCNN	Multi-Layer Perceptron Neural Network
NASA	National Aeronautics and Space Administration
NBR	Normalized Burn Ratio
NDBI	Normalized Difference Build-up Index
NDI	Normalized Difference Index
NDSI	Normalized Difference Snow and Ice Index
NDVI	Normalized Difference Vegetation Index
NDWI	Normalized Difference Water Index
NIR	Near InfraRed
OA	Overall Accuracy
OLI	Operational Land Imager
PCA	Principal Component Analysis
PSD	Product Specification Document

PSNR	Peak Signal-to-Noise Ratio
RABASAR	Ratio-Based Multi-temporal SAR Images Denoising
RBF	Radial Basis Function
RD	Range Doppler
ReLu	Rectified Linear Unit
RGB	Red, Green, and Blue optical bands
RNN	Recurrent Neural Network
S-1/2	Sentinel-1/2
S2AC	Sentinel-2 Atmospheric Correction
SAR	Synthetic Aperture Radar
SAVI	Soil-Adjusted Vegetation Index
SCL	Sentinel-2 L2A Scene Classification Layer
SIFT	Scale-Invariant Feature Transform
SITS	Satellite Image Time Series
SLC	Scan-line corrector
SNAP	Sentinel Application Platform
SR	Surface Reflectance
SRTM	Shuttle Radar Topography Mission
SVM	Support Vector Machine
SWIR	Short Wave InfraRed
TempCNN	Temporal Convolutional Neural Network
TIRS	Thermal Infrared Sensor
TM	Thematic Mapper
TOA	Top Of Atmosphere
TS	Time Series
TSA	Time Series Analysis
UEXT	Urban EXTent
UTM	Universal Transverse of Mercator
VH	Vertical-Horizontal polarization
VHR	Very High Resolution
VV	Vertical-Vertical polarization
WSL	Weakly Supervised Learning

1.5 List of Symbols

$*$	Convolution operation
$\oplus$	Dilation operation
$\mathbf{X}$	Satellite image / multidimensional tensor
$\mathbf{X}(b)$	Grayscale image of satellite image band b
T	Number of images in SITS
t	Timestep of image in SITS
$\mathbf{X}_t$	Satellite image tensor at timestep t in SITS
$\boldsymbol{W}_g, \boldsymbol{W}_x, \boldsymbol{W}_\rho$	Weights of the 1×1 convolution layers
(i, k, t)	Pixel location in space and time
(i', k')	Pixel location in space in the neighborhood $\mathcal{N}$
j	Imaginary unit
$\boldsymbol{x}$	Feature vector / generic pixel in image $\mathbf{X}$
x	Scalar / greyscale value of generic pixel in image $\mathbf{X}$
$\boldsymbol{x}(i, k)$	Pixel spectral vector at location (i, k) of image $\mathbf{X}$
$x(i, k, b)$	Scalar value of band b at location (i, k) of image $\mathbf{X}$
$x(i, k)$	Scalar value at location (i, k) of image $\mathbf{X}$
$\sigma^0(i, k)$	Backscattering coefficient at pixel (i, k)
$R(i, k)$	Slant range distance between the radar and the pixel (i, k)
$h_r(\cdot)$	Matched Filter function
Y	Year in multi-annual data
M	Number of years
B	Total number of bands
I	Width of satellite image

K	Height of satellite image
W	Width of kernel
H	Height of kernel
$H_a(\cdot)$	Azimuth filter
$Watershed(\cdot)$	Watershed function
A	Calibration constant or Scaling factor
C	Calibration factor / SVM parameter
$\mathcal{C}$	Number of land cover classes
P	Total number of pixels
D	Total number of pixels in the neighborhood
F	Total number of features
S	Shape parameter
$\mathcal{F}$	Fourier transform
$\mathcal{F}^{-1}$	Inverse Fourier transform
$\mathcal{D}$	Training dataset
N	Number of training samples
ℓ	Land cover label
$\mathcal{L}$	Loss function
$\mathcal{N}$	Neighborhood defined by the kernel size
$\mathcal{R}$	Empirical risk
$\alpha(i,k)$	Local incidence angle for pixel (i,k)
$\boldsymbol{\vartheta}$	Model parameters vector
$\mathcal{H}$	Hyperplane
F	Functional margin
G	Geometric margin
$\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j)$	Kernel function
η^2	Noise variance (constant)
ξ	Cluster of the k-NN algorithm / SVM slack variable
$U(\cdot)$	Energy function
$V(\cdot)$	Potential function

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, Amazonia and Siberia URD [AD3]. LC maps will be provided at 10m resolution for static map of 2019 (HRLC10) and at 30m resolution for the backward/forward historical record of LC and LCC from 1990 to 2024 every five years for HRLC30 and yearly forHRLCC30. The high-resolution (HR) classification legend as agreed by the Consortium is listed in URD [AD3]. The processing chain, outlined in Figure 1 and Figure 2, was initially developed during phase 1 and operates independently without relying on pre-existing LC products.

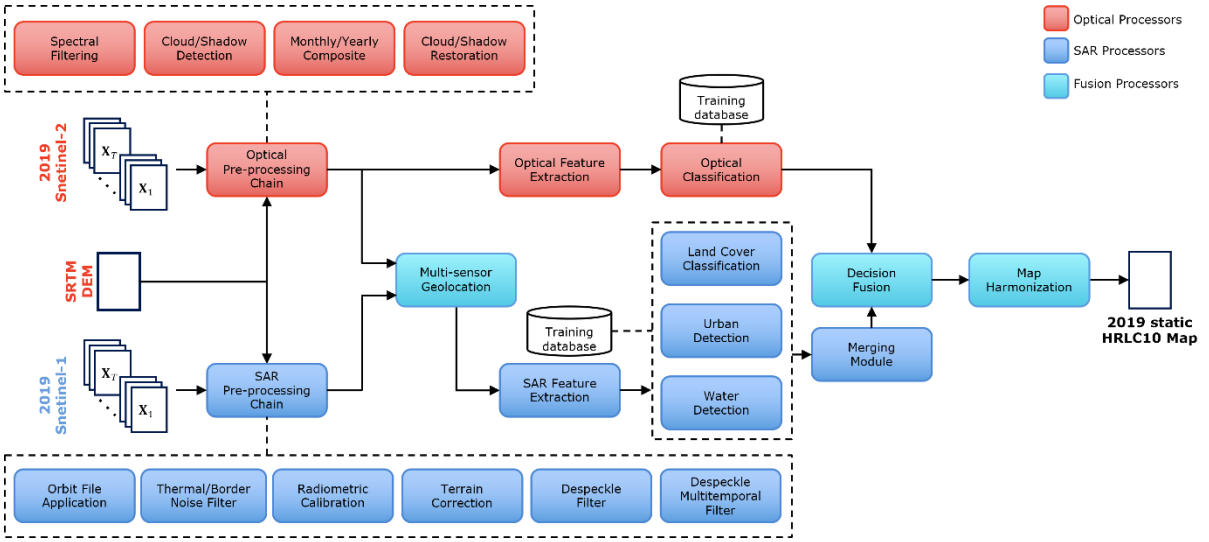


Figure 1. Block-based representation of the processing chain for the production of static HRLC10 maps.

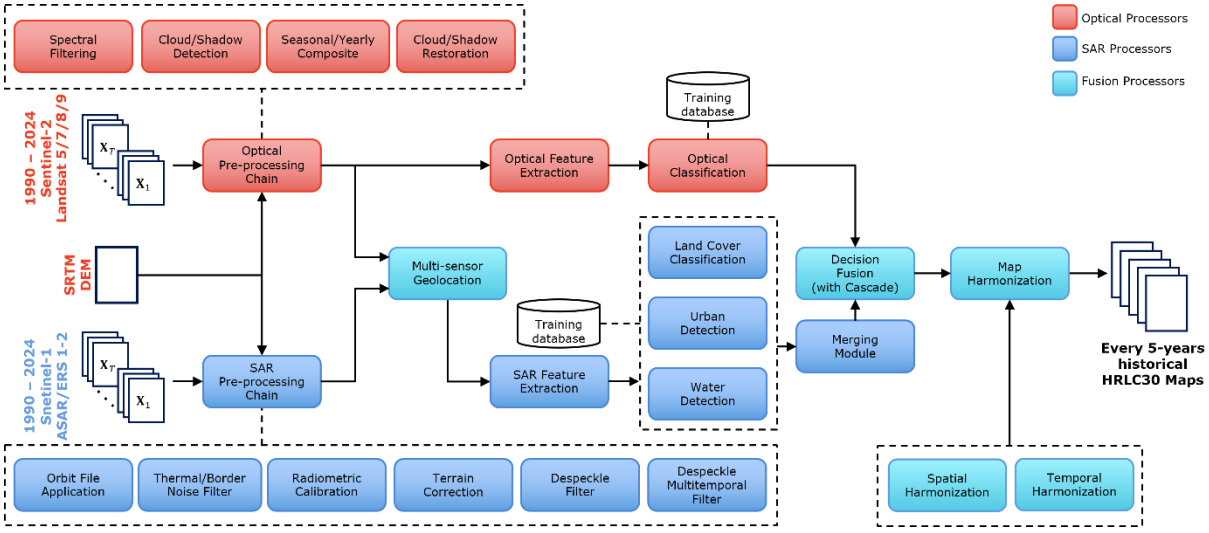


Figure 2. Block-based representation of the processing chain for the production of historical HRLC30 maps.

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-8/9 in the recent years and legacy L-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 then be 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 analysed 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.

Final high resolution land cover classification legend defined by the Climate Research Group (CRG) for the choice of the best performing classification algorithm is shown in Table 1.

Table 1. Final high resolution HR Land Cover classification legend defined during the HRLC project activity.

HRLC CLASSES			
CODE	DESCRIPTION		
0	No data		
10	Tree cover evergreen broadleaf		
20	Tree cover evergreen needleleaf		
30	Tree cover deciduous broadleaf		
40	Tree cover deciduous needleleaf		
50	Shrub cover evergreen		
60	Shrub cover deciduous		
70	Grasslands		
80	Croplands		
90	Woody vegetation aquatic or regularly flooded		
100	Grassland vegetation aquatic or regularly flooded		
110	Lichens and mosses		
120	Bare areas		
130	Built-up		
140	Open water	141	Open water seasonal
		142	Open water permanent
150	Permanent snow and/or ice		

3 Optical pre-processing

The optical pre-processing follows the same logical steps used in Phase 1, as reported in the ATDB [AD4]. Nonetheless, we plan on adjustments aimed at improving the quality of the pre-processing chain output (i.e., the optical composites), and the computational costs associated with it. The following will describe the theoretical basis for the optical pre-processing chain, with additional details on the proposed improvements under investigation and the related motivations.

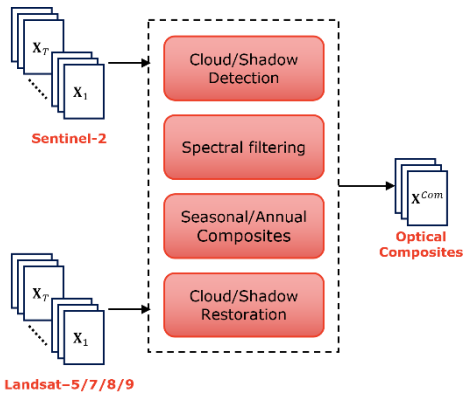


Figure 3. Optical pre-processing chain.

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 cloud 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.

3.1 Atmospheric Correction and Cloud / Cloud Shadow Detection

The input data to the processing chain are the atmospherically corrected S-2 Collection 1 data (i.e., L2A products) and atmospherically corrected L-5/7/8/9 Collection 2 images (i.e., L2 products), i.e., Surface Reflectance (SR) products. Although the data used in the processing chain of Phase 1 are already atmospherically corrected, in the following subsections we described the algorithms used to generate such products as well as the cloud and shadows masks. In Phase 2, an alternatives to the original L2A and L2 product are currently being considered, which are based on FORCE [1] and HLS [2], for the production in the Sentinel Era (2015 onward), as they provide frameworks for seamlessly integrating Landsat and Sentinel-2 data. They also provide alternative cloud detection algorithms that are being compared with current operational methodologies. Therefore, some details are needed to understand the key differences of these approaches.

3.1.1 Sentinel-2 – Sen2cor

The precomputed SR products in L2A are generated using Sen2cor. The Sen2cor processor allows calculation of atmospherically corrected SR from Top Of Atmosphere (TOA) reflectance images available in L1C products. S-2 atmospheric correction (S2AC) is based on an algorithm proposed in [3]. The method performs atmospheric correction based on the LIBRADTRAN radiative transfer model presented in [4].

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 with 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 the SR. 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.

The Scene Classification (SCL) algorithm allows the detection of clouds, snow and cloud shadows and generation of a classification map, which consists of three different classes for clouds (including cirrus), together with six different classifications for shadows, cloud shadows, vegetation, not vegetated, water and snow. Cloud screening is applied to the data in order to retrieve accurate atmospheric and surface parameters during the atmospheric correction step. The L2A SCL map can also be a valuable input to the optical processing chain for further processing steps or data analysis (e.g., composite generation).

The SCL algorithm uses the reflective properties of scene features to establish the presence or absence of clouds in a scene. It is based on a series of threshold tests that use as input the following: TOA reflectance of several S-2 spectral bands, band ratios and indexes like Normalised Difference Vegetation Index (NDVI) and Normalised Difference Snow and Ice Index (NDSI). For each of these threshold tests, a level of confidence is associated. It produces at the end of the processing chain a probabilistic cloud mask quality indicator and a snow mask quality indicator. The most recent version of the SCL algorithm includes also morphological operations, usage of auxiliary data like DEM and LC information and exploit the parallax characteristics of S-2 MSI instrument to improve its overall classification accuracy.

The S-2 SR products are used as the input to the pre-processing chain for the static HRLC10 maps production, and the associated SCL is used as starting point for the cloud and cloud-shadow masks definition.

3.1.2 Landsat 5/7/8 – LEDAPS, LaSRC

L-5 TM and L-7 ETM+ Collection 2 SR products are generated using the Landsat Ecosystem Disturbance Adaptive Processing System (LEDAPS) algorithm (version 3.4.0), 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 [5]. The software applies Moderate Resolution Imaging Spectroradiometer (MODIS) atmospheric correction

routines to L1 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, SR, TOA brightness temperature, and masks for clouds, cloud shadows, adjacent clouds, land, and water.

L-8/9 OLI Collection 2 SR data are generated using the Landsat Surface Reflectance Code (LaSRC) (version 1.5.0), 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 [6]. 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. Table 1 below illustrates both.

Table 2. Differences between Landsat-5/7 and Landsat-8/9 surface reflectance algorithms.

Parameter	Landsat-5/7 (LEDAPS)	Landsat-8/9 (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 (Bands 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	Not Used
DEM	ETOPO5 (CMGDEM)	ETOPO5 (CMGDEM)
Ozone	OMI/TOMS	MODIS CMG Coarse resolution ozone
AOT	Correlation between chlorophyll absorption and bound water absorption of scene	Internal algorithm
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 (v3.3.1)	CFmask (v3.3.1)
Data format	INT16	INT16
Fill values	0	0
QA bands	Cloud Adjacent cloud Cloud shadow DDV Fill Land water Snow Atmospheric opacity	Cloud Adjacent cloud Cloud shadow Aerosols Cirrus Aerosol In

Identification of clouds, cloud shadows in optical images is necessary. The C version of Fmask (CFmask) v3.3.1 has been used in Collection 2 to accomplish these tasks for use with images from L-5/7/8/9 [7]. CFmask is a multi-

pass algorithm that uses decision trees to prospectively label pixels in the scene; it then validates or discards those labels according to scene-wide statistics. It also creates a cloud shadow mask by iteratively estimating cloud heights and projecting them onto the ground. Fmask algorithm can also be used with Sentinel-2 images, and the most recent Fmask v4.6 [8] implements strategies for exploiting the parallax effect as proposed in [9].

3.1.3 Sentinel-2 for HRLC10 Map – SCL Cloud Masks Improvement

Cloud and cloud shadow detection is based on cloud and cloud-shadow masks provided with the S2A SCL (for S-2) and Fmask (for Landsat). The Overall Accuracy (OA) of cloud and shadow masks provided by the S2A SCL (84%) is on average lower than the one provided by Fmask (90%) [10]. Therefore, the S2A SCL masks should be further enhanced to achieve the required accuracy. To this end, during Phase 1 we adopted two strategies, one for cloud detection and one for cloud shadow detection and removal. Let $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_T\}$ be the considered satellite image TS (SITS), which includes the T optical images acquired over a season. The multitemporal pattern associated to the pixel (i,k) of the SITS can be defined as $\{\boldsymbol{x}_1(i,k), \mathbf{x}_2(i,k), \dots, \boldsymbol{x}_T(i,k)\}$, where $\boldsymbol{x}_t(i,k) = [x_t(i,k,1), x_t(i,k,2), \dots, x_t(i,k,B)]^{\mathrm{T}}$ represents the column vector of B spectral values of the pixel (i,k) in the image X_t of the SITS. For cloud detection, we compute the cloudless background Blue image $\mathbf{X}^{Bg}$ for each season [11]:

$$x^{Bg}(i,k) = \mathrm{Quantile}_{0.25}(\{x_1(i,k,Blue), x_2(i,k,Blue), \dots, x_T(i,k,Blue)\})$$

The difference between the Blue bands of each image from the TS and the background image is computed. The pixels in the difference image are then clustered into 3 clusters. To understand which from the obtained clusters belong to cloud cover, the mean of each cluster is compared with the blue band mean of the cloudy pixels overall image. Finally, we merge the obtained cloud mask with the original S2A SCL mask. Note that this strategy is performed only for tiles with a sufficiently large cloud cover to properly model the clusters. Figure 4 shows the flowchart of the considered strategy.

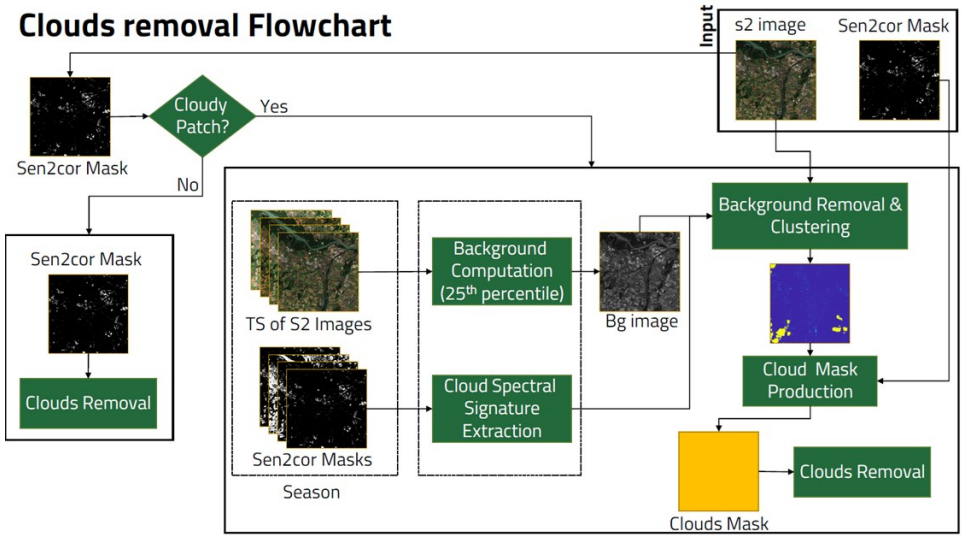


Figure 4. Flowchart of the Sen2cor cloud mask improvement

In order to detect and remove cloud shadows, the Cloud Shadow Index (CSI) [12] can be used, which is based on the physical reflective characteristic of cloud shadow. The $\mathbf{CSI}_t$ image of $\mathbf{X}_t$ for pixel (i,k) is computed by combining information provided by the NIR and SWIR bands:

$$csi_t(i,k) = \frac{1}{2}\left(x_t(i,k,NIR) + x_t(i,k,SWIR)\right)$$

To avoid confusion between shadows and water bodies, as they both have very similar spectral signatures associated with their low reflectance, an additional condition including shorter wavelengths, *i.e.*, the blue band reflectance, should also be analysed. Thus, the cloud shadow is identified in areas where the following conditions are fulfilled:

$$csi_t(i,k) < \min_{(i,k)} csi_t(i,k) + \lambda_1 \left(\mathrm{mean}_{(i,k)}\, csi_t(i,k) - \min_{(i,k)} csi_t(i,k)\right)$$

$$x_t(i,k,Blue) < \min_{(i,k)} x_t(i,k,Blue) + \lambda_2 \left(\mathrm{mean}_{(i,k)}\, x_t(i,k,Blue) - \min_{(i,k)} x_t(i,k,Blue)\right)$$

Coefficients were fine-tuned: $\lambda_1 = 1/2$ and $\lambda_2 = 1/4$. Note that this approach is performed only for tiles where there is enough cloud cover, and the cloud cover has on average a large reflectance. Figure 5 shows the flowchart of the considered strategy.

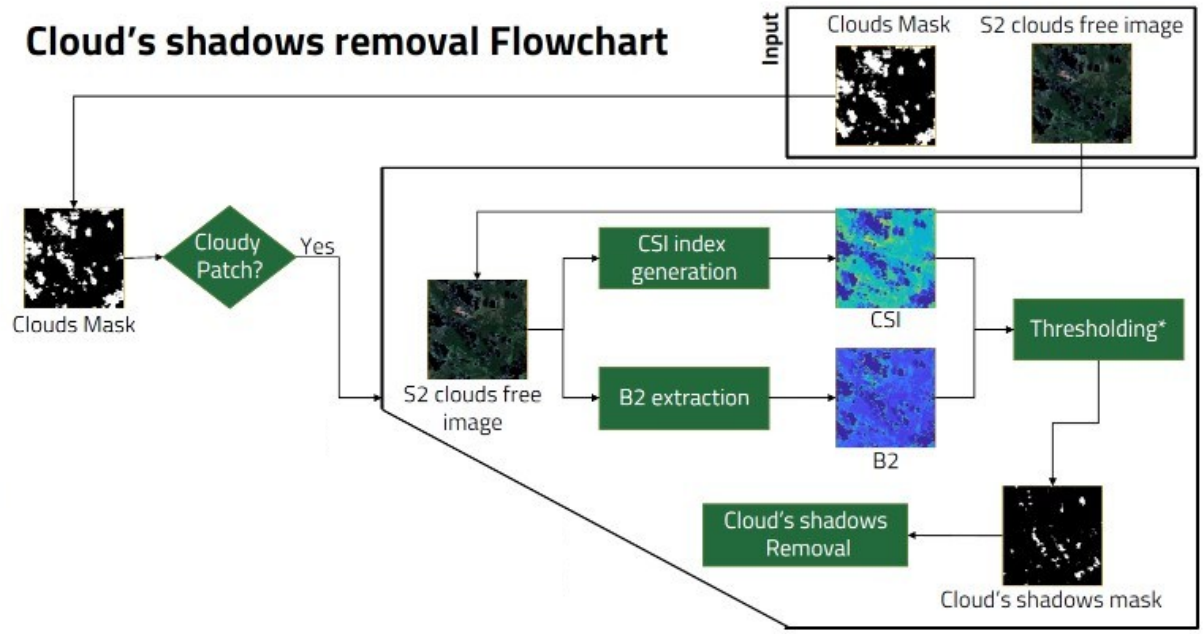


Figure 5. Flowchart of the Sen2cor cloud shadow mask improvement and removal

In Phase 2, this approach is being re-evaluated against the updated operational algorithm for the S-2 Collection 2 masks and the newer version of Fmask. A comparison will be performed using the recent multi-temporal global benchmark dataset, named CloudSEN12 [13], for cloud and cloud shadow detection with S-2. This dataset provides 49,250 S-2 image patches (IPs) with different annotation types: (i) 10,000 IPs with high-quality pixel-level annotation, (ii) 10,000 IPs with scribble annotation, and (iii) 29,250 unlabelled IPs. The labelling phase was conducted by 14 domain experts using a supervised active learning system. A rigorous four-step quality control was designed to guarantee high quality in the manual annotation phase. Furthermore, CloudSEN12 ensures that for the same geographical location, users can obtain multiple IPs with different cloud coverage: cloud-free (0%), almost-clear (0–25%), low-cloudy (25–45%), mid-cloudy (45–65%), and cloudy (>65%), which ensures scene variability in the temporal domain. Therefore, CloudSEN12 provides a reliable benchmark for precisely evaluating different cloud detection algorithms.

3.1.4 Harmonized Landsat Sentinel-2 data

Since the launch of the first S-2 satellite, both S-2 and Landsat missions acquired large volume of data over the globe, potentially increasing the temporal density of the acquisitions that can be considered for our analysis. However, the integration of the data of these two missions comes with its challenges related to the different characteristics of the sensors onboard the different satellites. For this reason, during Phase 1 these data were used separately to generate the intermediate results and combined at decision-fusion/cascade level (*i.e.*, HRLC30 2015-2019).

In the last few years, thanks to the more mature state of both Landsat and S-2 missions, international collaborations delivered a new dataset fully focused on a different processing of the Landsat and S-2 data to provide seamless products that bridge the gap between the two missions, the Harmonized Landsat and Sentinel-2 (HLS) dataset [2]. HLS is a NASA initiative aiming to produce a seamless surface reflectance record from the Operational Land Imager (OLI) and Multi-Spectral Instrument (MSI) aboard L-8/9 and S-2A/B remote sensing satellites, respectively. The HLS products are created from a set of algorithms:

1. Atmospheric correction: LaSRC is used for both Landsat and Sentinel-2 Level 1 acquisitions.
2. Cloud and cloud-shadow masking: Fmask version 4 is used for both missions.
3. Geographic co-registration and common gridding: Sentinel-2 bands are resampled to 30m, while Landsat bands are reprojected and resampled to match the Sentinel 2 MGRS tiling grid.
4. Bidirectional Reflectance Distribution Function (BRDF) normalization: The view angle effect on surface reflectance is noticeable even for narrow field-of-view sensors like Landsat and S-2, especially where forward scattering and backward scattering are concerned. HLS normalizes the view angle effect in the Landsat/S-2 common bands and the S-2 red-edge bands using the c-factor technique and the global coefficients provided in [14], [15].
5. Bandpass adjustment: The small differences between MSI and OLI equivalent spectral bands are adjusted. The OLI spectral bands are used as reference, to which the MSI spectral bands are adjusted.

The bandpass adjustment is a linear transformation between equivalent spectral bands (see Table 3). With four sensors currently in this virtual constellation, HLS provides observations once every three days at the equator and more frequently with increasing latitude.
This dataset is currently under study for being used in the production of Phase 2 HRLC30 and HRLCC30 maps. The use of this dataset can be particularly beneficial to produce historical maps at 30m resolution ranging from 2019 to 2024 for HRLCC30 for all areas, for the HRLC30 2015-2019-2024 maps of the new extended Amazonia area PSD [AD2] and for the updated HRLC30 2024 maps for historical areas in Siberia and Africa Sahel. Note that this dataset contains only L-8/9 data, thus this can only be used from 2013 onward.

Table 3. Coefficients of linear regression used to adjust Sentinel-2A/B MSI to Landsat 8/9 OLI.

HLS Band Name	OLI Band Name	MSI Band Name	Sentinel-2A		Sentinel-2B	
			Slope	Offset	Slope	Offset
Coastal Aerosol	1		0.9959	-0.0002	0.9959	-0.0002
Blue	2	2	0.9778	-0.0040	0.9778	-0.0040
Green	3	3	1.0053	-0.0009	1.0075	-0.0008
Red	4	4	0.9765	0.0009	0.9761	0.0010
NIR	5	8A	0.9983	-0.0001	0.9966	0.0000
SWIR 1	6	11	0.9987	-0.0011	1.0000	-0.0003
SWIR 2	7	12	1.0030	-0.0012	0.9867	0.0004

3.1.5 Framework for Operational Radiometric Correction for Environmental monitoring

As an alternative solution to the use of pre-computed SR products, we are investigating the use of FORCE (Framework for Operational Radiometric Correction for Environmental monitoring) [1]. It provides an all-in-one processing engine that can compute SR products as well as cloud and cloud shadow masks for both Landsat and S-2 images in a unified framework. The algorithm for cloud detection is based on Fmask modified with the latest improvements (*e.g.*, S-2 parallax effect exploitation). FORCE provides a tool for generating a dataset that integrates Landsat and Sentinel-2 acquisitions; thus, it is a direct competitor for HLS. FORCE AC [16] resembles the techniques used for Landsat data. Radiometric correction includes radiative-transfer-based atmospheric correction. Aerosol optical depth is estimated over dark water and dense dark vegetation objects using multiple scattering. Water vapor is estimated for each S-2 pixel; auxiliary data are used for Landsat. Topographic correction is performed with an enhanced C-correction. The C-factor is estimated for each pixel in the image and then propagated through the spectrum using radiative transfer theory. Three kernels of increasing size are used to approximate the background reflectance for environment correction. Nadir BRDF-adjusted reflectance is retrieved using a global set of MODIS-derived BRDF kernel parameters. Figure 6 shows the flowchart of FORCE for the generation of Analysis Ready Data (ARD), *i.e.*, SR, and the related cloud masks.

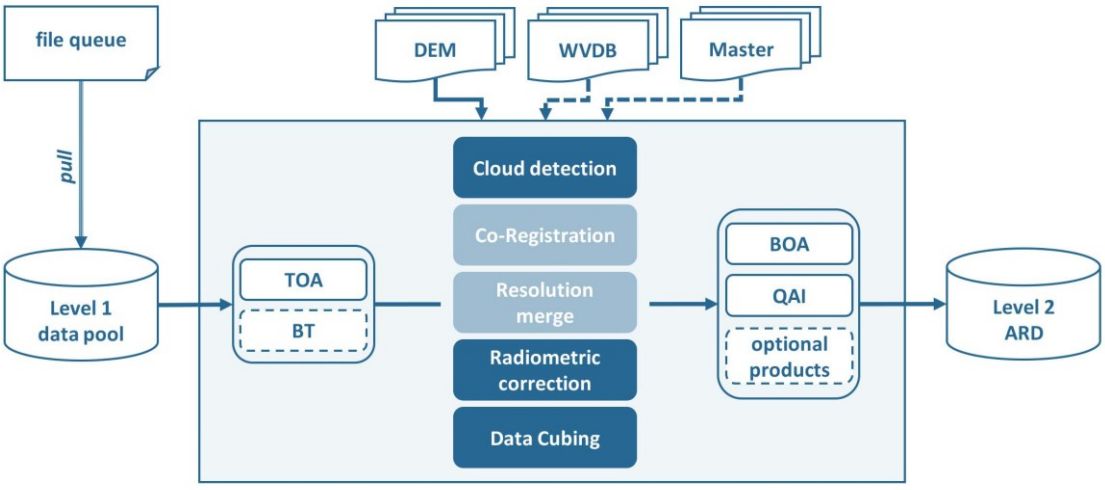


Figure 6. Flowchart of FORCE AC, cloud detection and generation ARD (*i.e.*, SR) products for Landsat and S-2 data.

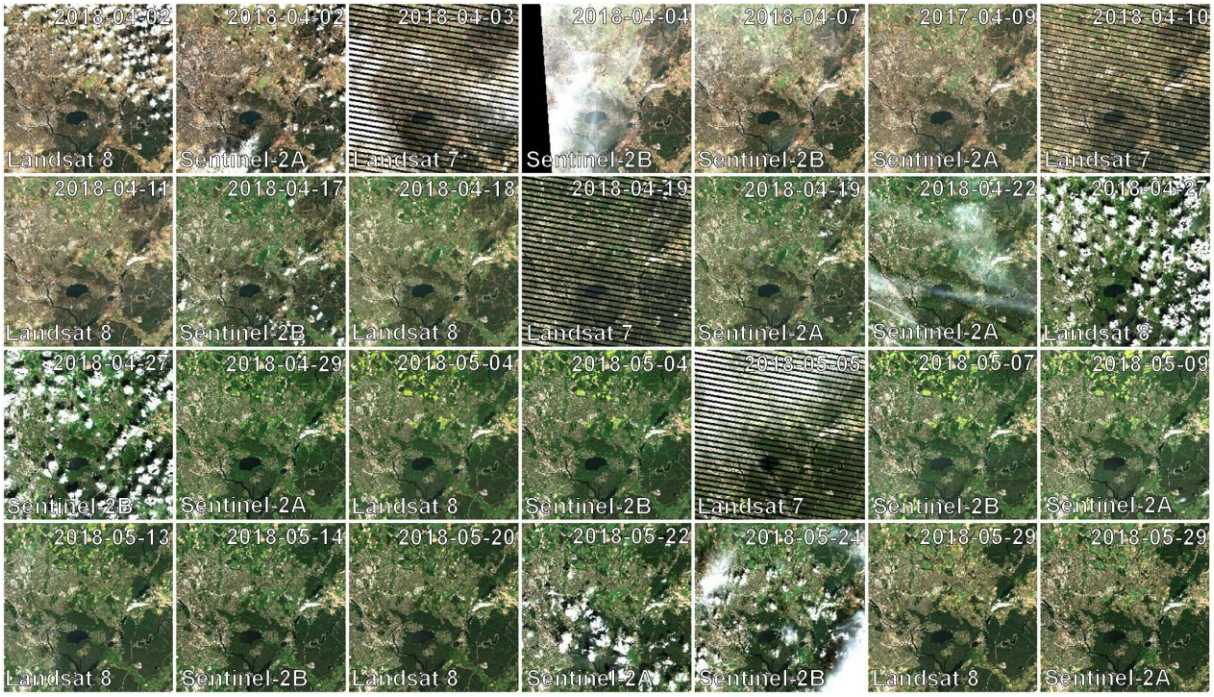


Figure 7. Example of Data Cube of L-7/8 and S-2 Level 2 SR as generated by FORCE.

3.2 Spectral Filtering

The spectral filtering aims to detect and remove the outlier present in the optical images. To this end, in this step we discard the reflectance values higher than the 0.999 quantile and lower than the 0.001 quantile of each spectral band. All the images considered in the experiments have cloud coverage less than 40%. In order to mitigate any possible effect of clouds and shadow present on the image, they have been detected by using the available cloud masks and discarded from the quantitative evaluation. S-2 bands at 60m resolution are discarded, and 20m resolution bands are up-sampled to 10m resolution by replication.

3.2.1 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. This problem affects the considered composite strategy when the available acquisitions are scarce and come mainly or only from L-7 (e.g., Africa 2005 and 2010). To avoid affecting the composites and the classification (*i.e.*, striping in the composites and in the land-cover maps), a gap-filling strategy based on interpolation has been used to fill in the values of the missing pixels. While accurate spatial information is not retrieved, the subsequent composite strategy is able to partially retrieve it by exploiting the multitemporal acquisitions. Even though the spatial detail might be reduced, this strategy results in improved spectral uniformity and consistency across pixels in the composite. This improved the performance of the classifier, which uses the spectral bands as its primary features. This step is currently under upgrade in Phase 2 to improve the radiometric and geometric properties of the gap-filling operator. We are considering images acquired just before and after the image of interest to provide spatial information within the affected stripe. Then, we will consider approaches reconstruct the stripe based on the radiometric properties of the local neighbourhood in the image of interest. An example of such a strategy in the literature is IROBOT [16], a method that utilizes the Neighbourhood Similar Pixel Interpolator to fill in missing values and leverages the time-series information to reconstruct high-resolution images. Therefore, by combining the spatial information of close acquisitions and the radiometric properties of the image of interest, we expect to be able to further improve the reconstruction quality. Figure 8 shows a qualitative example of gap filling that can be obtained with IROBOT.

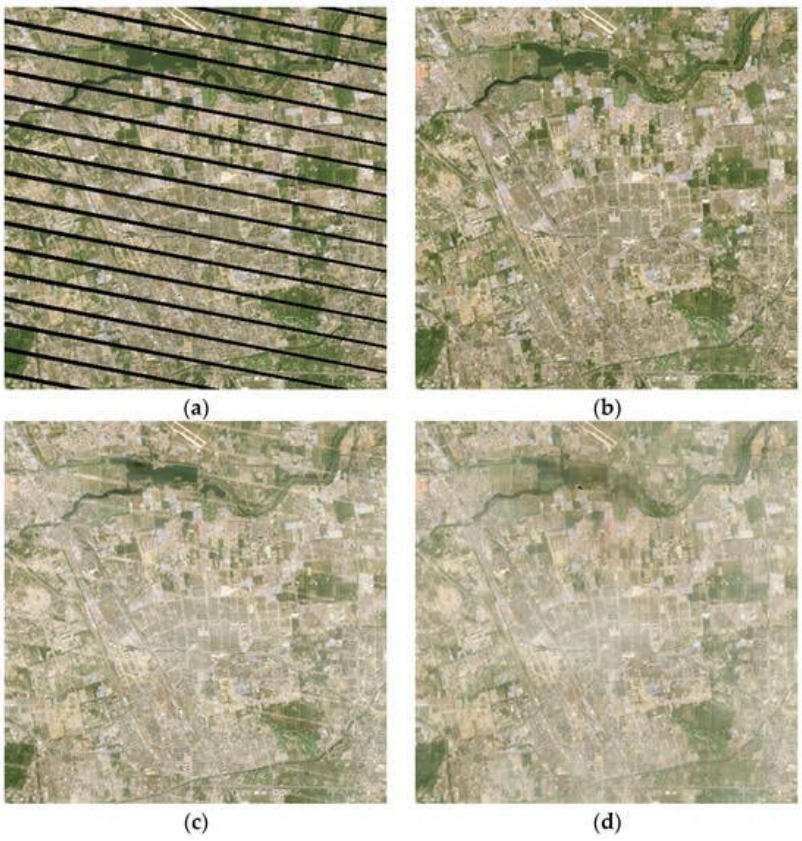


Figure 8. Comparison example from [16] of different gap-filling methods. (a) Landsat 7 OLI image. (b) The reconstruction image of IROBOT method proposed in [16]. (c) The reconstruction image of Linear-ROBOT method. (d) The reconstruction image of IDW-ROBOT method.

3.3 Composite Generation

When working at large scale, it is necessary to harmonize the TS of images acquired over different tiles which are characterized by different lengths and are acquired at different times. This is mainly due to the irregular cloud coverage (which hampers the use of some images of the time-series) and the different orbit acquisitions (different temporal sampling). To solve this problem, in the pre-processing step we generate monthly, seasonal and annual composites. This condition allows us to mitigate cloud occlusions problem and minimize the processing resources. To this end, we consider a statistic-based approach that computes the median value for each pixel. This approach can generate consistent results at large scale in an automatic way by sharply reducing the spatial noise. Let $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_T\}$ be the considered SITS which includes the optical images acquired over a month, a season or the whole year (*i.e.*, according to the sensor and the considered study area). The pixel (i, k) of the composite $\mathbf{X}^{Com}$ is generated by computing the band-wise median of the cloud-free multispectral pixels of the SITS as follows:

$$x^{Com}(i,k,1) = \text{Median}(\{x_1(i,k,1), x_2(i,k,1), \dots, x_T(i,k,1)\})$$
$$x^{Com}(i,k,2) = \text{Median}(\{x_1(i,k,2), x_2(i,k,2), \dots, x_T(i,k,2)\})$$
$$\vdots$$
$$x^{Com}(i,k,B) = \text{Median}(\{x_1(i,k,B), x_2(i,k,B), \dots, x_T(i,k,B)\})$$

Cloud, cloud shadow and snow mask pixels are ignored during median computation. Table 4 summarizes the kind of composite generated per study area according to different optical sensors. Due to the increased revisit time of S-2 (5 days) with respect to Landsat (16 days), denser time-series are available for 2019 that can be used to generate monthly composites. In the case of Sentinel data for HRLC10 over Amazonia and Africa, we computed 12 monthly composites. Due to dense cloud coverage over some regions, each monthly composite is computed using a buffer of 15 days around the considered month (*i.e.*, February is computed with data from 15th Jan to 15th Mar). This conservative choice allows us to sharply reduce the probability of having cloudy pixels in the TS. For S-2 data over Siberia, we generate yearly composites due to heavy cloud and snow coverage problems which hampered the use of images acquired for most of the year. Hence, the Siberian yearly composite is computed as the median of data acquired in July and August.

In the case of Landsat data for HRLC30, we similarly consider yearly composite for Siberia, which is computed as the median of data acquired between April and September. Finally, for Landsat data over Amazonia and Africa

we compute four seasonal composites considering the optical data acquired in the following months: (i) January – March, (ii) April – June, (iii) July- September, and (iv) October – December.
For the new products to be generated in Phase 2, both Landsat and S-2 images (within HLS or FORCE) will be considered for the composite generation over the extended historical area in Amazonia and the historical in Africa and Siberia maps for the period 2019-2024. This allows us to have an even denser time series, thus significantly improving the composite quality over the areas.

Table 4. Composites generated for the different study areas according to the availability of cloud free optical images.

Area	Sentinel 2	Landsat 5/7/8
Siberia	Yearly (July - August)	Yearly (April – September)
Amazonia	12 Monthly Composites	4 Seasonal Composites
Africa	12 Monthly Composites	4 Seasonal Composites

3.3.1 Additional compositing strategies

In Phase 2, alternatives to the Phase 1 band-wise median approach are being investigated. The main drawbacks of the median approach of Phase 1 are it being computationally demanding and it not always generating representative values of the temporal range considered. Indeed, the median values of each band may not belong to the same acquisition, thus the temporal mosaic would not depict a real spectral signature for the temporal range considered. A solution to this problem is to change the approach to the selection of the most representative image for each pixel. Such an approach guarantees that each pixel in the temporal mosaic reports a real spectral signature. Then, the issue we need to address is the selection of the representatives. A common approach is the medoid [17], widely used for Landsat and more recently also for S-2 data [18]. The medoid computes the representative object of a data set whose average dissimilarity to all the objects in the data set is minimal. Therefore, retaining the notation used before, we could use it to compute the composite $\mathbf{X}^{Com}$ for pixel (i,k) from T images as follows:

$$\boldsymbol{x}^{Com}(i,k) = \underset{\mathbf{x}_{t'}(i,k)}{\operatorname{argmin}} \sum_{t=1}^{T} d\big(\boldsymbol{x}_t(i,k), \boldsymbol{x}_{t'}(i,k)\big),$$

where $d\big(\boldsymbol{x}_t(i,k), \boldsymbol{x}_{t'}(i,k)\big)$ is a dissimilarity measure, *e.g.*, the Euclidean distance:

$$d\big(\boldsymbol{x}_t(i,k), \boldsymbol{x}_{t'}(i,k)\big) = \sqrt{\sum_{b=1}^{B} [x_t(i,k,b) - x_{t'}(i,k,b)]^2}.$$

Another alternative to median or medoid compositing strategies is the use of the Time Series Analysis (TSA) approach proposed in [1], depicted in Figure 9. The time series can be interpolated / smoothed at custom time steps. Currently available are linear interpolation, moving average filter, and Radial Basis Function (RBF) ensembles. TSA not only provides a tool for processing SITS, but also strategies for aggregating the temporal information over predefined temporal ranges. The time series can be "folded" by year, quarter, month, week or day, which perfectly align with our requirements for monthly, seasonal (i.e., quarterly) and yearly composites. The time series can be folded with any available statistics, e.g. mean or median. TSA approach is already part of the FORCE [1] suite of utilities, providing a convenient workspace for all the pre-processing operations.

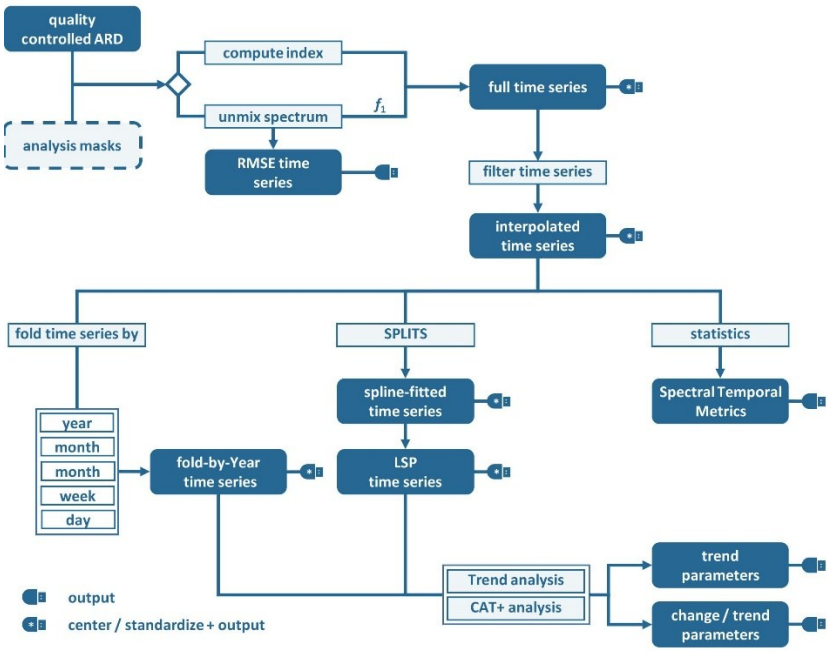


Figure 9. Flowchart of the TSA module of FORCE.

3.4 Cloud and cloud shadow restoration

Cloud and cloud shadow restoration is an important step in the optical image pre-processing part. Although we consider the composites instead of original time-series of images, missing information due to poor atmospheric conditions (*e.g.*, thick clouds and related shadows) or defective sensors may be present in the composites. In the literature, a large effort has been devoted to solving this problem. However, to properly recover missing information, sophisticated and usually computationally intensive techniques should be used, increasing significantly the computational complexity of the pre-processing part. Instead of considering computationally demanding approaches, a simple and effective linear temporal gap filling was employed. In this method the missing information is restored as the average of the spectral values acquired in the previous and the following images in the time series. If clouds are present in the first or last image in SITS, the second or the one before last image are considered, respectively.

4 SAR pre-processing

The Synthetic Aperture Radar (SAR) pre-processing chain aligns with Phase 1 production steps, as documented in the relevant ATDB deliverable [AD4]. A 10m resolution static map was generated using Sentinel-1 data, while mapping land cover (LC) back to 1990 at 30m resolution incorporated SAR data from Sentinel-1, ERS-1/2, and ENVISAT ASAR. Sentinel-1's Interferometric Wide Swath (IW) mode data has a resolution of 20x22m with 10x10m pixel spacing and a 12-day revisit period since 2015.
For historical LC mapping, SAR Level 1 Precision Image Products (SAR_IMP_1P) from ERS-1/2 [19] and ASAR IM Precision Level 1 (ASA_IMP_1P) from ENVISAT [20] were used. Both products provide multi-look, ground-range images with specific corrections to ensure consistency with ERS-SAR data.

Gaps in data coverage from 1990 to 2015 posed significant challenges, requiring careful data selection for consistent mapping. Data gaps particularly affected Amazonia (2015, 2010, 1990), Africa (2015, 1990), and Siberia (2015, 2000, 1990). The Table 5 summarizes the distribution of SAR datasets used for producing historical maps in the three target regions identified in Phase 1: Amazonia, Africa, and Siberia.

Table 5. SAR data availability in the three areas identified in Phase 1—Amazonia, Africa and Siberia—for the production of the historical products at 30m.

Area	Year	Date range	Season	SAR historical product	# images
Amazonia	2005	01.01 - 03.31	Winter	ENVISAT_ASA.IMP.1P	466

Area	Year	Date range	Season	SAR historical product	# images
Amazonia	2000	01.01 - 03.31	Winter	ERS_SAR.IMP.1P	396
Amazonia	1995	04.01 - 06.30	Spring	ERS_SAR.IMP.1P	421
Africa	2010	01.01 - 03.31	Winter	ENVISAT_ASA.IMP.1P	274
Africa	2005	07.01 - 09.30	Summer	ERS_SAR.IMP.1P	350
Africa	2000	07.01 - 09.30	Summer	ERS_SAR.IMP.1P	350
Africa	1995	04.01 - 06.30	Spring	ERS_SAR.IMP.1P	323
Siberia	2010	07.01 - 09.30	Summer	ERS_SAR.IMP.1P	895
Siberia	2005	07.01 - 09.30	Summer	ENVISAT_ASA.IMP.1P	315
Siberia	1995	07.01 - 09.30	Summer	ERS_SAR.IMP.1P	548

To process and analyze the available SAR data, custom codes were developed in the Python programming language. These codes were deployed using Docker containers, enabling automated, platform-independent execution across various operating systems. This approach ensured consistent and efficient processing workflows, regardless of the underlying computing environment.

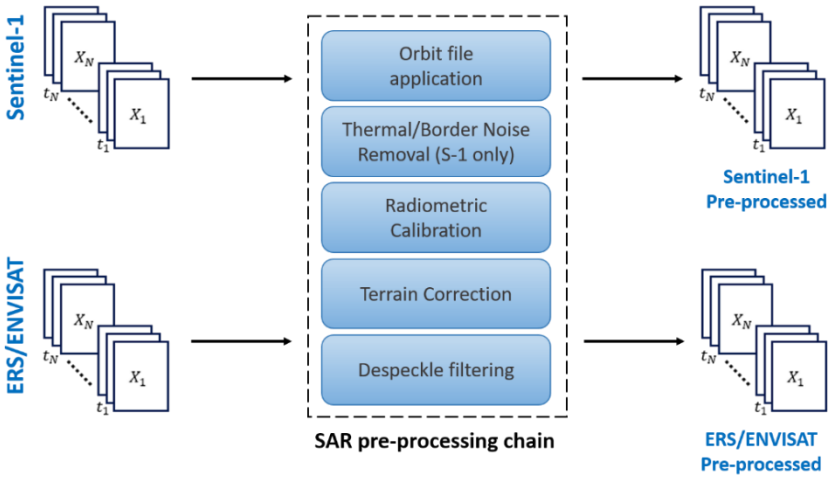


Figure 10. Block diagram illustrating the processing chain used for the pre-processing of SAR data: Sentinel-1 for generating the static map at a 10m resolution, and ERS/ENVISAT for producing historical maps at a 30m resolution.

The pre-processing involves several key steps, shown in block scheme in Figure 10:

- **Orbit File application:** Corrects satellite position and velocity for accurate geolocation using precise orbit data.
- **Thermal Noise removal (for Sentinel-1 only):** Enhances backscatter reliability by removing noise, especially from the cross-polarization channel.
- **Border Noise removal (for Sentinel-1 only):** Applies a threshold-based masking approach using a *NoiseMak*.
- **Radiometric calibration:** Converts SAR signals to calibrated backscatter values, enabling comparability across sensors.
- **Geometric Terrain correction:** Uses Range Doppler (RD) techniques with a Digital Elevation Model (DEM) for accurate geographic representation.
- **Despeckle filtering:** Reduces speckle noise, improving clarity while preserving details.

The initial processing involves **orbit file application**, integrating satellite trajectory data for geolocation tasks using interpolation methods like cubic splines or Lagrange interpolation to account for irregular time spacing [21].

The range compression uses a matched filter:

$$s_c(t) = s(t) * h_r(t)$$

where $s_c(t)$ is the range-compressed signal, $s(t)$ is the received SAR signal, and $h_r(t)$ is the matched filter function. The azimuth compression uses Fourier transforms:

$$S_c(f) = \mathcal{F}\{s_c(t)\}, \quad s_f(t) = \mathcal{F}^{-1}\{S_c(f) \cdot H_a(f)\}$$

With $H_a(f)$ as the azimuth matched filter. The final SAR image $\boldsymbol{X}$ is reconstructed as $\boldsymbol{X} = |s_f(t)|$

Thermal noise correction is performed using:

$$x_{thermal\_corrected}(i,k) = x(i,k) - n_T(i,k),$$

Where $x_{thermal\_corrected}(i,k)$ is the noise-corrected pixel intensity at location (i,k); $x(i,k)$ is the SAR pixel intensity (output of the orbit file application step) at location (i,k), and $n_T(i,k)$ represents the thermal noise estimate for that pixel. For SAR data like Sentinel-1, thermal noise removal often involves using noise vectors from the provided metadata noise.xml file [22]. Effective noise removal normalises the backscatter signal across the entire scene, crucial for multi-swath acquisitions to minimise discontinuities between sub-swaths. [23].
Tools like SNAP facilitate thermal noise removal for Sentinel-1 data by providing a specialised operator [24], capable of updating product annotations and handling Look-Up Tables (LUTs) for calibrated noise profiles. This enhances image coherence and quality for various remote sensing applications.

To **remove border noise**, a threshold-based masking approach is commonly used [25]. The basic concept is to set the pixel values at the borders of the SAR image (where the noise is prevalent) to zero or to interpolate the values based on neighbouring pixels. To remove border noise in Sentinel-1, the *NoiseMask* included in the metadata, which flags areas affected by noise, is utilised. SNAP provides algorithms to remove border noise, enhancing overall image quality by filtering out low-intensity artifacts at the edges [26]. This improvement is vital for applications such as land cover mapping, where edge effects can lead to inaccuracies

Radiometric calibration converts the corrected data $\boldsymbol{x_{border_{corrected}}(i,k)}$ to the backscattering coefficient $\sigma^0(i,k)$ [22], representing the radar reflectivity of target surfaces:

$$\sigma^0(i,k) = \frac{x_{border\_corrected}(i,k)}{A} \cdot \frac{C^2}{R^2(i,k)} \cdot \cos(\alpha(i,k))$$

Where A and C are calibration factors, $R(i,k)$ is the slant range distance and $\alpha(i,k)$ is the local incidence angle for the pixel (i,k). For Sentinel-1 data:

$$\sigma^0(i,k) = x_{border\_corrected}(i,k) \cdot CalibrationFactor(i,k)$$

The **geometric terrain correction** aligns pixels to geographic coordinates using a DEM, correcting distortions like foreshortening and shadowing [27]:

$$\sigma^0{}_{corrected}(i,k) = \frac{\sigma^0{}_{original}(i,k)}{\cos(\alpha(i,k))},$$

Pre-processing ensures high-quality SAR data for applications such as land cover classification and environmental monitoring, using tools like the European Space Agency (ESA) Sentinel-1 Toolbox in SNAP. Further details are available in the SNAP Wiki [24].

4.1 Application of the Despeckling Algorithm

SAR images are inherently affected by speckle, a "noise-like" signal that arises from the coherent nature of electromagnetic scattering [28]. Although speckle contains some information about the illuminated surface, it degrades image quality and impairs the performance of scene analysis tasks, such as segmentation and classification, typically carried out by automated systems. To address this issue, a range of filtering techniques have been developed to reduce speckle significantly while preserving important scene features, including radiometric and textural information.

Speckle is a multiplicative noise, meaning its intensity is proportional to the local grey level of the image. Therefore, speckle filtering is essential to suppress noise and enable better interpretation and analysis of backscatter data. However, it is crucial to recognize that speckle filtering can also unintentionally remove valuable information related to key land surface characteristics, such as soil moisture, biomass, and flood extent. Thus, the goal of an effective speckle filter is to minimize noise without sacrificing important image structures.
There are several techniques for speckle removal, and each method involves a trade-off between noise suppression and preserving spatial resolution. One of the earliest and most widely used methods is the Lee filter [29], which was designed to reduce speckle while retaining essential features [30].

Another advanced technique is time-series-based processing, which leverages a sequence of SAR images captured over time. In recent years, multitemporal despeckling has emerged as a more effective approach, exploiting time-series data to address spatial denoising challenges while preserving spatial resolution. This method benefits from the increasing availability of SAR time-series, and it is commonly implemented in advanced processing pipelines, such as those available in Docker containers that support both the classical Lee filter and more sophisticated multitemporal filters. These multitemporal techniques are particularly useful in applications where spatial detail is critical and must be preserved while reducing noise over a series of observations.

In summary, speckle removal is a critical step for improving SAR image interpretability, but it requires balancing noise suppression with the preservation of key scene features. The use of multitemporal methods represents a significant advancement, offering enhanced results compared to traditional single-image techniques.
Lee speckle filtering

4.1.1 Lee Speckle Filtering

The Lee filter is an adaptive filtering technique specifically designed to reduce speckle noise in Synthetic Aperture Radar (SAR) images. It is the first model-based filter for this purpose, based on the Minimum Mean-Square Error (MMSE) algorithm. By transforming multiplicative speckle noise into additive noise, the Lee filter facilitates more effective analysis. Local statistics, such as mean and variance, are computed within a user-defined window to determine the new intensity value $\hat{x}(i,k)$ for each pixel (i,k) is determined using:

$$\hat{x}(i,k) = \mu(i,k) + \omega(i,k) \cdot (x(i,k) - \mu(i,k)) \,,$$

Where $\mu(i,k)$ represent the local mean at pixel (i,k), and $\omega(i,k)$ is the weighting factor given by:

$$\omega(i,k) = \frac{\sigma^2(i,k)}{\sigma^2(i,k)+\eta^2} \,,$$

Here, $\sigma^2(i,k)$ is the local variance of the pixel (i,k), and η^2 is the noise variance, which is assumed constant across the image and nd is determined by the Equivalent Number of Looks (ENL):

$$\eta^2 = \frac{1}{ENL} \,.$$

ENL reflects the level of averaging applied to mitigate speckle noise and influences the filter's smoothing effects; a higher ENL results in more aggressive smoothing, while a lower ENL retains more detail but some speckle. Users

can experimentally adjust ENL to balance noise suppression with image detail preservation, making the Lee filter adaptable for various SAR image characteristics and applications.

Recent research has highlighted the effectiveness of the Lee filter in enhancing SAR image quality by improving speckle suppression while maintaining spatial detail. Studies suggest that modifying the window size of the filter based on input image characteristics can significantly enhance speckle reduction. One study employed neural networks to predict optimal filtering parameters, leading to improved image quality and reduced speckle in Sentinel-1 SAR images [30].

Several investigations have explored combining the Lee filter with advanced techniques. For instance, integrating the Lee filter with non-linear diffusion and fusion-based thresholding methods has shown effective speckle suppression while preserving edge details, outperforming traditional filtering techniques on various metrics [31]. Another approach utilised discrete wavelet transforms alongside the Lee filter, achieving effective noise reduction while maintaining crucial image features, surpassing conventional methods [32].

Studies [33] and [34] indicate that using a moving kernel size of 5x5 or 7x7 achieves an optimal balance between speckle suppression and the preservation of image details and textures. The Lee filter is recognised for its ability to maintain prominent edges, linear features, point targets, and texture information, achieved by minimising mean square error or using weighted least squares estimation techniques.

4.1.2 Multi-Look Speckle Filtering

Multi-look processing is a prevalent technique in Synthetic Aperture Radar (SAR) imaging, renowned for its effectiveness in enhancing image quality by reducing speckle noise, an inherent granular disturbance that complicates fine detail interpretation. This process involves averaging multiple independent views of the same scene, either in the range (horizontal) or azimuth (vertical) direction, or both, resulting in a smoother and more coherent image. However, this averaging leads to a trade-off: while the image becomes less grainy, its spatial resolution diminishes, causing fine details to be slightly blurred. This compromise is often acceptable, particularly for applications such as terrain mapping, object detection, and environmental monitoring, allowing for flexibility depending on the desired outcome of SAR image analysis.

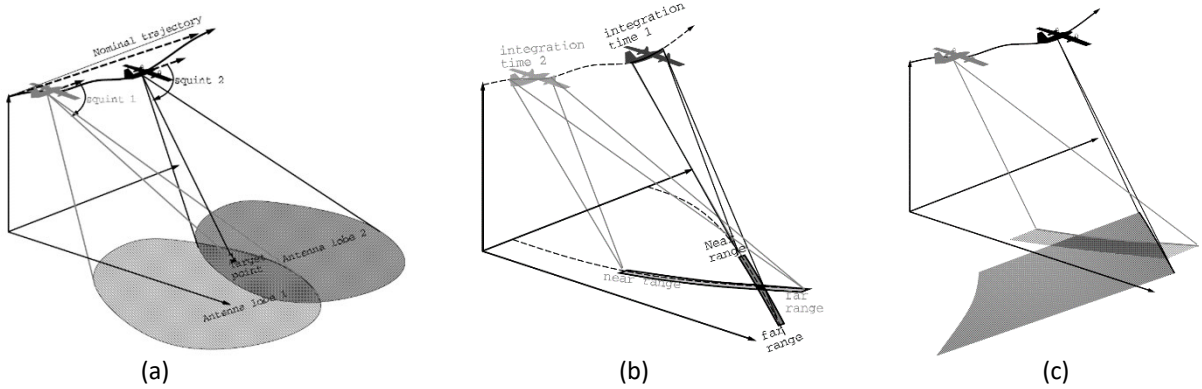


Figure 11. Principle of multi-look processing (a), acquiring a point on the ground from separated integration intervals (synthetic antennae) (b) and corresponding single-look images with range axis oriented along different squint angles (c).

In SAR imaging, extended illumination occurs because of the low directivity of the radar antenna, causing specific ground points to be illuminated for durations significantly exceeding the integration time. As shown in Figure 11(a) a ground point is illuminated as it moves through the antenna lobe, leading to the computation of multiple images for different integration intervals, illustrated in Figure 11(b). Due to different observation angles, the "range axis" of these single-look images does not align without proper geometric correction, especially when the platform's trajectory deviates from a straight line, necessitating an accurate geometrical model for alignment. Since each integration interval involves distinct observation angles, the "range axis" in these images, often

referred to as "single-look images," does not align without appropriate geometric correction, as demonstrated in Figure 11(c). This mismatch is particularly pronounced when the platform's trajectory deviates from a straight line, necessitating a precise geometrical model for accurate alignment of single-look images.

Multi-look processing effectively reduces speckle noise caused by diffuse reflections from rough surfaces. Speckle noise is independent across single-look images derived from non-overlapping integration intervals. By co-registering and averaging multiple single-look images within the same coordinate system, a smoother multi-look image is created. Nevertheless, certain textures, particularly those with fractal-like surfaces, may retain their graininess despite the number of looks averaged, underscoring the importance of understanding the statistical characteristics of textures in the context of speckle reduction [35].

The multi-look intensity image $x_{\mathrm{ML}}(i,k)$ is obtained by averaging the intensities of the L independent looks:

$$x_{\mathrm{ML}}(i,k) = \frac{1}{L}\sum_{n=1}^{L}|x_n(i,k)|^2$$

Where $x_n(i,k)$ is the complex value of the n-th independent look, $|x_n(i,k)|^2$ represents the intensity (squared magnitude of the complex value), and L is the total number of the independent looks.

Speckle noise follows a multiplicative noise model. The variance of speckle noise in the multi-look image, σ^2_{ML}, is reduced compared to that of the single-look image, σ^2_{SLC}, as quantified by:

$$\sigma^2_{\mathrm{ML}} = \frac{\sigma^2_{\mathrm{SLC}}}{L}$$

As L increases, speckle noise decreases, enhancing image quality but reducing spatial resolution. The relationship between the resolution of the multi-look image $\boldsymbol{X}_{\mathrm{ML}}$ and that of the single-look image $\boldsymbol{X}_{\mathrm{SLC}}$ is given by:

$$\boldsymbol{X}_{\mathrm{ML}} = \sqrt{L}\cdot\boldsymbol{X}_{\mathrm{SLC}}$$

Thus, while increasing the number of looks results in a smoother image with reduced noise, it simultaneously reduces spatial resolution. In summary, multi-look processing creates an intensity image by averaging multiple independent looks, effectively mitigating speckle noise at the cost of spatial resolution. This technique is crucial in SAR image processing, enhancing clarity and interpretability while highlighting the trade-off between improved image quality and resolution, contingent on the number of looks employed.

4.1.3 Multi-Temporal Speckle Filtering

The multitemporal despeckling filter is a denoising approach that leverages a ratio-based framework for processing multitemporal SAR data, the RABASAR method, which stands for Ratio-Based Multi-temporal SAR Images Denoising. Instead of working directly on the noisy images, it computes a ratio image by dividing each noisy image by the temporal mean of the entire stack. This ratio image exhibits improved stationarity compared to individual noisy images, making it easier to reduce noise effectively.
One of the key advantages of this method is that it better preserves thin structures that remain consistent across time, thanks to the multitemporal averaging. These stable features are maintained with greater accuracy, preventing them from being smoothed out during the denoising process [36].

Furthermore, because the ratio images have more uniform statistical properties, applying speckle-reduction techniques to these images yields better results than directly processing the original noisy images in the temporal stack. Another benefit is that the amount of data to be processed is reduced by creating a “super-image”, which sums up the essential information from the entire temporal stack. This allows the framework to more efficiently exploit the relevant content in the data, leading to both enhanced noise reduction and preservation of critical

image details across the stack.
The process can be broken down into three key steps:

1. **Super-Image Calculation**: The first step involves generating a "super-image," which is essentially the result of averaging a series of SAR images over time. This reduces speckle noise while maintaining the important spatial details of the image. The super-image can be spatially filtered to further suppress speckle noise.
2. **Ratio Image Creation**: Once the super-image is calculated, the next step is forming the ratio between the noisy SAR image and the super-image. This ratio simplifies the noise structure, making it easier to denoise compared to directly denoising the original image. The ratio image primarily retains residual speckle, which is easier to address due to its stationarity.
3. **Reconstruction**: After denoising the ratio image using conventional speckle reduction methods, the final denoised image is reconstructed by multiplying the denoised ratio image with the super-image. This method effectively suppresses speckle noise while preserving crucial geometrical and radiometric information.

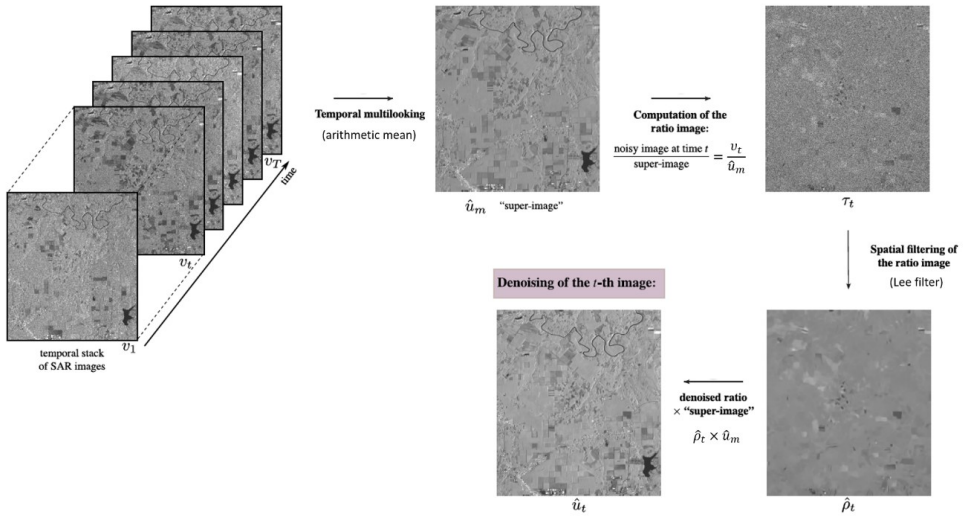


Figure 12. Overall process of the multitemporal despeckling method as applied to SAR time series. It visually summarizes the key steps, including the creation of a super-image from temporally averaged SAR data, the generation of ratio images, and the final denoising process. This method leverages both temporal and spatial information to effectively reduce speckle noise while preserving critical structural details across the time series.

RABASAR outperforms many other techniques by preserving fine details, such as temporally stable thin structures, while achieving a good balance between noise suppression and detail preservation. It has been tested on simulated and real SAR data (e.g., Sentinel-1 and TerraSAR-X), showing improvements over other state-of-the-art despeckling techniques, both visually and in metrics like PSNR (Peak Signal-to-Noise Ratio) and MSSIM (Mean Structural Similarity Index). The RABASAR framework makes the processing of SAR time series more efficient by focusing on reducing speckle in ratio images instead of the entire multi-temporal stack, and it can easily adapt to new data as they become available.

According the sheme in Figure 12 and using its notation, The super-image $\hat{u}_m(i,k)$ is computed by averaging the SAR time series of spatially registered and radiometrically calibrated SAR images, reducing the speckle while preserving spatial resolution. If we have a series of T SAR images denoted by $v_t(i,k)$ (where (i,k) is the spatial location and t is the time index), the super-image is defined as:

$$\hat{u}_m(i,k) = \frac{1}{T}\sum_{t=1}^{T} v_t(i,k)$$

This corresponds to the temporal multi-looking, producing a reduced-speckle image known as the super-image.

Next, the ratio image $\tau_t(i,k)$ is calculated for each image $v_t(i,k)$ in the series by dividing the image by the super-image $\hat{u}_m(i,k)$ at each spatial location s:

$$\tau_t(i,k) = \frac{v_t(i,k)}{\hat{u}_m(i,k)}$$

This ratio image isolates the residual speckle noise between the image $v_t(i,k)$ and the super-image. The ratio image is easier to denoise as it tends toward pure speckle noise when the super-image closely approximates the true reflectivity.

The ratio image is processed using a speckle-reduction algorithm. Since the ratio image follows specific statistical properties, often modeled by a gamma distribution, the denoising step is tailored to the statistics of the ratio image. For a gamma distribution, the likelihood of the speckle noise in the ratio image can be modeled as:

$$p(\tau_t) \sim \Gamma(S, \varphi)$$

where S is the shape parameter and φ is the scale parameter.

Once the ratio image $\tau_t(i,k)$ is denoised, the final denoised SAR image $u_t(i,k)$ is recovered by multiplying the denoised ratio image with the super-image:

$$\hat{u}_t(i,k) = \hat{\tau}_t(i,k) \cdot \hat{u}_m(i,k)$$

Here, $\hat{\tau}_t(i,k)$ is the denoised ratio image, and $\hat{u}_t(i,k)$ is the final denoised image at time t.
This formulation ensures that both spatial and temporal information is efficiently used, while speckle noise is suppressed, and important image structures are preserved.

5 Training dataset

Due to the missing availability of training data during Phase 1, a lot of effort has been devoted to the preparation of photo-interpretation activity carried out to define the training sets. In order to generate a representative and informative training set, a stratified random sampling strategy was carried out to define to the prior probabilities of the land cover classes, computed according to the 2015 Copernicus Global Land Service Dynamic Land Cover map at 100 m resolution (CGLS-LC100). This first sampling was adopted to generate the photo-interpreted training points for the three static areas of Phase 1 for 2019 Africa Sahel, Amazonia and Siberia. These data collection of each area was performed by the EOS members UniGE, UniTN and UniPV, respectively. Then, this dataset served as the starting point for the definition of the photo interpreted datasets of all the historical HRLC30 products for 1990, 1995, 2000, 2005, 2010 and 2015. Using a backward approach starting from 2019, each training point in the dataset has been either confirmed or rejected in the preceding year. In Phase 2, a photo-interpretation activity will be carried out for the historical area for the 2024 HRLC30 production, and also in the extended Amazonia area for all HRLC30 years. For 2024, a similar approach to Phase 1 can be adopted except in the forward direction, *i.e.*, either confirming or rejecting 2019 training points in 2024. The following subsections describe the training dataset definition adopted in Phase 1.

5.1 Photo-interpreted training sets generation

Operational land cover map production over large areas cannot rely on field campaigns because huge amounts

of costly data have to be collected, most importantly jeopardising the timeliness of the land cover map. In order to generate the training set used to perform the supervised classification of the considered study areas, a lot of effort has been devoted to photo-interpretation activities. Hence, even though existing thematic products represent a valid source of information, ground reference data are needed to model complex classes (e.g., aquatic vegetation types and seasonal shrubs) which require reliable samples that cannot be extracted from the outdated coarse thematic products. Although extremely complex and time consuming the reference data allows the production of high-quality training set which matches the definition of the legend and corresponds to the exact same time frame (see Figure 13 and Figure 14).

To properly generate the training set, which is representative of the considered area, the team first estimated the prior probabilities of the classes by considering the information provided by the CGLS-LC100 map. Then, the samples to be labelled, were selected according to the stratified random sampling strategy. The label of each sample was defined by photointerpretation of both S-2 data and SPOT images in the RR areas. For areas where SPOT images were not available, we exploited the public very high-resolution (VHR) Google and ESRI images (i.e., 50 cm). The labels of the first level of hierarchy are assigned according to the rules presented in Figure 15. In particular, the data were pixel-wise labelled, thus we avoided the strong positive correlation between samples units, which is the case for polygon-wise labelling.

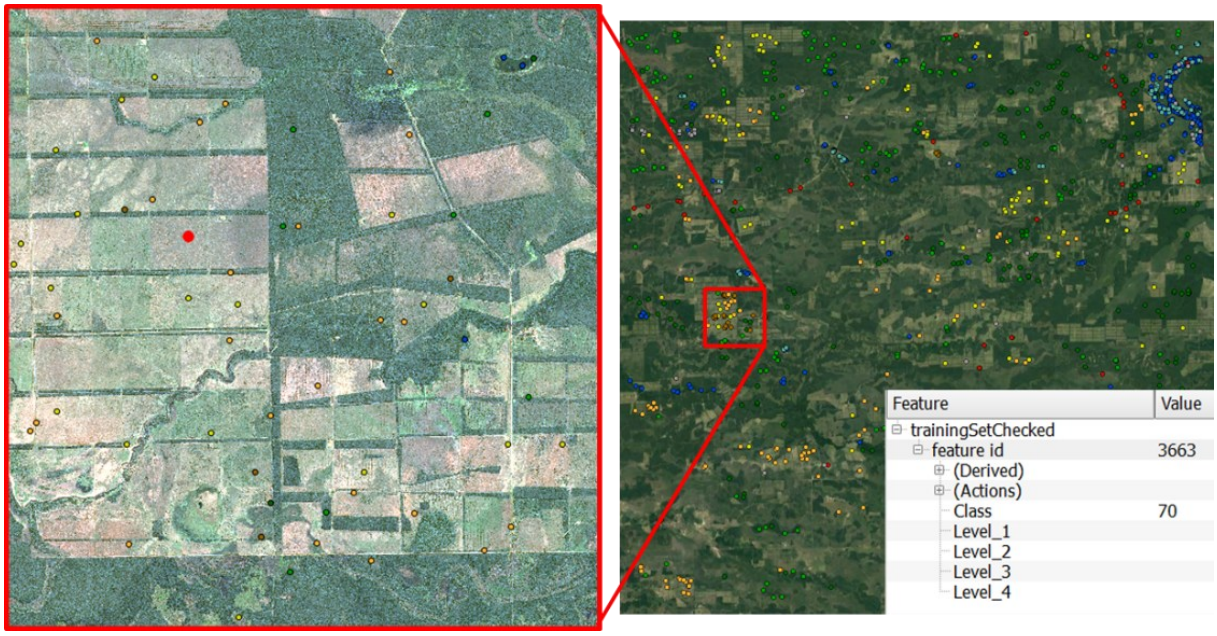


Figure 13. Training Set Production conducted via photointerpretation.

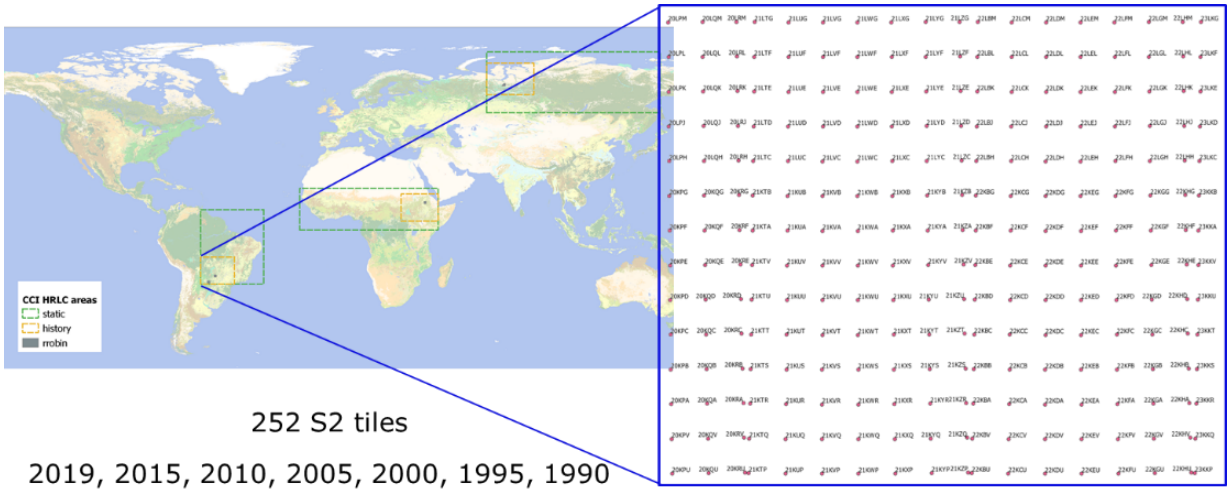


Figure 14. Example of number of tiles to be covered by photointerpretation in Amazonia during Phase 1. In Phase 2, the photointerpretation is being spatially extended according to the Amazonia area extension and updated for 2024.

Although the photointerpretation represents a valid solution for generating the training set, the legend scheme reported in Figure 15 presents some discrepancy with a set of classes which can be discriminated by the considered remote sensing data. For example, it is difficult to separate shrubs and tree cover by height using HR optical imagery only. Additionally, differently from the medium resolution no mixed classes are present in the legend (*e.g.*, Mosaic herbaceous cover (>50%) / and shrub (<50%)). Although we are working at 10 m spatial resolution, the detection of shrubs in the Sentinel 2 images is challenging (see Figure 16). The identification of deciduous and evergreen shrubland is even more challenging.

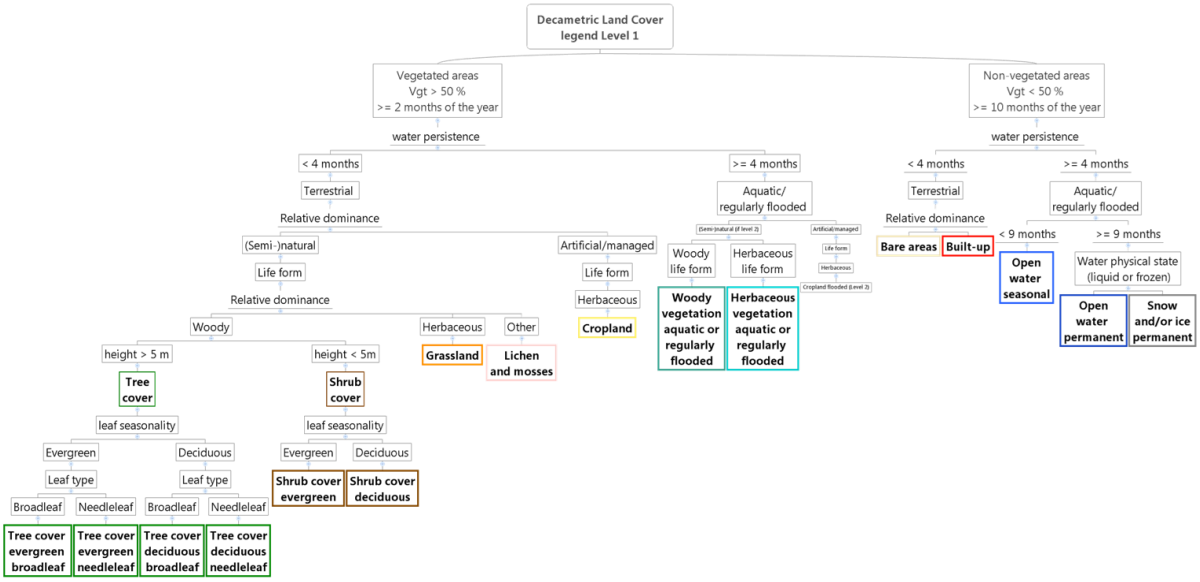


Figure 15. The classification scheme of the training-set production.

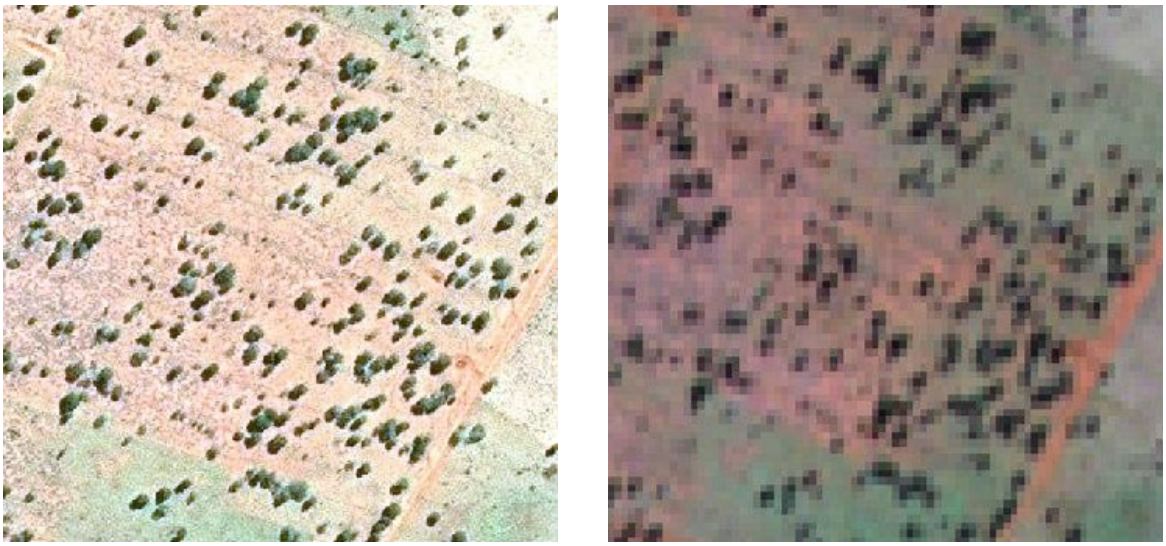


Figure 16. Differently from the medium resolution no mixed classes are present in the legend (e.g., Mosaic herbaceous cover (>50%) / and shrub (<50%). Although we are working at 10 m spatial resolution, the detection of shrubs in the Sentinel 2 images is challenging. The identification of deciduous and evergreen shrubland is even more challenging.

In the case of the historical training set photo-interpretation activity, and at the same time changing the resolution of the available images from 10 to 30 meters, the team has identified following challenges:

- less HR images are available;
- L-7 images are corrupted;

- spatial resolution of 30m hampers extraction of training points as the spectral information is often mixed. Moreover, the NDVI and NDWI trends (crucial to differentiate some very similar classes e.g. grassland vs cropland) are unclear and difficult to interpret.

Considering all the above-mentioned points, the team has decided to update the training set extracted in 2019. This means to confirm the label assigned to a sample in 2019 or otherwise to eliminate the sample from the training set. Thus, the training set produced for the years 1990-2015 have smaller number of samples compared to the one used to classify the static map. Figure 17 shows a qualitative example of the data used to perform the photointerpretation for 2005 in Africa.

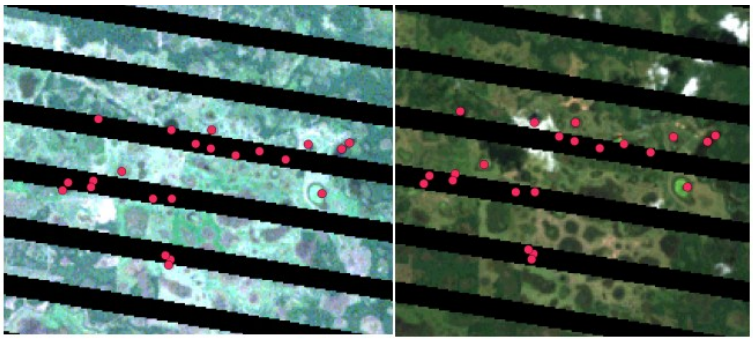
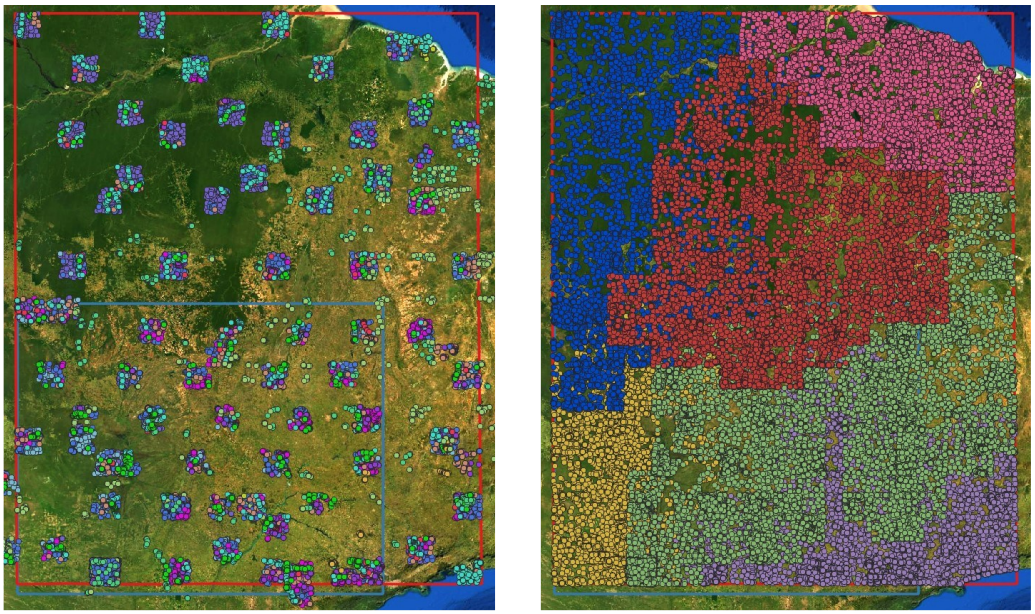


Figure 17. Many difficulties going back in the past for the photo-interpretation process: (i) less images are available; (ii) Landsat 7 Corrupted; (iii) NDVI and NDWI trend not clear; (iv) the spatial resolution of 30m.

5.2 Final static training sets generation

While complex classes require reliable samples that cannot be extracted from the outdated coarse thematic products, existing thematic products represent a valid source of information for the other classes, allowing to significantly expand the training set and properly represent the whole areas to map. For this reason, only for the static map production for 2019, we integrated the training sets delivered through photointerpretation with samples extracted from the agreement of land cover products available. Oversampling of the complex classes was performed to keep the training set prior distribution of the land-cover classes constant. Moreover, the increased amount of training labels unlocked the possibility of exploiting the specific properties of the local land cover. This can be done by considering the terrestrial ecoregions [37], which are areas of water or land that contain characteristic assemblages of natural communities and species. By training a classifier for each ecoregion, we can exploit the fact that inside an ecoregion the probability of encountering different vegetation species (which may be mapped in the same class) and communities remains relatively constant. This feature is important in land-cover mapping with remote sensing images, as it allows to mitigate the intra-class variance, a well-known issue in remote sensing.



(a) (b)

Figure 18. Amazon static: (a) photo-interpreted training set, (b) final training sets divided by ecoregions.

Therefore, we combined the photo-interpreted training sets with samples extracted from the agreement of land cover products available in MOLCA [38], and then divided each area in smaller areas defined by considering the ecoregions. This was done at tile level and by aggregating ecoregions to avoid excessive fragmentation of training set. Figure 18 shows as an example the photo-interpreted training set and the final training set of Amazonia static area, respectively. Figure 19 shows the final division into ecoregions of the three mapped areas in Phase 1. Note that the ecoregion training sets are slightly larger and overlapping with each other to guarantee consistent predictions of the land cover on the ecoregion borders.

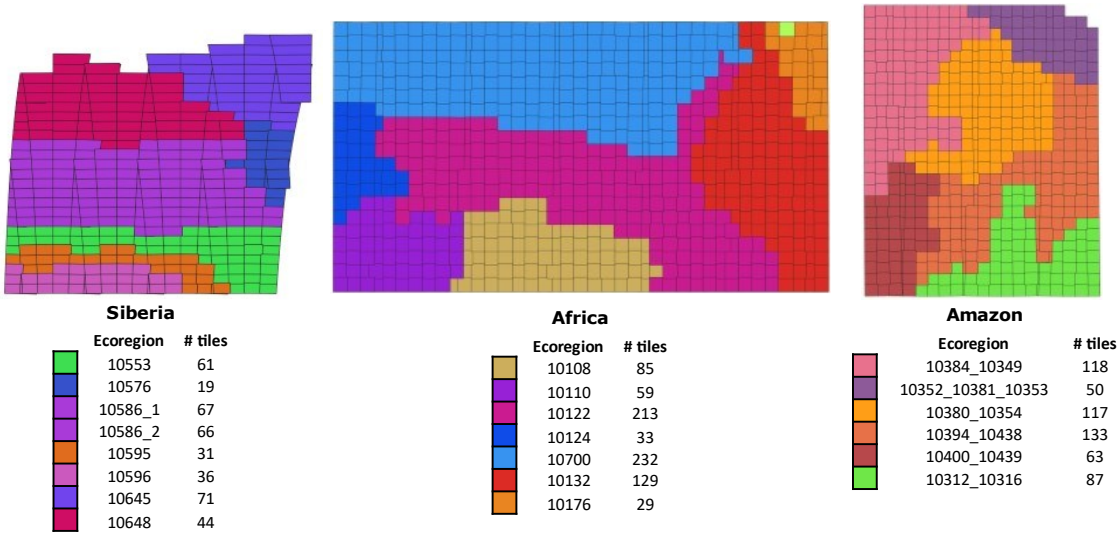


Figure 19. Final division into ecoregion for the three mapped areas.

For the historical training datasets, only confirmed photo-interpreted training points from 2019 were used to train the classifiers. For this reason, it was not possible to consider an ecoregion-based subdivision as performed for the static map. However, in Phase 2, ecoregions will be taken into consideration also for the historical production 1990-2024 in the extended Amazonia area. Given the larger extent of the historical production of Phase 2 in Amazonia, an ecoregion-based approach becomes necessary in order to adapt the model to the local characteristics of the territory.

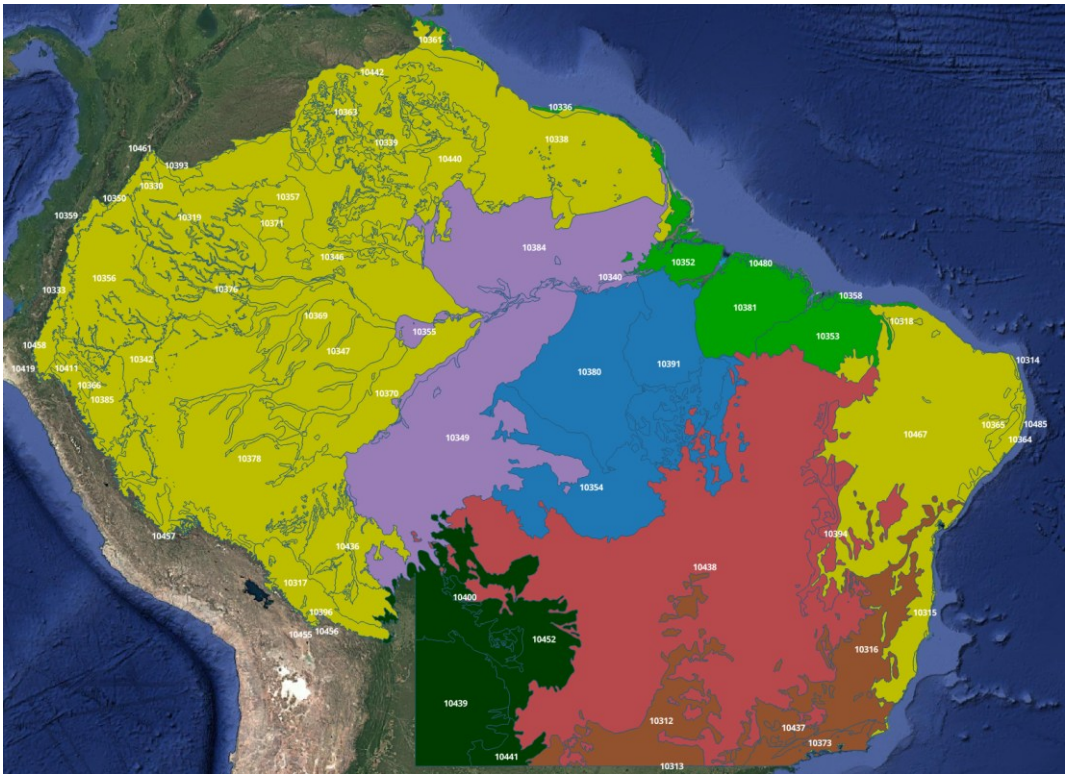


Figure 20. Terrestrial ecoregions in the south America continent. The colour represents the different ecoregion aggregation considered during Phase 1 without the MGRS S-2 tile based coarsening. The colour yellow highlights new ecoregions in the extended area (note that the exact borders are still to be defined, see PSD [AD2]), whose aggregation into larger ecoregions will be finalized once the final borders of the extended area are defined.

To this end, we are considering adopting a less coarse division of the area based on the ecoregions, which will still be aggregated to avoid excessive fragmentation. Figure 20 shows the ecoregions of the extended Amazonia area (note that the exact borders are still to be defined, see PSD [AD2]), where the yellow areas refers to new ecoregions that are going to be added and aggregated in Phase 2. The other coloured ecoregions show instead the aggregation of ecoregions that has been performed in Phase 1 within the static area.

5.3 Training Set Generation for DL algorithms applied to SAR LC classification

The Map of Land Cover Agreement (MOLCA) was used to create the training set for the SAR DL architecture in the three static areas identified in Phase 1 of the Climate Change Initiative Extension (CCI+) project: Amazonia, Africa, and Siberia. MOLCA was generated by integrating existing global High Resolution Land Cover (HRLC) maps, retaining only those areas where all datasets concur on the same land cover class while discarding areas of disagreement. These disputed pixels are marked as "no data" and set to zero in the map to prevent the model from learning erroneous relationships associated with the "no data" class, which would be both useless and misleading.

Table 6. The MOLCA classification legend that aligns with many existing high-resolution land cover datasets. It consists of nine distinct land cover classes, which help in the categorization and analysis of land use in the regions covered, including the Amazon, Africa, and Siberia.

MOLCA label	LC type	Color
20	Forest	
5	Shrubland	
7	Grassland	
8	Cropland	
9	Wetland	
11	Lichens and mosses	
12	Bareland	
13	Built-up	
15	Water	
16	Permanent ice and snow	

The MOLCA images are structured according to the S-2 L1C product tiling grid and distributed in GeoTIFF format, encompassing approximately 117 billion pixels at a resolution of 10 meters. This dataset was produced as part of the CCI+ Phase 1 project.

The land cover classes represented in MOLCA are detailed in Table 6, covering the period from 2016 to 2020. The accuracy estimate for MOLCA indicates an overall accuracy (OA) of 96% [38].

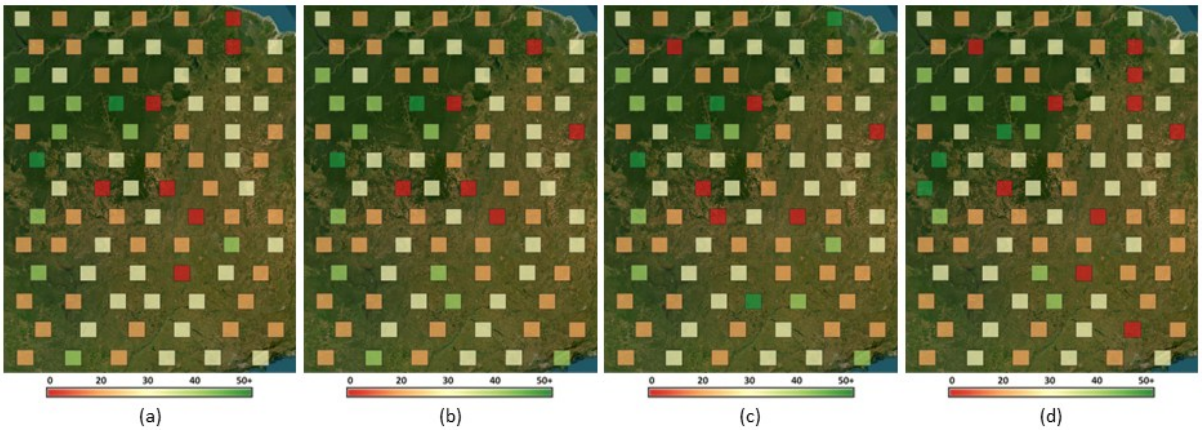


Figure 21. The seasonal distribution of Sentinel-1 acquisitions for 2021 corresponding to the selected Sentinel-2 tiles in the Amazon region is illustrated for each season: (a) winter, (b) spring, (c) summer, and (d) autumn. This distribution highlights the varying availability of SAR data across different times of the year, which is crucial for accurate land cover classification and analysis.

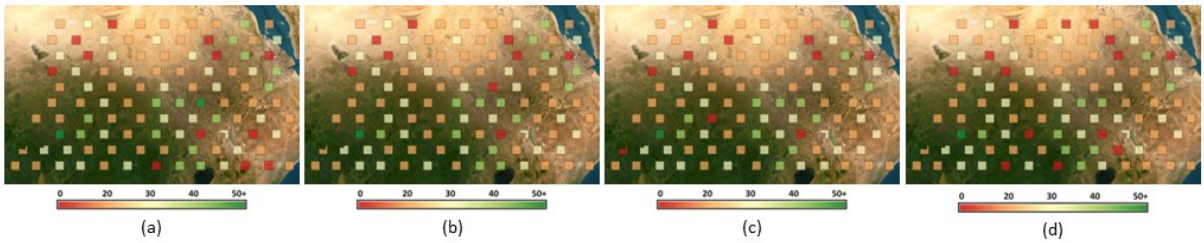


Figure 22. The seasonal distribution of Sentinel-1 acquisitions for 2021 corresponding to the selected Sentinel-2 tiles in the Africa region is illustrated for each season: (a) winter, (b) spring, (c) summer, and (d) autumn. This distribution highlights the varying availability of SAR data across different times of the year, which is crucial for accurate land cover classification and analysis.

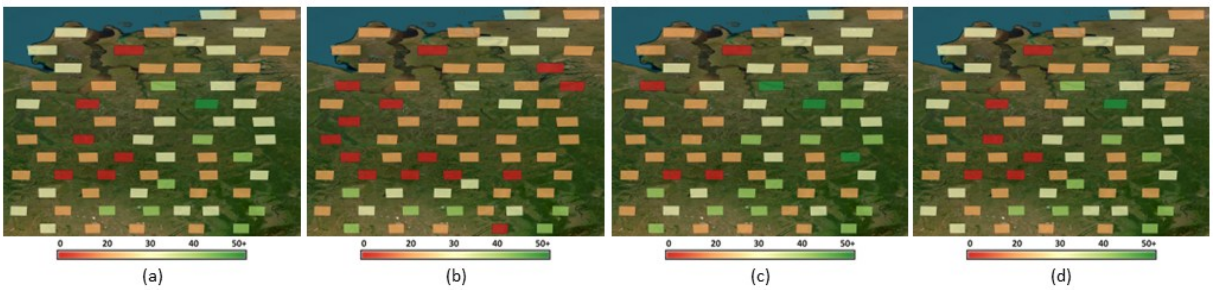


Figure 23. The seasonal distribution of Sentinel-1 acquisitions for 2021 corresponding to the selected Sentinel-2 tiles in the Siberia region is illustrated for each season: (a) winter, (b) spring, (c) summer, and (d) autumn. This distribution highlights the varying availability of SAR data across different times of the year, which is crucial for accurate land cover classification and analysis.

To ensure a significant training dataset, the test areas were randomly and uniformly sampled according to the Sentinel-2 tiling scheme, with the spatial coverage displayed in Figure 21, Figure 22, and Figure 23 for Amazonia, Africa, and Siberia, respectively. Each tile, measuring 10980 × 10980 pixels in the UTM coordinate reference system, was subdivided into smaller areas of 549 × 549 pixels, representing 1/20th of the tile's linear dimensions. The most significant patches, defined as those containing the largest number of land cover classes, were selected through visual inspection for each tile and region. This ensured a balanced representation of the land cover classes present in the scenes. Special attention was given to including samples from classes that appear in only

a few patches, such as lichens, mosses, and permanent ice in the selected Siberian region.

Once the most representative patches were identified, the corresponding Sentinel-1 features were computed following the methodology outlined in the previous section. The seasonal spatial distributions concerning the availability of 2021 Sentinel-1 acquisitions are illustrated in Figure 21, Figure 22, and Figure 23 for Amazonia, Africa, and Siberia, respectively. The colour map used in the graphs indicates varying acquisition scenarios, ranging from 5-10 images (red) to more than 50 acquisitions (dark green). Despite the presence of red tiles in each season, the number of acquisitions is sufficient to carry out spatio-temporal feature extraction [39].

The final training sets comprise 86 MOLCA patches for Amazonia, 103 MOLCA patches for Africa and 64 MOLCA patches for Siberia.

6 Multi-sensor geolocation

In the CCI+ HRLC pipeline, the multi-sensor geolocation is applied to the outputs from the optical and SAR pre-processing chains to align the data from both chains spatially. During the Phase 1, the effectiveness of this processor was confirmed by its extensive validation (not only in this multi-sensor optical-SAR application but also in its use within the SAR pre-processing chain). For this reason, no modification is planned in Phase 2 for the multi-sensor geolocation module, which is confirmed in its formulation developed in Phase 1. Accordingly, the detail of the corresponding algorithms (information-theoretic area-based registration, direct maximization method, tiling-based processing) can be found in the latest version of the ATBD of Phase 1 [AD4].

7 Optical data classification

For the classification step in the optical processing chain, the main challenges in Phase 1 were defined by i) the scarcity of available photo-interpreted data able to properly characterize the large areas that need to be mapped, ii) the considered input features, and iii) the optimization and efficiency of the considered classification algorithm. Therefore, Phase 2 is focusing on the improvement of the overall classification pipeline. In the following, the optical classification pipeline is described in detail, with the addition of the information related to the ongoing Phase 2 activities.

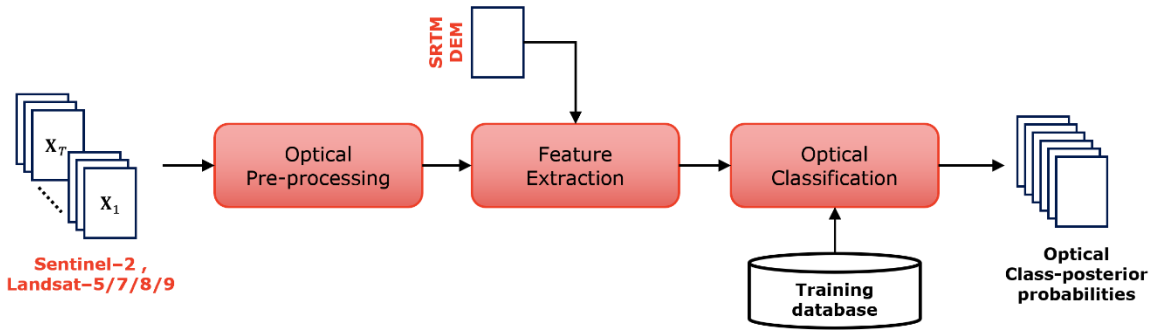


Figure 24. Optical data processing chain for the prototype production of both the static and the historical HRLC maps obtained by classifying the time series of HR optical data.

Figure 24 depicts the optical data processing chain for the production of both the static and the historical HRLC maps obtained by classifying the time series of S-2 and Landsat data. The images are first pre-processed in order to generate the optical composites. Then, the composites are combined with ancillary data (*i.e.*, SRTM DEM) to extract the final features used by the classifiers. The classifiers are first trained using the available training points and then used to generate the pixel-wise class-posterior probabilities adopted by the decision fusion processing chain to generate the final LC products.

7.1 Feature extraction

The feature extraction step aims at generating a set of representative attributes for the given pixel to maximise the ability of the classifier in detecting the correct land cover. In Phase 1, the features used as input to the classifier were the spectral bands of the time series of optical composites combined with the altitude of the pixel as given by the SRTM DEM and the textural features of the first composite. While temporal and spectral features are good in representing the seasonality of the classes, the aim of the textural and altitude features extraction is to provide to the classifier information about spatial context of the samples which can provide better land cover

discrimination. Texture allows the accurate characterization of the contextual information of a pixel in the image. In the literature, it can be found that the use of textural information can significantly improve the classification results. Hence, such features can be more distinctive than spectral features for some land cover classes. Instead of considering complex spatial features, such as shape and size, which required the unsupervised segmentation of the image, in Phase 1 we considered other textural feature extractors. First, the Gray-Level Co-Occurrence Matrix (GLCM) is computed. Then the GLCM is used to extract the following statistical measures, used as features:

- Dissimilarity;
- Correlation;
- Contrast;
- Homogeneity;
- Energy;
- Angular Second Moment (ASM).

However, during Phase 1, GLCM-based textural features have shown to be computationally demanding to generate. Therefore, we are investigating alternative strategies, such as precomputed convolutional filters or the adoption of deep learning algorithms able to inherently extract spatial features, *e.g.*, Convolutional Neural Networks (CNNs). Textures are not the only features being improved in Phase 2. Focus will also be given to the extraction of better spectral and topographical features. The former relies on the use of NDIs, which can be defined by using two spectral bands b_1, b_2 of the optical composites $\mathbf{X}^{Com}$ as follows:

$$\mathbf{NDI}^{Com}(b_1, b_2) = \frac{\mathbf{X}^{Com}(b_1) - \mathbf{X}^{Com}(b_2)}{\mathbf{X}^{Com}(b_1) + \mathbf{X}^{Com}(b_2)} \in [-1, +1].$$

The latter relies on specific topographic information that can be extracted from a DEM, *i.e.*, slope and aspect, which can be defined starting from common edge detector filters applied to the DEM.

7.2 Classification

Once features are extracted and the training datasets defined, for each training set a supervised classification model is trained. Then, each model is used to generate the class-posterior probabilities of the corresponding area (or ecoregion) and year. Given a feature vector $\boldsymbol{x}$ for a given pixel, the objective is to train a classifier that predicts the class posterior probabilities $P(\ell|\boldsymbol{x})$ for each land cover $\ell = 1, \dots, \mathcal{C}$, where $\mathcal{C}$ is the number of land covers and $\sum_{\ell=1}^{\mathcal{C}} P(\ell|\boldsymbol{x}) = 1$. Ideally, $P(\ell|\boldsymbol{x})$ represents the probability of land cover ℓ given observed featured vector $\boldsymbol{x}$. Many statistical-based machine learning methods rely on the approximation of $P(\ell|\boldsymbol{x})$. The most common approach to achieve this is to train the model by minimizing the cross-entropy loss on the training set. Let $\mathcal{D} = \{(\boldsymbol{x}_i, \ell_i) | i = 1, \dots, N\}$ be a training set where for each sample i we observe a feature vector $\boldsymbol{x}_i$ and a land cover label ℓ_i. Then, let $P(\hat{\ell}|\boldsymbol{x}; \boldsymbol{\vartheta})$ be the predicted optical class-posterior probabilities from a model parametrized by $\boldsymbol{\vartheta}$. The model can be trained by minimizing the empirical risk $\mathcal{R}_{\mathcal{D}}(\boldsymbol{\vartheta})$ over the training set $\mathcal{D}$ with the cross-entropy loss, where the empirical risk is defined as follows:

$$\mathcal{R}_{\mathcal{D}}(\boldsymbol{\vartheta}) = -\sum_{i=1}^{N} \log P(\hat{\ell}_i|\boldsymbol{x}_i; \boldsymbol{\vartheta}).$$

In the case the chosen classification strategy does not rely on the approximation of $P(\ell|\boldsymbol{x})$, the class-posterior probabilities can still be estimated by means of probability calibration strategies, *e.g.*, Platt scaling or Isotonic Regression. They train a logistic regression model and a non-parametric regression model on top of the decision function scores of the base model to predict $P(\hat{\ell}|\boldsymbol{x})$, respectively.

During Phase 1, the final choice resulted from the algorithm selection phase was the use of Support Vector Machines (SVMs), which do not estimate the class-posterior probabilities directly, later estimated using an Isotonic Regression model. SVMs have shown to be the optimal candidate for the optical classification given data and compute constraints, related to the use of GPUs, limiting the possibility of working with deep learning (DL) strategies. However, GPU availability is now increasing. Thus, in Phase 2, the classification strategy to adopt in the optical processing chain is being re-evaluated. Several deep learning approaches are being considered in our analysis. Focus will be given to strategies for properly handling intra-annual TS of composites [40], but also to multi-year classification for temporally consistent classifications [41], [42]. The considered models will be compared both in terms of performance and inference time. Indeed, focus will be given to the optimization and efficiency of the model inference step, to allow faster generation of optical land cover maps. The following subsections will provide details on the considered methods as well as on the currently adopted SVM classifier.

7.2.1 Deep Learning Approaches

In Phase 2, the main deep learning methods being studied are multitemporal architecture able to model the temporal aspects of the SITS observations. Among them, we identified the following architectures [40].

- **Recurrent Neural Networks (RNNs) with Long Short-Term Memory (LSTM) cell architecture** [43]. This

network encodes a SITS to increasingly higher-level d-dimensional representations through many cascaded bidirectional LSTM layers. Each LSTM layer processes the TS processed by the previous one, using gates (input, forget, and output gates) to control the flow of information. This allows the network to retain important temporal features and discard irrelevant ones, which is crucial for capturing seasonal or periodic changes in land cover.

- **Encoder part of a Transformer** [44]. The Encoder leverages self-attention mechanisms to capture both temporal dependencies and global context more efficiently compared to traditional RNNs like LSTMs. Transformer models, originally designed for natural language processing tasks, have shown exceptional performance in sequential data modelling, and they have been adapted to handle time series data, including satellite imagery. The key component is the Self-Attention mechanism, which allows the model to focus on different parts of the SITS when learning representations for each composite. Instead of processing data sequentially, the self-attention mechanism computes the relationships between all the time steps in the series simultaneously, making it more efficient in capturing long-term dependencies and global patterns than traditional RNNs.
- **Temporal CNNs (TempCNN)** [45]. TempCNN is a lightweight architecture composed of sequential 1D convolutional layers followed by ReLU activation functions and Dropout layers. The 1D convolutions are applied pixel-wise along the temporal dimension, which allows the model to learn temporal patterns specific to each land cover.
- **DUal view Point deep Learning architecture for time series classificatiOn (DuPLO)** [46]. DuPLO is a complex DL model designed for crop type classification from sequences of small satellite images of five-by-five pixels. It consists of two streams. A three-layer CNN stream uses 2D convolutions to aggregate spatial features independently of time. The second stream implements a 2DCNN encoder and monodirectional RNN layer implemented by a Gated Recurrent Unit (GRU) for temporal characteristics.

Among these, only DuPLO is originally designed to manage both spatial and temporal information. As anticipated in Section 7.1, we are considering alternatives to GLCM features. With DuPLO, it is possible to learn convolutional filters able to extract this type of information. In order to not exclude any of the other approaches, we are also investigating the use of few convolutional layers as first layers of the other considered architectures, thus effectively making all of them able to exploit both spatial and temporal characteristics of the SITS. The best candidate architectures will be selected based on internal benchmarking on some selected S-2 MGRS tiles in the different areas. Additional details will be provided in the context of the deliverable D2.1 Product Validation and Algorithm Selection Report.

7.2.2 Weakly Supervised Learning

Given the complexity of the considered classification problem, the training of the classifiers can be performed in a completely supervised, a partially supervised (or semi-supervised) and an unsupervised framework. In Phase 2 of the project, Weakly Supervised Learning (WSL) [47] is being considered. WSL stands in between complete supervision and partial supervision and is based on the use of unreliable sources of training labels. In the context of the project, WSL can be used to leverage obsolete maps as an additional source of labels [48], [49], [50]. In Phase 1, the training set was augmented using part of the maps intercomparison activities (*i.e.*, MOLCA), which provided weak training labels where the available land cover maps agreed. While this was shown to be helpful, there is still room for improvement. Indeed, labels produced in this way tend to be biased towards "easy" samples, thus providing little help in points where existing maps disagree. Instead, WSL provides a framework where all the available labels (not only the maps agreement) can be exploited, and the uncertainty of the label can be considered during training to guide its effect on the learning process. In the framework of Phase 2, this allows to use a much larger pool of reference (weak) labels for training our classification models. Therefore, available land-cover products can be used not only for the production of the static HRLC10 maps, but also for the historical HRLC30 maps, providing two major benefits:

1. Ecoregions can effectively be used for training also the historical models;
2. DL solution, known for being data-hungry, can be used for both HRLC10 and HRLC30.

Let $\widetilde{\mathcal{D}} = \{(\boldsymbol{x}_i, \tilde{\ell}_i) | i = 1, \dots, N\}$ be a dataset containing N instances labelled by an inaccurate source, where $\tilde{\ell}_i$ is the weak label. Let ℓ_i be the true label of the i-th instance. We can assume that the dataset is sampled *i.i.d.* from the following joint distribution:

$$p(\boldsymbol{x}, \tilde{\ell}) = \sum_{\ell} P(\tilde{\ell}|\ell, \boldsymbol{x}) P(\ell|\boldsymbol{x}) p(\boldsymbol{x}),$$

where $P(\ell|\boldsymbol{x})$ is the true class-posterior distribution and $P(\tilde{\ell}|\ell, \boldsymbol{x})$ represents the noise process that makes the given labels inaccurate. WSL aims at training a model to be able to predict $P(\ell|\boldsymbol{x})$ despite being trained on $\widetilde{\mathcal{D}}$. To achieve these, several strategies can be adopted:

- Noise-model-free approaches: these approaches do not make any assumptions on the process

$P(\tilde{\ell}|\ell, \boldsymbol{x})$. They usually are approaches that either exploit alternative robust loss functions to the common cross entropy, aiming for robustness and overfitting reduction, or use the classification confidence feedback from the model to try sifting out wrong annotations during training [51].

- Noise-model-based approaches: these approaches tend to be more successful if they make the correct assumptions or provide the correct prior information regarding the process $P(\tilde{\ell}|\ell, \boldsymbol{x})$. Such approaches explicitly consider the noise process and exploit training strategies that avoid using the weak labels $\tilde{\ell}$ directly. In this category we find class-dependent label-noise-based approaches, which exploit the confusion matrix of the sources to model the label noise in the labels (i.e., they assume $P(\tilde{\ell}|\ell, \boldsymbol{x}) \sim P(\tilde{\ell}|\ell)$) [49], or approaches that model different aspects characteristics of the weak labels, such as semantic differences or pixel resolution differences [52].

Ongoing activities are focused on the development and comparison of deep learning classifiers trained with multisource weak labelled data as alternative to the well-established SVM classifiers, used in Phase 1. The detailed analysis on the model selection process will be thoroughly described in the context of deliverable D2.1 Product Validation and Algorithm Selection Report.

7.2.3 Support Vector Machines

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 [53]. 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 [54], [55]. 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 [55]. 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.

Using the same notation as above, let us assume that a training set is given by $\mathcal{D} = \{(\boldsymbol{x}_i, \ell_i)|i = 1, \dots, N\}$. In their basic form, SVMs perform binary classification, and ensembles of SVMs are used to perform multi-class classification. Therefore, for the sake of simplicity, let's consider the scenario where there are $\mathcal{C} = 2$ land covers, and $\ell_i \in \{+1, -1\}$ is the binary label of the sample $\boldsymbol{x}_i \in \mathbb{R}^d$. 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 $\mathcal{H}: \langle \boldsymbol{\vartheta}, \boldsymbol{x} \rangle + b = 0$, where $\langle \boldsymbol{a}, \boldsymbol{b} \rangle$ is the inner product between vectors $\boldsymbol{a}$ and $\boldsymbol{b}$. The final decision rule used to find the membership of a test sample is based on the sign of the discrimination function $f(\boldsymbol{x}) = \langle \boldsymbol{\vartheta}, \boldsymbol{x} \rangle + b$ associated to the hyperplane. Therefore, a generic sample $\boldsymbol{x}$ will be labelled according to the following rule: $\ell = \text{sign}\, f(\boldsymbol{x})$.

The training of an SVM consists in finding the position of the hyperplane $\mathcal{H}$, estimating the values of the parameter vector $\boldsymbol{\vartheta}$ and the scalar b, according to the solution of an optimization problem. From a geometrical point of view, $\boldsymbol{\vartheta}$ is a vector perpendicular to the hyperplane $\mathcal{H}$ and thus defines its orientation. The distance of the $\mathcal{H}$ to the origin is $b/\|\boldsymbol{\vartheta}\|$, while the distance of a sample $\boldsymbol{x}$ to the hyperplane is $f(\boldsymbol{x})/\|\boldsymbol{\vartheta}\|$. Let us define the

functional margin $F = \min_{i=1,\dots,N} \ell_i f(\boldsymbol{x}_i)$, and the geometric margin $G = F/\|\boldsymbol{\vartheta}\|$. 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 $\mathcal{H}$ 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 ξ_i as follows:

$$\xi_i = \max\left[0, 1 - \ell_i(\langle \boldsymbol{\vartheta}, \boldsymbol{x}_i \rangle + b)\right]$$

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:

$$\begin{cases} \min_{\boldsymbol{w},b} \left\{ \frac{1}{2}\|\boldsymbol{\vartheta}\|^2 + C\sum_{i=1}^{N} \xi_i \right\} \\ \ell_i(\langle \boldsymbol{\vartheta}, \boldsymbol{x}_i \rangle + b) \geq 1 - \xi_i, \qquad \xi_i \geq 0, \forall i = 1, \dots, N \end{cases}$$

where $C \geq 0$ is the regularization parameter that allows one to control the penalty associated to errors (if $C \to +\infty$, 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:

$$\begin{cases} \max_{\boldsymbol{\alpha}} \left\{ \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \ell_i \ell_j \alpha_i \alpha_j \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle \right\} \\ \sum_{i=1}^{N} \ell_i \alpha_i = 0, \qquad 0 \leq \alpha_i \leq C, 1 \leq i \leq N \end{cases}$$

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:

$$\begin{cases} \alpha_i \left[\ell_i(\langle \boldsymbol{\vartheta}, \boldsymbol{x}_i \rangle + b) - 1 + \xi_i\right] = 0, \qquad i = 1, \dots, N \\ \xi_i(\alpha_i - C) = 0, \qquad i = 1, \dots, N \end{cases}$$

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

$$\begin{cases} \alpha_i = 0 \Rightarrow \ell_i f(\boldsymbol{x}_i) > 1 \\ 0 < \alpha_i < C \Rightarrow \ell_i f(\boldsymbol{x}_i) = 1 \\ \alpha_i = C \Rightarrow \ell_i f(\boldsymbol{x}_i) < 1 \end{cases}$$

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 $0 < \alpha_i < C$ are called non-bound support vectors (NBSV) and lie on the margin hyperplane $\mathcal{H}_1$ or $\mathcal{H}_2$ ($\ell_i f(\boldsymbol{x}_i) = 1$).

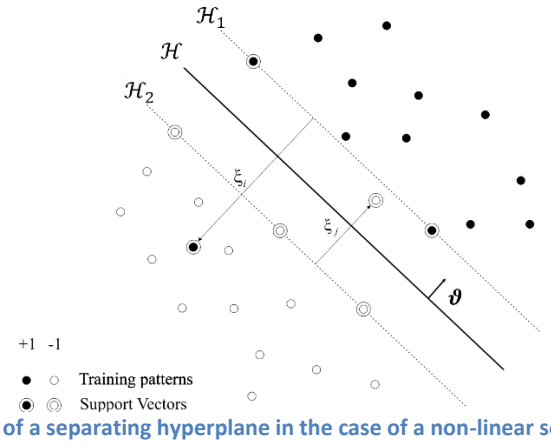


Figure 25: Qualitative example of a separating hyperplane in the case of a non-linear separable classification problem.

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 $\boldsymbol{\Phi}(\boldsymbol{x}) \in \mathbb{R}^{d'}$ $(d' > d)$, where the transformed samples can be better separated by a hyperplane (Figure 26). The main problem is to explicitly choose and calculate the function $\boldsymbol{\Phi}(\boldsymbol{x}) \in \mathbb{R}^{d'}$ for each training sample. 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:

$$\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \boldsymbol{\Phi}(\boldsymbol{x}_i), \boldsymbol{\Phi}(\boldsymbol{x}_j) \rangle, \qquad i, j \in \{1, \dots, N\},$$

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

$$\begin{cases} \max\limits_{\boldsymbol{\alpha}} \left\{ \sum\limits_{i=1}^{N} \alpha_i - \frac{1}{2} \sum\limits_{i=1}^{N} \sum\limits_{j=1}^{N} \ell_i \ell_j \alpha_i \alpha_j \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) \right\} \\ \sum\limits_{i=1}^{N} \ell_i \alpha_i = 0, \qquad 0 \le \alpha_i \le C, 1 \le i \le N \end{cases},$$

and the discrimination function becomes:

$$f(\boldsymbol{x}) = \sum_{i=1}^{N} \ell_i \alpha_i^* \mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}) + b,$$

where only support vectors (*i.e.*, $\{\boldsymbol{x}_i | \alpha_i^* \neq 0, i = 1, \dots, N\}$) contribute to (and therefore are used for) the decision. 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: $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle^p, p \in \mathbb{Z}$;
- Inhomogeneous polynomial function: $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \left(c + \langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle\right)^p, p \in \mathbb{Z}, c > 0$;
- Gaussian function (*a.k.a.* Radial Basis Function (RBF)): $\mathcal{K}(\boldsymbol{x}_i, \boldsymbol{x}_j) = e^{\frac{\|x_i - x_j\|^2}{2\sigma^2}}, \sigma \in \mathbb{R}^+$.

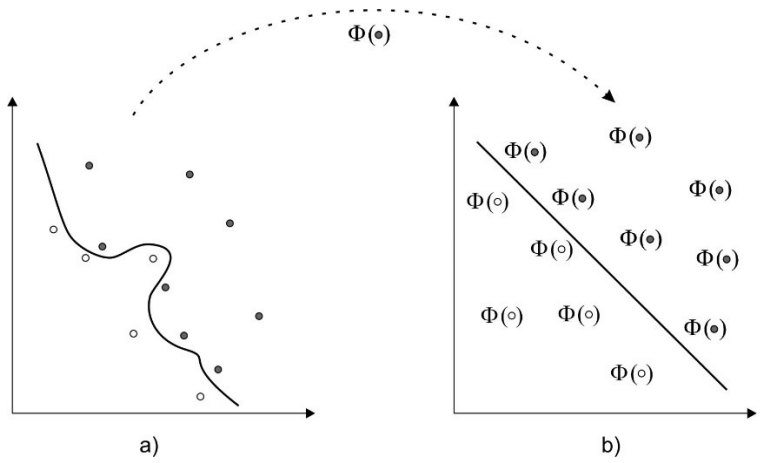


Figure 26: Transformation of the input samples by means of a kernel function into a high dimension feature space: a) Input feature space; b) kernel induced high dimensional feature space.

From an operational perspective, a possible implementation would use the RBF 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 behaviour 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 [56], optimal choice is made in proper training stage, where the meta parameters σ and C are tuned by considering the model performance during k-fold cross validation on the training set. Since the problem is multi-class classification problem with $\mathcal{C}$ land cover classes, $\mathcal{C}$ binary SVMs are trained to discriminate each land cover from the others, resulting in in $\mathcal{C}$ discrimination functions $f_\ell(\boldsymbol{x})$ with $\ell =$

$1, \ldots, \mathcal{C}$. However, $f_\ell(\boldsymbol{x})$ do not provide class-posterior probabilities. To solve this problem, Isotonic Regression is used to estimate them. It fits a left-out subset of the training set with a non-parametric isotonic regressor, which outputs a stepwise non-decreasing function that approximates the posterior probability of each land cover independently, and then normalizes them to make them sum to 1.

8 SAR Data Classification

To perform land cover classification using SAR datasets, based on the classes defined in Table 1, feature extraction will utilize the polarimetric properties of the data [57], [58]. The classification process aims to enhance the ability of the classifier to identify and distinguish various environmental textures and morphological features, such as urban areas, agricultural fields, forests, and other land cover types. This will be achieved by leveraging the amplitude of different polarization channels (e.g., HH, HV, VH, VV) and/or their combinations. Even when the SAR data used in the project (Sentinel-1, ERS and ENVISAT) are not fully polarimetric, valuable information can still be obtained from the available polarization intensities. This can be done by analyzing individual channels (e.g., selecting a specific polarization like HV or VV) or through combined metrics, such as calculating the mean, ratio, or other derived features from multiple channels. These combinations help capture essential polarimetric information, allowing the distinction between different scattering mechanisms, such as specular (mirror-like) scattering and diffuse (random) scattering. This is crucial for accurately characterizing different land cover types and improving the overall classification accuracy.

The following section outlines the SAR features that will serve as inputs to the deep learning (DL) network. It is important to note that these features are consistent with those used in Phase 1, ensuring continuity in the classification approach. These features have been carefully selected based on their effectiveness in capturing relevant information from the SAR data, facilitating robust and accurate classification results.

8.1 Feature Extraction

The proposed enhancement for SAR LC classification planned for Phase 2 aims to use a classification pipeline that incorporates a deep learning (DL) network applied to multitemporal SAR data. The approach involves segmenting the SAR time series into SAR seasonal subsequences (similar to optical composites, this approach differs by not using temporal filtering. Instead, it focuses on enhancing SAR textural features) to capture temporal variations in land cover, which can significantly impact classification performance. Spatial features are first extracted from each seasonal segment, similar to the initial step carried out in Phase 1, and then processed using a deep learning framework to identify patterns and characteristics relevant for classification.
The methodology supports a flexible approach by working on spatial subsets of the data, allowing for comprehensive geographical coverage while managing the computational complexity of large-scale datasets. By focusing on multitemporal sequences, the approach leverages temporal information to improve the discrimination of land cover types that exhibit seasonal changes, such as agricultural fields, forests, and wetlands.

To analyze and explore the spatial information contained within a single SAR image, whether using VH (vertical-horizontal) or VV (vertical-vertical) polarization, a Docker-based application has been developed. This application provides a suite of spatial domain filters designed specifically for SAR image processing. The primary criterion for selecting these algorithms was their execution speed, making them suitable for rapid application across large stacks of SAR images. While these filters may not be the most precise compared to more computationally intensive methods, their ability to be applied quickly to extensive datasets offers a significant advantage for wide-area processing, enabling efficient and timely analysis of large geographical regions.

This section describes the algorithm used for extracting SAR features in the classification process. All spatial features described below are applied to the “super image”, which is the output of the multi-temporal speckle denoising algorithm discussed in Section 4.1.3.

The features computation process involves the following steps:

1. **Pre-processing of SAR Time Series**: the SAR time series is first subjected to pre-processing methods, including techniques for filtering and de-speckling as described in Section 4. These steps aim to reduce speckle noise, improve image quality, and enhance the signal-to-noise ratio. The choice of filters balances noise suppression with the preservation of spatial details. Pre-processing also involves calibration and geo-referencing to ensure that the images are consistent and comparable across time.
2. **Seasonal Aggregation of Images**: the de-speckled images collected throughout the year are grouped based on seasons (e.g., winter, spring, summer, and autumn). For each season, the images are merged into a single composite image called “super-image”. This aggregation step serves as a trade-off: it helps retain multitemporal information critical for distinguishing seasonal variations in land cover while also reducing the computational load for the subsequent classification. The seasonal averaging helps to smooth out short-term variations in the data and enhances features that exhibit seasonal consistency, such as vegetation growth cycles.
3. **Feature Extraction from Multitemporal Sequence**: once the seasonal composite images are generated, features are computed from the final multitemporal sequence. These features are designed to capture spatial and temporal patterns that are relevant for identifying various land cover classes, such as urban areas, water bodies, forests, and agricultural fields.

The computed features are then used as inputs for classification algorithms, which may include traditional machine learning models (e.g., Random Forest) or deep learning networks (e.g., *Attention Unet*, *Swin-Unet* or 3-Dimensional - Fully Convolutional Network (3D-FCN)) capable of learning complex patterns in the data. By leveraging both spatial and temporal characteristics, the approach aims to improve classification accuracy across diverse land cover types. These features are crucial inputs for the deep learning model, allowing it to learn complex patterns and improve classification accuracy across various land cover categories.

8.1.1 Mean Filter

The mean filter, a type of low-pass filter (LPF), is one of the simplest methods for image smoothing and is straightforward to implement. It is typically used as a convolution filter, where a kernel defines the size and shape of the neighborhood sampled to calculate the mean value. The core concept of mean filtering is to replace each pixel value in the image with the average of the pixel values within the specified neighbourhood, including the pixel itself. The filter window moves across the image pixel by pixel, covering the entire image.
The use of mean filters in SAR images has been widely studied due to their ability to reduce speckle noise, a common issue in these images. Mean filtering, specifically local mean filtering, averages pixel values within a defined neighborhood, helping to smooth out the noise while retaining some image features.
Research shows that traditional filtering methods like the mean filter can improve image quality by reducing random variations in pixel intensity caused by speckle noise. However, while mean filters are effective at averaging out noise, they may also blur significant details, particularly in high-frequency areas of the image [59], [60].
As a result, the noise becomes less noticeable, but the image appears "softened." In theory, bright and dark speckle pixels within the filter window can cancel each other out, especially as the filter window size increases (e.g., 7x7 or 9x9), which can enhance noise reduction. However, larger filter sizes also tend to blur the image, causing a loss of fine details and spatial resolution. For this reason, smaller filter sizes such as 3x3 or 5x5 are often recommended for a balance between noise reduction and detail preservation.

Mathematically, for a given pixel at coordinates (i, k) in the SAR super image $\boldsymbol{X}$, the output pixel value $x_{Mean}(i, k)$ can be defined as:

$$x_{Mean}(i,k) = \frac{1}{D} \sum_{(i',k') \in \mathcal{N}} x(i',k')$$

Where $\mathcal{N}$ is the neighborhood defined by the kernel size (e.g., 3x3, 5x5), D is the total number of pixels in the neighborhood, and (i', k') are the coordinates of the pixels in the neighbourhood around (i, k).

Mean filtering is not suitable for removing impulse noise, such as salt-and-pepper noise, where pixel values differ significantly from their surroundings. In such cases, a median filter is more effective. The median filter replaces the central pixel with the median of its neighbourhood, preserving edges and reducing noise while retaining important image features.

8.1.2 Median Filter

The median filter is widely used for noise reduction in images, like the mean filter; however, it often excels in preserving useful image details. Like the mean filter, the median filter processes each pixel individually, examining its neighbouring pixels to determine if it is representative of the surrounding area. Rather than replacing the pixel value with the mean of its neighbours, the median filter substitutes it with the median value. This method is especially effective at retaining important features such as edges, step changes, and ramps, making it suitable for tasks where edge preservation is criticaly, as it minimizes the risk of losing significant structural details while still reducing noise levels [61].

To compute the median, the pixel values in the neighbourhood are first sorted in numerical order, and the middle pixel value is then selected to replace the current pixel. This approach has two main benefits:

1. **Robustness Against Outliers**: the median is less affected by extreme values, ensuring that noise reduction is achieved without distorting the image.
2. **Edge Preservation**: As the median corresponds to one of the existing pixel values in the neighbourhood, it avoids creating unrealistic pixel values, which is beneficial for maintaining sharp edges compared to the mean filter [62].

It is important to highlight that while the median filter preserves edges, it can still result in the removal or suppression of smaller or linear features, similar to its effect on speckle noise. For example, a 3×3 median filter can effectively reduce noise but may slightly degrade the overall image quality. In contrast, a larger 7×7 median filter can completely eliminate noisy pixels, though this may cause the image to appear "blotchy." A more balanced approach is to use a 3×3 or 5×5 median filter and apply it multiple times, achieving significant noise reduction while retaining more image details [63].

For a given pixel at coordinates (i, k) in a SAR image super image $\boldsymbol{X}$, the output pixel value $x_{\text{Median}}(i, k)$ after applying a median filter with a kernel of size $W \times W$ can be mathematically expressed as:

$$x_{\text{Median}}(i, k) = \text{median}\{ x(i', k') \mid (i', k') \in \mathcal{N}(i, k) \}$$

where $\mathcal{N}(i, k)$ is the neighbourhood defined by the kernel centered at pixel (i, k), the set $\{ x(i', k') \}$ contains the pixel values within the kernel surrounding the pixel (i, k). The median function selects the middle value from the sorted pixel values in the neighbourhood.

For a 3×3 kernel, the neighbourhood includes the pixels from $(i-1, k-1)$ to $(i+1, k+1)$. If the number of pixels is odd, the median is the middle value; if even, practical implementations usually select a value from the neighbourhood rather than averaging the two central values.

In SAR data processing, the median and mean filters have limited effectiveness due to the multiplicative nature of speckle noise, which correlates with signal intensity. Both filters are non-adaptive and do not consider the specific characteristics of speckle noise. Adaptive filters like the Lee filter, which adjust based on local mean and variance within a moving window, offer more effective noise reduction tailored to SAR image characteristics.

8.1.3 Maximum and Minimum Filters

Minimum and maximum filters, known as *erosion* and *dilation* filters respectively, are morphological filters that operate on a neighbourhood (window) around each pixel, defined by a specified radius. For instance, a radius of 1 corresponds to a 3x3 window, while larger radii yield larger windows (e.g., 5x5 or 7x7). These filters are shift-invariant, meaning their effects are consistent across all pixel positions. The minimum filter (erosion) replaces the central pixel with the lowest intensity value in the neighbourhood, expanding dark areas and contracting bright regions. Conversely, the maximum filter (dilation) replaces the central pixel with the highest intensity value, expanding bright areas and reducing dark regions, thus aiding in image enhancement, feature extraction, and noise reduction [64].

Mathematically, for a pixel at (i, k) in the SAR super image $\boldsymbol{X}$, with an $W \times W$ neighbourhood $\mathcal{N}$:

- The **minimum filter** is given by $x_{\min}(i,k) = \min_{(i',k')\in\mathcal{N}} x\,(i',k')$
- The **maximum filter** is given by $x_{\max}(i,k) = \max_{(i',k')\in\mathcal{N}} x\,(i',k')$

Smaller windows (e.g., 3x3) preserve fine details, while larger windows (e.g., 5x5, 7x7) provide stronger effects, potentially connecting separate regions or removing small objects. Odd-sized windows ensure a central pixel for symmetry and ease of calculation..

In SAR imaging, these filters help manage speckle noise and enhance feature boundaries. Erosion suppresses isolated noise, while dilation highlights structural features like ridges or linear patterns, improving visibility and detail delineation.

8.1.4 Max-Min Filter

The Max-Min filter enhances image contrast by calculating the difference between the dilation and erosion (maximum and minimum) of the original image. For the SAR super image $\boldsymbol{X}$, the filtered output output $\boldsymbol{X}_{\text{max-min}}$ at pixel (i, k) is given by:

$$x_{\text{max-min}}(i,k) = x_{\max}(i,k) - x_{\min}(i,k)$$

where $\boldsymbol{X}_{\max}$ and $\boldsymbol{X}_{\min}$ are the results of applying maximum (dilation) and minimum (erosion) filters to the input image at pixel (i, k), respectively.

The Max-Min filter replaces each pixel with the difference between the highest and lowest intensity values within a specified neighbourhood, commonly using window sizes like 3x3, 5x5, or 9x9. A 9x9 window is typically used to balance smoothing while preserving spatial details.

This filter sharpens edges and highlights texture by amplifying intensity variations within local neighbourhoods, making it valuable for tasks requiring detail enhancement. In SAR imaging, it helps improve feature recognition by reducing speckle noise and accentuating edges. The Max-Min filter's relation to morphological transformations, such as gradient filtering, adds to its utility in image processing where detail and noise suppression are important.

8.2 Land Cover Classification

The classification approach employed in this work utilizes a hierarchical method to extract specific land cover classes, followed by a general classification for the remaining ones. The procedure is organized as follows:

- The process begins with isolating classes that can be easily identified using a specific subset of features. Unsupervised classification methods are employed for this purpose, focusing currently on detecting

built-up areas and water bodies, which exhibit distinctive characteristics in the data. By utilizing unsupervised techniques at this stage, the complexity of the classification task is reduced, as easily recognizable areas are pre-classified, simplifying the subsequent classification of more complex regions.

- Deep learning (DL) techniques are applied to classify the remaining land cover types based on a broader set of features. Three DL-based systems are evaluated: *Attention Unet* [65], *Swin-Unet* [66], or 3D-FCN [67]. The performance of these models will be compared to identify the most effective method for land cover classification. These DL systems utilize solely radar data, incorporating temporal and spatial synthetic features derived from annual series organized into seasonal clusters. Instead of using dense temporal image sequences, the synthetic features are input into the DL network. This methodology not only captures spatial information about the scenes but also integrates multitemporal data through seasonal partitioning, enhancing the model's ability to discern complex land cover patterns.

This structured approach effectively combines unsupervised and supervised classification methods, leveraging the strengths of each to improve overall classification accuracy in land cover mapping.

The LC information from the Map Of LC Agreement (MOLCA) [38] will be used to build the training set for the DL approach.

8.2.1 Urban EXTent (UEXT) Algorithm

The Urban EXTent (UEXT) algorithm [68] primarily focuses on identifying artificial structures, such as buildings, which manifest as bright points in multitemporally averaged and despeckled SAR image datasets. When the temporal intervals of interest contain no more than one dataset, the algorithm may operate on single SAR images.

The process begins by selecting artificial structures associated with the highest normalized backscattered power values as "seed pixels." Following this selection, an iterative flooding algorithm is applied to the neighborhood of these seed pixels until a predefined lower threshold is reached. To mitigate the risk of false positives, particularly in mountainous areas, a series of post-processing steps are executed, which also incorporate the DEM of the region.

In the approach described in [68], several intermediate steps have been streamlined to reduce computational demands. Instead of the iterative flooding technique, a single watershed algorithm has been introduced. In this refined method, the identified seed pixels serve as "markers" for the watershed algorithm, while the saliency map is generated using an occurrence data range filter applied to the SAR datasets. This filter operates with a 3x3 pixel window, carefully chosen to maintain the spatial resolution of the data.

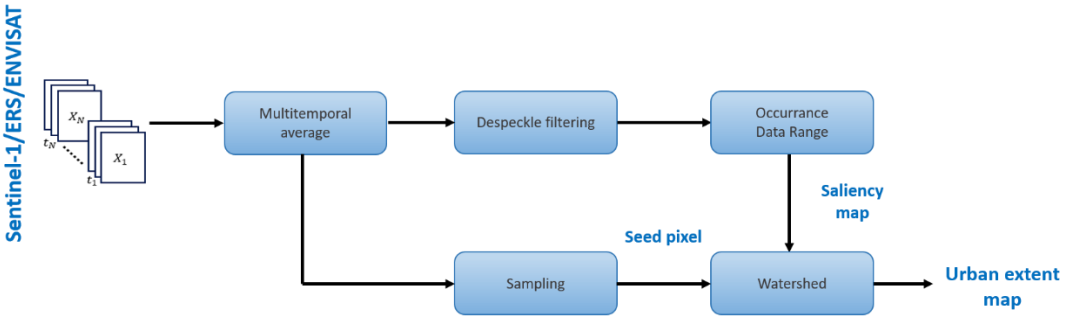


Figure 27. The block diagram workflow for the UEXT algorithm.

The workflow is visually summarized in Figure 27 and involves the following key steps:

- Temporal averaging enhances urban features.

- Seed pixels are identified by thresholding the average image.
- A saliency map is generated using a range filter.
- The watershed algorithm segments urban regions using the seed pixels as markers.
- The final output is the urban extent map.

The first step consists of computing the average across the datasets over the specified time period. Given a set of SAR images over time, let $x(i,k,t)$ represent the backscatter intensity for the pixel at coordinates (i,k) at time t. The temporal average over T SAR images is:

$$x_{\text{avg}}(i,k) = \frac{1}{T}\sum_{t=1}^{T} x(i,k,t)$$

This operation takes advantage of the coherent response of urban areas along the temporal axis, resulting in bright backscattering pixels within human settlements. In contrast, vegetated areas typically exhibit seasonal variations in backscattered values, leading to lower average values.
The resulting image is employed in two ways: first, it identifies seed pixels through hard thresholding; second, it generates a saliency map via the data range filter. This filter accentuates urban areas, and the preceding low-pass filter ensures that the map reflects homogeneous regions.

A threshold Th is applied to the averaged image $\boldsymbol{X}_{\text{avg}}$ to identify the seed pixels that correspond to bright, artificial structures:

$$\text{Seed}(i,k) = \begin{cases} 1, & x_{\text{avg}}(i,k) > Th \\ 0, & \text{otherwise} \end{cases}$$

An occurrence data range filter is applied to generate a saliency map $\text{Sal}(i,k)$, based on the range of pixel values within a predefined neighborhood:

$$\text{Sal}(i,k) = \max_t x\,(i,k,t) - \min_t x\,(i,k)$$

This filter highlights urban regions by emphasizing areas with less temporal variability.

The final stage involves the watershed algorithm, where the identified seed pixels expand within the saliency map, culminating in the production of the final urban extent map. The watershed algorithm is applied using the seed pixels as markers. A typical mathematical formulation for the watershed algorithm is based on the concept of region-growing from the markers. The saliency map $\text{Sal}(i,k)$ is used as a gradient image, and the goal is to segment it into homogeneous regions. The watershed function $W(i,k)$ partitions the image into regions corresponding to the urban areas:

$$Watershed(i,k) = \text{argmin}(\text{Sal}(i,k))$$

This function segments the saliency map based on local minima and grows regions around the seed pixels.

Finally, the urban extent map $\boldsymbol{U}$ is the result of applying the watershed algorithm on the saliency map using the identified seeds:

$$u(i,k) = \begin{cases} 1 \text{ if } the\ pixel\ (i,k)\ belongs\ to\ an\ urban\ region \\ 0\ otherwise \end{cases}$$

This output map represents the urban areas as a binary image, where urban regions are marked as 1 and non-urban regions as 0.

This methodological refinement not only enhances the efficiency of urban area detection using SAR imagery but

also maintains high accuracy in distinguishing built environments from surrounding landscapes.

8.2.2 Water Extraction Algorithm

The water extraction method applied in this project builds on the approach introduced in [69], with the main steps illustrated in Figure 28.

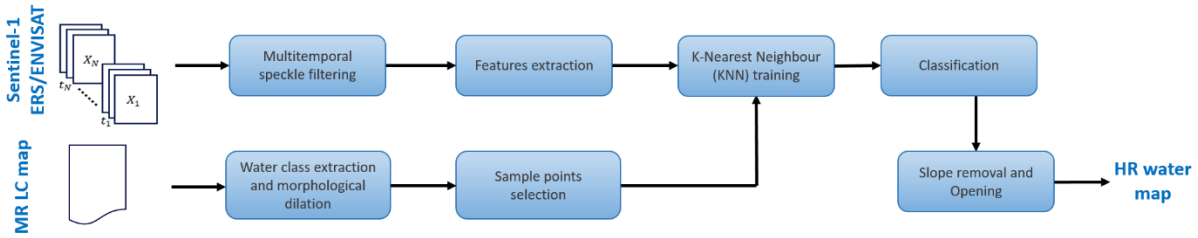


Figure 28. The comprehensive workflow for detecting both temporary and permanent water bodies from SAR imagery involves a series of sequential processing steps.

The process is designed to enhance the accuracy of water surface detection from SAR imagery, particularly by employing multitemporal denoising, feature extraction, clustering, and post-processing techniques.

Initially, the SAR image stack is subjected to a multitemporal denoising process, described in Section 4.1.3. This step reduces speckle noise across the temporal series, improving the signal quality for feature extraction.

A variety of statistical and temporal features are calculated to represent the temporal dynamics of water bodies. These features include:

- **Temporal composites**, created by averaging the SAR images over the entire temporal series or specified time windows.
- **Statistical metrics**, such as the temporal mean, minimum, maximum, and variance, which provide insights into the variation of backscatter intensity over time. The features are computed as follow:

    - *Temporal mean*: $\mu_{\mathrm{SAR}}(i,k) = \frac{1}{T}\sum_{t=1}^{T} x(i,k,t)$
    - *Temporal variance*: $\sigma_{\mathrm{SAR}}{}^2(i,k) = \frac{1}{T}\sum_{t=1}^{T}(x(i,k,t) - \mu_{\mathrm{SAR}}(i,k))^2$
    - *Minimum and maximum values over the time*: $x_{\mathrm{SAR\_min(max)}}(i,k) = \min_t\left(\max_t\right) x_{\mathrm{SAR}}\,(i,k,t)$

    where $x_{\mathrm{SAR}}(i,k,t)$ is the pixel intensity at position (i,k) for the t-th time step in the SAR image series, and T is the total number of images.

The algorithm can be improved by adding an optional step that enables the integration of SAR and optical data. This extension effectively addresses the limitations of using SAR data alone and helps resolve common misclassification issues.

SAR provides key advantages, such as cloud penetration and the ability to collect data in all weather and at night. However, it faces challenges in accurately mapping water bodies in areas with specific geomorphological features, like narrow urban rivers or flat, sandy regions, which can cause signal reflections and misclassification.

In contrast, optical data from the Sentinel-2 mission offers detailed spectral information that helps distinguish land cover types, including vegetation and water. However, it is limited by cloud cover, restricting continuous monitoring in frequently cloudy regions.

Leveraging both SAR and optical data capitalizes on their complementary strengths, significantly enhancing mapping accuracy. This integration merges features from both datasets, enhancing the model's ability to distinguish between water and non-water surfaces under diverse environmental conditions.

The expansion methodology employs a sophisticated integration of features derived from both SAR and optical sources to significantly enhance mapping precision. By combining the unique strengths of these two data types, the approach takes advantage of ability of SAR to penetrate cloud cover and provide consistent information in various weather conditions, while utilizing the high-resolution visual detail offered by optical imagery. This synergistic relationship allows for a more comprehensive analysis, enabling the accurate identification and classification of land cover types, including the distinction between water and non-water surfaces, under a wide range of environmental scenarios. Ultimately, this integration leads to improved accuracy and reliability in mapping applications. Optical features are extracted and integrated into the existing SAR feature set that was previously computed.

Let $x_{\text{opt}}(i,k,b)$ represent the optical reflectance at spatial coordinates (i,k) for a specific band b, the Normalised Difference Vegetation Index (NDVI) and Normalised Difference Water Index (NDWI) are defined as follows:

- *Normalised Difference Vegetation Index (NDVI)*: This index measures vegetation health and density by comparing the reflectance in the near-infrared (NIR) and red bands. It is useful for identifying areas where vegetation may obscure or interact with water surfaces.

$$\text{NDVI}(i,k) = \frac{x_{opt}(i,k,\text{NIR}) - x_{opt}(i,k,\text{RED})}{x_{opt}(i,k,\text{NIR}) + x_{opt}(i,k,\text{RED})}$$

- *Normalised Difference Water Index (NDWI)*: The NDWI assesses the presence of water by comparing the reflectance in the green and NIR bands. It helps to detect water bodies even in complex environments where water may be mixed with other land cover types.

$$\text{NDWI}(i,k) = \frac{x_{opt}(i,k,\text{GREEN}) - x_{opt}(i,k,\text{NIR})}{x_{opt}(i,k,\text{GREEN}) + x_{opt}(i,k,\text{NIR})}$$

Additionally, other statistical metrics are derived from the red, green, and blue (RGB) bands, such as maximum and minimum reflectance values, as well as temporal variance. These statistics are employed to improve the model's sensitivity to different land cover types and to refine the classification process.
Compute statistical measures, such as maximum $x_{\text{opt\_max}}(i,k,b)$, minimum $x_{\text{opt\_min}}(i,k,b)$, and variance $\sigma^2_{\text{opt}}(i,k,b)$ for the red, green, and blue bands over the time window T:

$$x_{\text{opt\_max}}(i,k,b) = \max_{t=1}^{T} x_{\text{opt}}(i,k,b,t)$$

$$x_{\text{opt\_min}}(i,k,b) = \min_{t=1}^{T} x_{\text{opt}}(i,k,b,t)$$

$$\sigma^2_{\text{opt}}(i,k,b) = \frac{1}{T}\sum_{t=1}^{T}\left(x_{\text{opt}}(i,k,b,t) - \mu_{opt}(i,k,b,t)\right)^2$$

where $\mu_{\text{opt}}(i,k,b,t)$ is the mean reflectance over time for band b.

Construct a combined feature vector $\boldsymbol{F}(i,k)$ at each spatial location (i,k) by concatenating features derived from both SAR and optical data:

$$\boldsymbol{F}(i,k) = \left[\mu_{\text{SAR}}(i,k), x_{\text{SAR\_min}}(i,k), x_{\text{SAR\_max}}(i,k), \text{NDVI}(i,k), \text{NDWI}(i,k), x_{\text{opt\_max}}(i,k,b), x_{\text{SAR\_min}}(i,k,b), \sigma^2_{\text{opt}}(i,k,b)\right]$$

Then the features are used to train a k-NN algorithm 4 clusters. The value of the clusters is selected to account for the expected clusters corresponding to water, vegetation, bare soil, and impervious surfaces.

$$\xi = argmin_{z\in[1,4]}\|\mathbf{F}(i,k) - \mathbf{F}_z\|$$

where ξ represents the cluster label assigned to the pixel (i,k), $\mathbf{F}(i,k)$ is the feature vector at pixel (i,k), and $\mathbf{F}_z$ is the centroid of the z-th cluster.
A 300 m resolution water mask from the ESA CCI land cover map [70] is resampled to 10 m, dilated with a 3 x 3 kernel window and used to guide the selection of training samples. From each tile, 10000 pixels representing water and non-water regions are randomly sampled for training. Once the clusters are formed, the results are compared against the water mask from the ESA-CCI map. The cluster that most closely matches the water distribution in the reference map is designated as the "water cluster."

Several refinements are applied to improve the classification accuracy. Digital Surface Model (DSM) filtering is applied by calculating the slope for each pixel. Areas with steep slopes, which may indicate radar shadows in hilly or mountainous regions, are excluded from the final water body map to improve the accuracy of the water surface detection. Then, an "opening" operation (erosion followed by dilation) is performed on the slope-filtered output:

$$\boldsymbol{W}_{map}=\text{Opening}(\boldsymbol{Bin}) = \big(\text{Erosion}(\boldsymbol{Bin},\boldsymbol{S})\big) \oplus \boldsymbol{S}$$

where $\boldsymbol{Bin}$ is the binary water mask, $\boldsymbol{S}$ is the structuring element, and $\oplus$ denotes the dilation operation. Hence, $\boldsymbol{W}_{map}$ is the final water map after the opening operation. This mechanism helps eliminate small isolated false positives without significantly affecting the extent of larger water bodies.

This water detection workflow efficiently combines multitemporal SAR information, clustering techniques, and morphological operations to produce reliable water body maps. In addition, the dual-sensor approach improves the ability of the model to accurately distinguish water bodies from other land surfaces, particularly in challenging environments. In flat, sandy regions, SAR signals can reflect away from the sensor, leading to dark images misclassified as non-water areas. By integrating optical data, the model gains additional spectral information, reducing false positives and enhancing water body mapping accuracy.

Using SAR data alone, the model struggled to detect narrow rivers in urban areas due to interference from structures and vegetation. The incorporation of optical indices provided critical spectral details, allowing better identification of these waterways.

This fusion of SAR and optical data strengthens water body mapping models, enabling more precise detection and classification across diverse landscapes. It is especially valuable for monitoring water resources, assessing climate change impacts, and informing water management policies, offering a comprehensive solution for global water monitoring efforts.

The DSM filtering and opening operations help refine the results, enhancing the classification accuracy across diverse landscapes.

8.2.2.1 Water Masking

Misclassification of bare soil as water in remote sensing imagery is a well-documented challenge in the field of remote sensing, attributable to several interrelated factors. One primary issue arises from the spectral similarities between bare soil and water, which can confound classification algorithms. Wet bare soil, in particular, exhibits reflectance properties that closely mimic those of water bodies, especially when utilizing limited spectral bands. Research indicates that smooth, wet soil can produce low backscatter responses, further complicating the distinction between these two land cover types.

In SAR imagery, the texture and moisture content of bare soil significantly influence backscatter characteristics,

often causing it to appear similar to water. For example, smooth or saturated soil surfaces can reflect radar signals in a manner that resembles water bodies. SAR systems, which are particularly sensitive to surface roughness, may misinterpret these textures, leading to classification errors.

Seasonal variations in soil conditions also contribute to this misclassification. Following rainfall, the appearance of bare soil can change dramatically, making it resemble water. This variability poses a challenge for classification algorithms that are trained on datasets not representative of such conditions.

A pertinent study [71] explored the effectiveness of Sentinel-1 SAR imagery in distinguishing various land cover types in a tropical coastal environment, specifically the Douala region in Cameroon. The research emphasized the critical role of textural analysis due to the inherent speckle noise present in SAR images, which can result in misclassification. The findings highlighted that SAR data often struggles to accurately separate certain classes, particularly between water and bare soil—two crucial categories for accurate land cover classification in coastal regions.

To mitigate the issue of misclassification, a masking operation is applied during the post-processing phase of the water map. Let W_{map} be the final binary water map output of the water detector in Section 8.2.2, the masking can be expressed as follow:

$$w_{masked}(i,k) = \begin{cases} w_{map}(i,k), & \text{if } mask(i,k) = 0 \\ 0, & \text{if } mask(i,k) = 1 \end{cases}$$

Where $mask(i,k)$ is the binary mask and $w_{masked}(x,y)$ is the binary water map at pixel (i,k) after the masking operation. The mask is generated from a LC map (e.g., ESA CCI LC at 300m, or the Copernicus Global Land Cover Layers (CGLS) [72] at 100m resolution) after identifying the specific LC value associated with the bare soil class. Here, $mask(i,k)$ = 1 for bare soil pixels and $mask(i,k)$ = 0 otherwise.

This process involves analyzing the LC map to isolate areas classified as bare soil, which helps in accurately distinguishing these regions from other land cover types. The resulting mask can then be used for further analysis, such as assessing soil erosion, land use changes, or environmental monitoring. This technique aims to enhance the accuracy of land cover classification by refining the distinction between water and bare soil, thereby improving overall interpretability and reliability of remote sensing data.

8.2.2.2 Water Dynamics Analysis: Seasonal vs. Permanent Water Identification

A dedicated module was developed to differentiate between seasonal and permanent water land cover (LC) classes. This distinction is achieved by applying the water extraction method outlined in Section 8.2.2 to a time series of monthly SAR images. As a result, a series of monthly water maps is generated, with each map representing the spatial extent of water bodies for a specific month.

To analyze the seasonal dynamics of water presence, a logical XOR (exclusive OR) operation is performed across all monthly water maps. This operation effectively highlights areas where water is present in some months but absent in others, thereby identifying seasonal water bodies. Following this, a thresholding operation is applied to the results of the XOR operation, denoted as $\boldsymbol{W}_{XOR}$.

The thresholding process is critical for establishing the criteria used to differentiate between transient (seasonal) and consistent (permanent) water presence. The classification rule can be mathematically expressed as follows:

$$w_{seasonality}(i,k) = \begin{cases} 1, & \text{if } 5 \leq \boldsymbol{W}_{XOR}(i,k) < 9\ months \\ 2, & \text{if } \boldsymbol{W}_{XOR}(i,k) \geq 9\ months \\ 0, & \text{otherwise} \end{cases}$$

In this equation, $w_{XOR}(i,k)$ represents the water presence derived from the XOR logical operation at pixel (i,k), while $\boldsymbol{W}_{seasonality}$ produces the final water classification map. The values in this final map are defined as follows: a value of 1 indicates areas of seasonal water presence, a value of 2 denotes permanent water bodies, and a value of 0 signifies non-water areas. This systematic approach allows for an accurate characterization of water dynamics over time, enhancing our understanding of hydrological processes in the study area.

8.2.3 Deep Learning Architectures

Land cover (LC) mapping is an essential tool used in various fields, including forest monitoring, agriculture, urban development, flood risk assessment, and climate change analysis. It supports the development of effective land use policies and the evaluation of ecosystem health by enabling the monitoring of environmental conditions across diverse regions.

The integration of deep learning (DL) methods, especially Convolutional Neural Networks (CNNs), has significantly advanced the field of remote sensing for LC mapping. CNNs are particularly well-suited for this task because they can directly extract local spatial features from satellite imagery [73], [74]. Architectures such as UNet, which employ encoder-decoder structures with skip connections, have proven highly effective in segmenting images while preserving spatial details.

Innovative methods also involve combining different types of neural networks to exploit their respective strengths. For example, hybrid models like Fully Convolutional Networks (FCNs) integrated with Convolutional Long Short-Term Memory (ConvLSTM) networks can combine spatial and temporal information from multitemporal SAR data, significantly improving classification accuracy over traditional approaches [75].

The following sections will discuss three deep learning (DL) architectures—Attention UNet, Swin UNet, and 3D-FCN—considered for mapping Synthetic Aperture Radar (SAR) datasets. These architectures address the unique challenges associated with SAR data, such as noise, speckle, and the complex nature of radar backscatter signals. Each architecture brings specific capabilities to enhance feature extraction, capture spatial and temporal information, and improve land cover classification accuracy.

These DL architectures leverage cutting-edge techniques to handle SAR data's distinct properties, providing robust and scalable solutions for land cover mapping across diverse environments and timeframes.

8.2.3.1 Attention Unet

The Attention Unet [65] architecture extends the traditional UNet [76] by incorporating attention mechanisms that focus on the most relevant parts of the input data. It maintains the standard structure of U-Net of a contracting path followed by an expansive path, allowing the network to capture both global context and fine details, which is especially useful for tasks such as image segmentation.
The main improvement comes from incorporating an *attention gate*, which focuses on relevant regions while suppressing irrelevant feature activations. This attention mechanism is integrated as a *skip connection* within the U-Net framework, making it particularly effective for tasks like extracting built-up areas from satellite images, where it allows the model to concentrate on distinct building patterns and structures, resulting in more accurate height estimations.

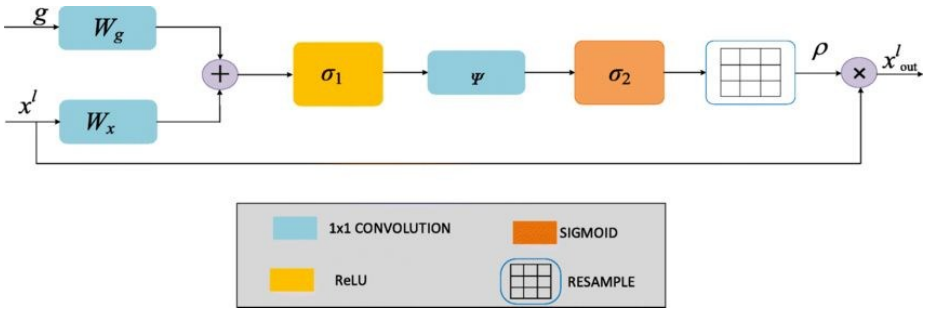


Figure 29. Architecture of an attention gate used in the Attention UNet model **[77]**.

The attention block is mathematically described according to the notation shown in Figure 29 and in the work [77], where two inputs are processed: $\boldsymbol{g}$, vector of the features from the decoder or expanded path, and $\boldsymbol{x^i}$, features from the encoder or compressed path, at a specific depth $i-th$ in the network. These feature representations undergo separate 1×1 convolutions, allowing the model to learn how to refine feature activations:

$$\widetilde{\boldsymbol{g}} = \boldsymbol{W}_g * \boldsymbol{g}$$
$$\widetilde{\boldsymbol{x}}^{\boldsymbol{i}} = \boldsymbol{W}_x * \boldsymbol{x}^i$$

Where $\boldsymbol{W}_g$ and $\boldsymbol{W}_x$ are learnable weights of the 1×1 convolution layers, and $*$ represents the convolution operation.

The outputs of these convolutions are summed element-wise to combine the refined feature maps:

$$\boldsymbol{\psi} = \widetilde{\boldsymbol{g}} + \widetilde{\boldsymbol{x}}^i$$

Next, the summed features are passed through a Rectified Linear Unit (ReLU) activation function, commonly used in deep learning models, particularly in CNNs. It introduces non-linearity into the model, enabling the network to learn complex patterns:

$$\boldsymbol{\psi}' = \mathrm{ReLU}(\boldsymbol{\psi})$$

A Batch Normalization layer is then applied to stabilize the training process:

$$\boldsymbol{\psi}'' = \mathrm{BatchNorm}(\boldsymbol{\psi}')$$

The normalized output is fed into another 1×1 convolution followed by a Sigmoid activation function to create an attention map $\boldsymbol{\phi}$:

$$\boldsymbol{\phi} = \mathcal{S}(\boldsymbol{\psi}'')$$

where $\mathcal{S}(\cdot)$ denotes the Sigmoid function. At this stage, a resampling operation is performed using an additional 1×1 convolution layer to reduce the dimensionality of the attention map, ensuring it matches the spatial dimensions of the original input features:

$$\boldsymbol{\rho} = \boldsymbol{W}_{\rho} * \boldsymbol{\phi}$$

Here, $\boldsymbol{W}_{\rho}$ represents the learnable weights of the resampling convolution.

Finally, the refined attention map $\boldsymbol{\rho}$ is multiplied element-wise with the original input features $\boldsymbol{x}^i$, selectively enhancing important regions:

$$\boldsymbol{x}^i_{\mathrm{out}} = \boldsymbol{\rho} \cdot \boldsymbol{x}^{\boldsymbol{i}}$$

The output $\boldsymbol{x}^i_{\mathrm{out}}$ is a weighted version of the input, emphasizing relevant features that are critical for the

downstream task. This attention mechanism allows the network to dynamically adjust its focus based on the input data, leading to improved segmentation performance and more accurate land cover classification, especially in complex scenarios like urban area extraction from satellite imagery.

In the context of SAR data, attention mechanisms help reduce the impact of noise and highlight significant features for land cover mapping. By dynamically weighting different regions of the image, Attention UNet can better capture subtle variations in land cover types, leading to more accurate segmentation and classification results.

8.2.3.2 Swin Unet

Swin UNet integrates Swin Transformers, a form of Vision Transformer (ViT), into the UNet framework. Unlike traditional convolutional methods, Swin Transformers utilize self-attention mechanisms to capture long-range dependencies and global context within the image. This architecture partitions the image into non-overlapping windows and performs self-attention within each window while allowing for cross-window connections.

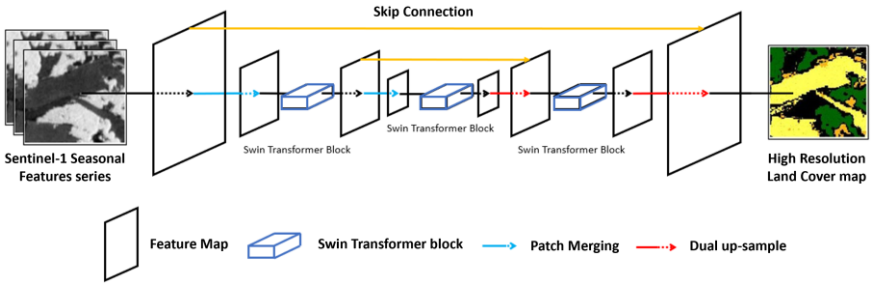


Figure 30. The Swin-UNet architecture, applied to the land cover (LC) classification task, integrates distinctive aspects of the Swin Transformer with the UNet architecture to achieve optimal performance. This combination leverages the Swin Transformer's ability to capture long-range dependencies and multi-scale contextual information, while the UNet structure ensures precise localization and segmentation capabilities.

The Swin-Unet architecture consists of three primary components: the encoder, the bottleneck, and the decoder. An illustrative diagram of the overall model is provided in Figure 30.

The **encoder** employs multiple Swin Transformer layers [78], designed to hierarchically process the input images across a series of stages. Each stage utilises the shifted window self-attention mechanism [79], allowing the model to efficiently capture local interactions in the initial stages and progressively build towards understanding larger areas of the input image. This approach reduces the resolution of feature maps while increasing feature dimensions, facilitating a deep and comprehensive understanding of the input data.

For a given input tensor $\boldsymbol{X} \in R^{Pt \times Ch}$, where Pt is the number of patches and Ch is the number of channels, the self-attention output $\boldsymbol{X}_{att\_output}$ can be formulated as:

$$\boldsymbol{X}_{att\_output} = \text{Softmax}\left(\frac{\boldsymbol{Q} \cdot \boldsymbol{Key}^T}{\sqrt{d_{key}}}\right) \cdot \boldsymbol{V}$$

where:

- $\boldsymbol{Q} = \boldsymbol{X} \cdot \boldsymbol{W}_q$, is the *Query matrix*
- $\boldsymbol{Key} = \boldsymbol{X} \cdot \boldsymbol{W}_k$ is the *Key matrix*
- $\boldsymbol{V} = \boldsymbol{X} \cdot \boldsymbol{W}_v$ is the *Value matrix*
- $\boldsymbol{W}_q, \boldsymbol{W}_k, \boldsymbol{W}_v \in R^{Ch \times d_{key}}$ are learnable weight matrices.
- d_{key} is the dimensionality of the keys.

The self-attention output, $\boldsymbol{X}_{att\_output}$, is applied during the encoder stage of the Swin UNet architecture.

Specifically, it is used in the process of self-attention within the Swin Transformer layers, where it captures local interactions and progressively builds an understanding of larger areas of the input image. This output is part of the hierarchical processing done in the encoder, where multiple Swin Transformer layers employ shifted window self-attention to analyze the input tensor, $\boldsymbol{X}$, representing the image patches and channels. The computed self-attention output helps encode information that contributes to feature extraction and down-sampling, which is essential for the network's understanding of global contextual relationships in the image.

The **bottleneck** acts as the crucial transition point between the encoder and decoder modules. It typically comprises one or more Swin Transformer layers located at the deepest part of the network, focusing on integrating and compressing the high-level features learned by the encoder.
Let $\boldsymbol{F}_{enc}$ be the feature output from the last encoder layer, the bottleneck layer can be represented as:

$$\boldsymbol{F}_{bottleneck} = \text{LayerNorm}(\boldsymbol{F}_{enc}\boldsymbol{W}_b + \boldsymbol{b}_{bias})$$

Where $\boldsymbol{W}_b$ is a weight matrix, and $\boldsymbol{b}_{bias}$ is a bias vector for the bottleneck layer.

The **decoder** then gradually expands the encoded features back to the original image resolution. It employs Swin Transformer layers arranged in stages, each incorporating a patch-expanding layer to progressively increase the spatial resolution of the feature maps. Using a patch-expanding layer, the output features from the bottleneck $\boldsymbol{F}_{bottleneck}$ are expanded:

$$\boldsymbol{F}_{dec}^{(i)} = \text{PatchExpand}(\boldsymbol{F}_{bottleneck})$$

where $\boldsymbol{F}_{dec}^{(i)}$ represents the feature maps at decoder stage i.

Additionally, skip connections from corresponding encoder stages are integrated at each level of the decoder, aiding in the restoration of spatial details often lost during down-sampling in the encoder. If $\boldsymbol{F}_{enc}^{(i)}$ is the output from the encoder at stage i:

$$\boldsymbol{F}_{dec}^{(i)} = \text{Concat}\left(\boldsymbol{F}_{dec}^{(i)}, \boldsymbol{F}_{enc}^{(i)}\right)$$

The final segmentation output $\boldsymbol{X}_{seg}$ can be produced by applying a 1×1 convolution followed by a softmax activation to the output of the last decoder layer:

$$\boldsymbol{X}_{seg} = \text{Softmax}\left(\boldsymbol{F}_{dec}^{(n)}\boldsymbol{W}_{out} + \boldsymbol{b}_{out}\right)$$

Where $\boldsymbol{W}_{out}$ and $\boldsymbol{b}_{out}$ are learnable weights and biases for the output layer.

For SAR data, Swin UNet offers an advantage in understanding global contextual relationships, which is crucial for distinguishing complex land cover patterns across large spatial areas.

8.2.3.3 3-Dimensional - Fully Convolutional Network (3D-FCN)

The 3D-FCN extends conventional 2D convolutional methods by adding a third dimension to the data input, allowing for the processing of multitemporal SAR datasets. This architecture captures temporal dynamics in the data, making it well-suited for applications where changes over time are significant, such as vegetation monitoring or urban expansion analysis. The 3D convolutions enable the network to extract spatial-temporal features simultaneously, providing a more comprehensive representation of the observed scene and improving classification accuracy for complex land cover types.

For the land cover mapping using SAR data, the methodology from [67] was adopted as a foundation. The referenced work provides a framework, shown in Figure 31, for utilizing SAR time series to classify different land cover types by capturing unique spatial and temporal characteristics. This approach was selected because of its ability to effectively leverage SAR sensitivity of the data to surface structure and moisture, making it suitable for

distinguishing between various land cover classes under diverse environmental conditions. By building upon this methodology, the current study enhances the classification accuracy and scalability for broader geographic areas and more detailed land cover categories.

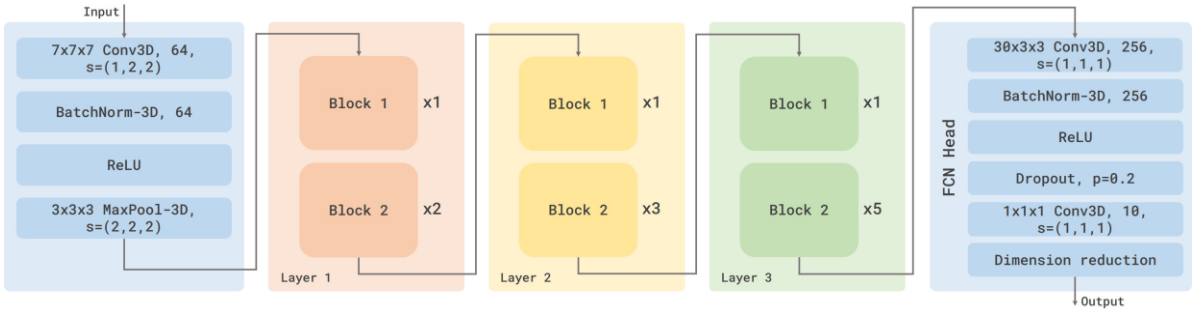


Figure 31. Block diagram of the land cover mapping procedure in [67].

The 3D Fully Convolutional Network (FCN) architecture in [67] consists of several key components and mathematical operations. The network's design includes a 3D ResNet-50 backbone and layers optimized to extract both spatial and temporal features from multitemporal Sentinel-1 SAR data. Please note that the notation and formulation in this section are the same as in the work [67].

As illustrated in Figure 31, the DL architecture is composed of five primary layers. The initial layer applies a $7 \times 7 \times 7$ convolutional kernel with a stride of $s = (1, 2, 2)$ and padding of $p = (3, 3, 3)$ to input data sized $1 \times T \times 256 \times 256$, where T represents the number of temporal images, resulting in the creation of 64 activation maps. Let $\boldsymbol{X}_{in} \in R^{1\times T\times 256\times 256}$ be the input tensor, the output is:

$$\boldsymbol{X}_{\text{conv}} = \text{Conv3D}(\boldsymbol{X}_{\text{in}}, \boldsymbol{W}_{\text{conv}}, s, p)$$

where $\boldsymbol{W}_{\text{conv}}$ represents the convolutional weights.

Following this, a 3D batch normalization step is implemented to standardise the output activations from the preceding convolutional neural network (CNN) layer, aligning them with a unit Gaussian distribution:

$$\boldsymbol{X}_{\boldsymbol{norm}} = \text{BatchNorm}(\boldsymbol{X}_{\boldsymbol{conv}})$$

where the normalization is done over the mean and variance of $\boldsymbol{X}_{\boldsymbol{conv}}$.

The next stage involves applying a *ReLU* activation function, $\boldsymbol{X}_{\boldsymbol{ReLu}} = \max(\boldsymbol{0}, \boldsymbol{X}_{\boldsymbol{norm}})$, succeeded by a $3 \times 3 \times 3$ max-pooling operation to produce a down-sampled feature map, $\boldsymbol{X}_{\boldsymbol{pool}} = \text{MaxPool3D}(\boldsymbol{0}, \boldsymbol{X}_{\boldsymbol{ReLu}})$**.** The resultant tensors are then processed through three layers, each consisting of two residual blocks.

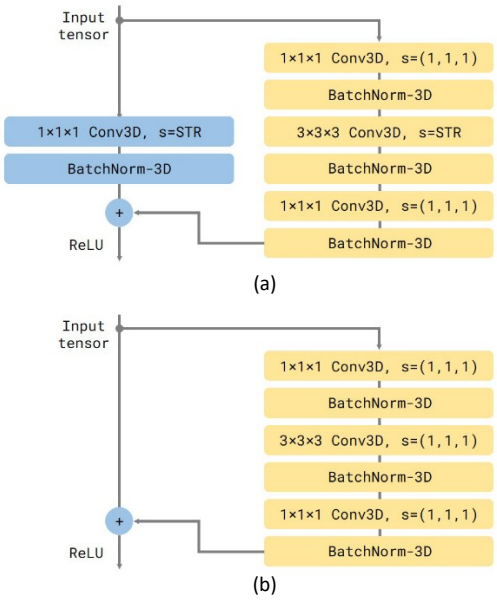


Figure 32. Scheme of the two main residual blocks.

The backbone of the network consists of two key types of residual blocks, in Figure 32, which are essential for enhancing feature extraction and facilitating gradient flow during training.

The first block, depicted in Figure 32(a), performs an addition of the result obtained after the 3D convolution and batch normalization steps with the output of a downsampling operation directly applied to the input data. As shown in Figure 32(a), the convolutions with a stride of $s = \mathrm{STR}$ indicate that the stride varies across layers. In particular, a stride of 1 is used in layer 1, while a stride of 2 (for all axes) is applied in layers 2 and 3. The 3D convolution followed by batch normalization can be formulated as

$$\boldsymbol{X}_{conv\_\mathrm{res}} = Conv3D(\boldsymbol{X}_{\mathrm{in}}, \boldsymbol{W}_{\mathrm{res}}) + \mathrm{BatchNorm}(\boldsymbol{X}_{\mathrm{in}})$$

The second block, in Figure 32(b), differs from the first by omitting the downsampling operation, allowing the residual data from the yellow boxes to be added directly to the input:

$$\boldsymbol{X}_{\mathrm{res}} = \boldsymbol{X}_{conv\_\mathrm{res}} + \boldsymbol{X}_{\mathrm{skip}}$$

Where $\boldsymbol{X}_{\mathrm{skip}}$ is either the original input or down-sampled input.

Once all the backbone layers have been traversed, the data enters the fully convolutional network (FCN) head, which comprises a $30 \times 3 \times 3$ convolutional kernel with a stride of $s = (1, 1, 1)$ and no padding, generating 256 activation maps. This is followed by another 3D batch normalization step $\boldsymbol{X}_{conv\_\mathrm{head}} = Conv3D(\boldsymbol{X}_{res}, \boldsymbol{W}_{\mathrm{head}})$ and a *ReLU* activation function, $\boldsymbol{X}_{ReLu\_\mathrm{head}} = \max(0, BatchNorm(\boldsymbol{X}_{\mathrm{conv\_head}}))$. A dropout layer is then incorporated to help prevent overfitting, with the probability p of a neuron being deactivated set to 0.2.

The final step employs a $1 \times 1 \times 1$ convolutional kernel with a stride of $s = (1, 1, 1)$ to produce a number of output maps M, corresponding to the number of classes.

$$\boldsymbol{X}_{\mathrm{out}} = \mathrm{Conv3D}(\boldsymbol{X}_{\mathrm{ReLu\_head}}, \boldsymbol{W}_{\mathrm{out}})$$

A dimension reduction layer is used to reshape the data into an $M \times 14 \times 14$ data cube, facilitating the final pixel-wise classification.

8.2.4 Posterior normalization

Posterior normalization is the final step in the SAR classification process. This stage integrates the posterior outputs generated from the DL model for land cover classification, as well as outputs for built-up areas and water bodies. The normalization process is critical for ensuring that the combined outputs accurately reflect the probabilities of each class, thus facilitating effective interpretation and decision-making in remote sensing applications.

This normalization process has been successfully implemented with support from the University of Genoa, which provided valuable expertise and resources throughout the development.

Currently, the project focuses on a comprehensive evaluation and analysis of the performance of the UEXT and water extraction algorithms. This assessment examines the effectiveness of the three mentioned deep learning architectures in recognizing built-up and water classes, both seasonal and permanent. By systematically comparing the performance of these algorithms across various scenarios, the research team aims to identify the most effective approaches for accurately distinguishing between built-up areas and water bodies. This research is essential for enhancing applications in remote sensing and urban planning, ultimately contributing to improved environmental monitoring and resource management.

Should the UEXT and water extraction algorithms demonstrate greater effectiveness in recognizing built-up and water classes, respectively, compared to the deep learning-based land cover classifiers, a final task will involve the normalization of the posteriors. The deep learning networks will be trained to recognize the land cover classes listed in Table 1, excluding built-up, seasonal, and permanent water classes, producing relevant posterior probability maps as output.

The posteriors for built-up and water classes will be extracted from the built-up and water maps generated by the UEXT and water extractor outputs, respectively. However, both urban and water recognition approaches do not yield information about the probabilities due to the masking operation applied during the extraction process. To address this, a value known as the *confidence index* will be assigned to each pixel. This confidence index quantifies the reliability of the classification results. The assignment process can be mathematically represented as follows:

- For a pixel identified as built-up:
- $c_{built-up}(i,k) = \begin{cases} 0.8, & \text{if the pixel is classified as built-up} \\ 0.05, & \text{if the pixel is no built-up} \end{cases}$
- For a pixel identified as water (both permanent and seasonal):
- $c_{water}(i,k) = \begin{cases} 0.7, & \text{if the pixel is classified as water} \\ 0.05, & \text{if the pixel is no water} \end{cases}$

In the case of conflicting classifications (i.e., if a pixel is classified as both built-up and water), an average confidence index is assigned:

$$c_{conflict}(i,k) = \frac{c_{built-up}(i,k) + c_{water}(i,k)}{2}$$

For example, if a pixel is classified as both built-up and water, it will receive a confidence index of 42.5%, reflecting the uncertainty in its classification.

The final posterior probabilities are calculated by combining the confidence indices with the outputs from the DL model. Additionally, the normalization of the posterior probabilities takes into account that the values are

typically ranged between [0, 255]. This scaling ensures that the output probabilities are accurately represented.

The normalized posterior for each pixel can be represented as:

$$p_{\text{normalized}}(i,k) = \begin{cases} 254 \times c_{built-up}(i,k), & \text{if classified as built} - \text{up} \\ 254 \times c_{water}(i,k), & \text{if classified as built} - \text{up} \\ 254 \times \dfrac{(p_{DL}(i,k)-1)}{254} \times \left(1 - c_{urban}(i,k) - c_{water}(i,k)\right), & \text{if classified as other land cover} \end{cases}$$

Here, $p_{DL}(i,k)$ represents the posterior probability output from the deep learning model for non-water and non-built-up classes.

Consequently, the normalized posterior image for the built-up areas will assign a value of 80% for those pixels recognized as built-up, and 5% for other pixels. Similarly, for the water class, a pixel recognized as water (either permanent or seasonal) will receive a value of 70%, while non-water pixels will have a value of 5%.

Finally, the normalized output is written to the output posteriors image, where each pixel's value is computed and saved:

$$\boldsymbol{P}_{final} = \left(\frac{254 \cdot \boldsymbol{P}_{normalized}}{100}\right) + 1$$

By implementing these steps, the deep learning probabilities are normalized to account for the confidence indices, ultimately providing a unified SAR classification posterior map. This posterior map enhances the accuracy and reliability of land cover classification, facilitating better decision-making in remote sensing applications.

9 Decision fusion

The Decision Fusion processing chain includes multi-sensor fusion, which combines pixel-wise posteriors of the classification from optical and SAR data, spatial fusion to consider the spatial context, multi-temporal fusion that makes use of the information along the time axis, and spatial harmonization that ensures the spatial smoothness of the mosaicking results of adjacent tiles (granules). Each subsection will start with a brief summary of the previous approach from Phase 1 (as the starting point), followed by the adopted methodology for Phase 2, be it an extension or a new different method.

9.1 Multi-sensor and spatial fusion

The decision fusion processing chain receives pixel-wise posterior probabilities from both the optical and SAR processing chains. Based on the sources of the data, there are three subsets of classes that are taken into account. The first one is the set of common classes, which are the classes that exist in both classification results from optical and SAR data. In order to fuse them, the consensus rule based on logarithmic opinion pool (LOGP) [80], which gives weights differently according to the classes, was used. This is to consider the difference in the sets of classes the optical and SAR sensors can classify confidently, i.e., the optical sensor is generally useful in discriminating all considered land cover classes while urban areas and water bodies are especially distinguishable using the SAR data. The second and third subsets of classes belong to the classes that are exclusively classified using only optical data and SAR data, respectively, which are used as they are, and combined in a unique decision rule as described in [80].

In order to consider the spatial context in terms of the interactions between a class and the classes of the neighboring pixels, a Markov Random Field (MRF) model [81] was applied after the pixel-wise decision fusion step. An MRF is determined by an energy function of which the minimization with respect to the labels is equivalent to the estimation by a maximum a-posteriori criterion [81]. Specifically, in Phase 1, an MRF model with only up to pairwise clique potentials was used. Hence, the energy function can be written as:

$$U(\mathbf{L}|\mathbf{X}) = -\sum_{s \in S} \alpha \log P_F(\ell_s|\boldsymbol{x}_S) - \sum_{\substack{s \in S \\ r \in \partial s}} V(\ell_s, \ell_r),$$

where S is the pixel lattice and s is a shorthand notation for a generic pixel location (i,k), i.e., $\boldsymbol{x}_s$ is short for

$\boldsymbol{x}(i,j)$ with $s=(i,k)\in S$ (and similarly ℓ_s is short for $\ell(i,k)$). Here, the feature vector $\boldsymbol{x}_s$ collects the input optical and SAR data, and the time argument t is dropped because the multi-sensor and spatial fusion stage operate on single-time imagery. α is a positive weight. The MRF considers ℓ_s as sample of the random field $\{\ell_s\}_{s\in S}$ of class labels, whose values are discrete. $\mathbf{L}$ and $\mathbf{X}$ indicate the output label map to be generated and the input image data. $\partial s \subset S$ is the set of neighboring pixels which can be in the form of four connected adjacent pixels (first order connectivity) or the surrounding eight pixels (second order connectivity). The first term of the energy function characterizes the likelihood of the class at the pixel level, which depends on the estimated fused posterior probability $P_F(\ell_s|\boldsymbol{x}_S)$. In the second term, $V(\ell_s,\ell_r)$ controls the label regularization which was defined as:

$$V(\ell_s,\ell_r)=\gamma[1-\delta(\ell_s,\ell_r)],$$

where $\delta(\cdot)$ is the Kronecker delta function, and γ is a weight. This term encourages two neighboring pixels to be labelled with the same class.

In the Phase 2 development, the focus is on extending this spatial model in order to reduce and mitigate the residual artifacts that were observed in the validation of the Phase 1 product. The rationale is twofold: first, these artifacts are generally class-dependent; second, the spatial modeling always needs to be tuned through a compromise between regularization and detail preservation.

In this context, considering that there are some classes needed to be regularized stronger while other classes need to be kept as they are or to be smoothened out less for detail preservation, in Phase 2, the weight is being parameterized according to the pair of class labels, through a function $\gamma(\ell_s,\ell_r)$ whose values are defined empirically. Hence, the first extension of the second term of the energy function now can be written as:

$$V_{sr}(\ell_s,\ell_r)=\gamma(\ell_s,\ell_r)\,[1-\delta(\ell_s,\ell_r)].$$

Furthermore, the extension for Phase 2 also includes the posteriors in the spatial-contextual energy term, i.e., the pairwise potential is also being conditioned on the feature vectors $\boldsymbol{x}_S$. For this, instead of the MRF approach that was adopted in Phase 1, the spatial model of Phase 2 is framed within the more general family of probabilistic graphical models called Conditional Random Fields (CRFs). Switching from MRF to CRF modelling allows significantly extending the flexibility of the local spatial model, while retaining a similar computational burden for the inference process. Specifically, the CRF model of Phase 2 makes use of up to pairwise non-zero potentials and can be defined as:

$$U(\mathbf{L}|\mathbf{X})=-\sum_{s\in S}\alpha\log P_F(\ell_s|\boldsymbol{x}_S)-\sum_{\substack{s\in S\\ r\in\partial s}}V_{sr}(\ell_s,\ell_r|\mathbf{X}),$$

where the pairwise potential also incorporates input image data. An effective choice for this potential is generally:

$$V_{sr}(\ell_s,\ell_r|\mathbf{X})=\gamma(\ell_s,\ell_r)[1-\delta(\ell_s,\ell_r)]K(\boldsymbol{x}_s,\boldsymbol{x}_r),$$

where $K(\boldsymbol{x}_s,\boldsymbol{x}_r)$ is a kernel function that expresses a similarity measure associated with the feature vectors of neighboring pixels. The typical choice is the contrast-sensitive CRF Potts' model, which corresponds to this Gaussian choice of the kernel (radial basis function):

$$\mathcal{K}(\boldsymbol{x}_s,\boldsymbol{x}_r)=e^{-\varphi\|\boldsymbol{x}_s-\boldsymbol{x}_r\|^2},$$

where φ is a positive parameter. On one hand, the contrast-sensitive Potts is known to be effective at locally tuning spatial regularization as a function of the input imagery. On the other hand, it makes use of the feature vectors extracted from the input imagery directly.

In order to be aligned with the CCI+ HRLC pipeline, in which the Decision Fusion processing chain only receives the posterior probabilities from the optical and SAR processing chain and not the optical and SAR images directly, the adopted CRF model is specifically formulated in terms of posteriors instead of feature vectors. The general model of the adopted pairwise potential can be written as:

$$V_{sr}(\ell_s,\ell_r|\mathbf{X})=\gamma(\ell_s,\ell_r)[1-\delta(\ell_s,\ell_r)]\mathcal{K}\big(\boldsymbol{P}_s(\mathbf{X}),\boldsymbol{P}_r(\mathbf{X})\big),$$

where $\boldsymbol{P}_s(\mathbf{X})$ is the vector collecting the fused posterior probabilities $P_F(\ell_s=\omega_k|\boldsymbol{x}_S)$, of all classes $\omega_k, k=$

$1,2,..,\mathcal{C}$, where $\mathcal{C}$ is the number of classes. Accordingly, the spatial model developed in Phase 2 can be summarized as follows:

$$U(\mathbf{L}|\mathbf{X}) = -\sum_{s\in S} \alpha \log P_F(\ell_s|\boldsymbol{x}_S) - \sum_{\substack{s\in S \\ r\in \partial s}} \gamma(\ell_s,\ell_r)[1-\delta(\ell_s,\ell_r)]\mathcal{K}\big(\boldsymbol{P}_s(\mathbf{X}),\boldsymbol{P}_r(\mathbf{X})\big),$$

We notice that, unlike the MRF model used in Phase 1, which was spatially stationary, this CRF model is spatially nonstationary, with the goal of aligning more adaptively with the spatial details in the scene.

We also mention that, with the possibility of using deep learning in Phase 2, there is the opportunity of using a deep model to address the pixel-wise posterior fusion while considering spatial information within one unique neural model. For this purpose, a Convolution Neural Network (CNN) can be used to fuse the posterior outputs coming from optical and SAR processing chains, directly as if they were the outputs of a softmax layer. However, while it comes with the benefit of having a possibly very flexible model to do two separate tasks at once, this approach also has a drawback, as the CNN needs training whereas the current probabilistic graphical method (CRF) does not need one. This implies a large modification in the CCI+ HRLC pipeline because the training set should be fed to the Decision Fusion processing chain as well. More generally, the adoption of a deep learning strategy within the CCI+ HRLC pipeline generally implies a strong reformulation of the whole pipeline itself – not only of the Decision Fusion components. In the framework of this overall cost/benefit tradeoff, this possibility to use deep learning formulation will be taken into consideration in view of the second production.

9.2 Multi-temporal fusion

As land cover mapping is addressed in multiple years, doing the classification and fusion in each time step independently from one another inevitably produces noisy results along the time axis. This is especially relevant in the case of historical maps because of the variability in the availability of the data, which causes temporal inconsistency in the classification. In order to prevent this drawback, multi-temporal fusion has been introduced to the Decision Fusion processing chain, allowing the information from other time steps to be taken into account in one particular time step. In Phase 1, a simple cascade model [82] was used to perform the multi-temporal fusion by using the static map of 2019 as a reference and by propagating the information to the other historical products backward. On one hand, this choice allowed greatly reducing false land-cover transitions, as compared to separate independent classifications at the various times. On the other hand, the cascade multi-temporal fusion model is limited in its use of the temporal information because it only considers one pair of time steps, while there exists other useful information in the other time steps of the time series. In Phase 2, to favor the temporal consistency, we use the information of the whole time series of the posteriors computed for all the years in which land cover is being mapped. For this purpose, the adopted approach is based on the theory of Hidden Markov Models (HMMs).
Given the set of fused posterior probabilities coming from the classification of optical data and SAR data at every time step in the considered time series, let us consider, on each pixel, the joint distribution $P_F(\boldsymbol{\ell}|\boldsymbol{x}) = P_F(\ell_1,\ell_2,\dots,\ell_T|\boldsymbol{x}_1,\boldsymbol{x}_2,\dots,\boldsymbol{x}_T)$ of the vector of all labels $\boldsymbol{\ell} = (\ell_1,\ell_2,\dots,\ell_T)$, given the vector of all feature vectors $\boldsymbol{x} = (\boldsymbol{x}_1,\boldsymbol{x}_2,\dots,\boldsymbol{x}_T)$. Here, we focus on a single individual pixel, dropping for the sake of clarity the explicit indication of its location (i,k). A first-order HMM is defined by these two conditions:

$$P(\ell_t|\ell_{t-1},\ell_{t-2},\dots,\ell_1) = P(\ell_t|\ell_{t-1}) \qquad \forall t \in \{2,3,\dots,T\}$$
$$P(\boldsymbol{x}|\boldsymbol{\ell}) = \prod_t P(\boldsymbol{x}_t|\boldsymbol{\ell}_t)$$

which formalize a Markovianity condition along time on the labels and a conditional-independence assumption on the relationship between observations and labels, respectively. Under this model, we can prove that the global posterior can be written as:

$$P(\boldsymbol{\ell}|\boldsymbol{x}) \propto \prod_t \frac{P_F(\ell_t|\boldsymbol{x}_t)}{P(\ell_t)} P(\ell_t|\ell_{t-1})P(\ell_1),$$

where $P(\ell_t)$ and $P(\ell_1)$ are priors and $P(\ell_t|\ell_{t-1})$ is the transition probability stating the probability of one class changing to another class across time. This formulation supports a maximum a-posteriori (MAP) inference directly. Similarly, the marginal posterior mode (MPM) inference of $P(\ell_t|\boldsymbol{x})$ can be done sequentially using the

forward-backward algorithm [83]. This formulation involves defining a forward procedure that evaluates the joint probability of observing all feature vectors up to time t and the label at time t:

$$\alpha(\ell_t) \equiv P(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \ell_t),$$

and a backward step that determines the conditional probability of all observations from time $t+1$ up to T given the value of ℓ_t:

$$\beta(\ell_t) \equiv P(\boldsymbol{x}_{t+1}, \ldots, \boldsymbol{x}_T | \ell_t).$$

$P(\ell_t|\boldsymbol{x})$, then, can be computed by:

$$P(\ell_t|\boldsymbol{x}) = \frac{\alpha(\ell_t)\beta(\ell_t)}{\sum_{\ell_t} \alpha(\ell_t)\beta(\ell_t)}.$$

The rationale is that inferring this in sequential order makes the algorithm computationally efficient while making use of the information from the full time series. This is the methodological formulation adopted in Phase 2.
In principle, the HMM theory would also allow to estimate automatically the matrix of the transition probability values through a case-specific formulation of the expectation-maximization (EM) algorithm. However, this option is not being considered, at least for the first production, within the CCI+ HRLC processing chain, in which application-guided parameter setting is preferable in order to align the product with the desiderata of the climate community.
As in the case of spatial multi-sensor fusion, multitemporal fusion can be addressed in terms of deep learning as well, with special focus on the family of Recurrent Neural Networks (RNNs) [84]. Several types of RNN such as the Long Short-term Memory (LSTM) and the Gated Recurrent Unit (GRU) networks [85], [86], are known from the literature to favor a good performance in terms of accuracy. However, as already mentioned in the previous section, this deep learning alternative comes with requirements of training data and generally higher cost in terms of processing time and computational load. Analogously, within this cost/benefit balance, the opportunity to use deep learning will be taken into account for the second production.

9.3 Spatial harmonization

The spatial harmonization module is in charge of favoring the spatial consistency across the boundary between adjacent mapping tiles during the process of mosaicking them together to generate the final land cover map. The main challenge of this is the presence of residual edge artifacts due, in the Phase-1 product, to the different properties of the data on the two tiles, which, in turn, is generally caused by the different data availabilities. In Phase 1, a linear opinion pool approach was applied across the overlapping areas of two neighboring granules.
The extension in Phase 2 incorporates class information into the spatial gradient from one granule to the other to favor a more seamless spatial fusion. This means that, during the spatial harmonization, the gradient across the two granules in the overlapping area is parameterized as a function of not only the spatial location but also the class labels. Operatively, this gradient is determined by space-varying weights, whose values are defined now by biasing on the labels.
The rationale is that the aforementioned residual artifacts were noticed to be class-dependent. Therefore, the class memberships estimated by the multi-sensor, spatial, and temporal fusion modules are used to partially guide the harmonization in a class-oriented (hence application-specific) manner.

10 Multitemporal change detection and trend analysis

The use of the High Resolution (HR) Satellite Image Time Series (SITS) in the Change Detection (CD) context defines challenges for processing the great amount of data and developing advanced CD methods for handling the optical data from 1990-2024 in the time dimension. In particular, challenges will be addressed for dealing with long SITS where $T \gg 2$ images. For the long time scale case (i.e., several years) with high temporal resolution, CD can be defined in several ways. Among them, we consider the possibility to analyze SITS to detect the changes that have happened between consecutive years. Methods developed for the CCI Medium Resolution Land Cover (MRLC) accounting for detecting change points of abrupt LC changes at annual scale. They mostly rely on medium resolution (300m to 500m) SITS and mainly compare vegetation indices [87]. However, the usage of those strategies does not fit the case of multiple class trends. There is a need to define new methods to analyze dense HR SITS using a multi-feature framework and to detect differences between the consecutive years. In this

context, several factors have been considered in the development of a method: i) it should extract relevant information from the multi-annual SITS to properly model the behavior of various LCs, ii) it should take into account the irregularity of SITS that is caused mostly by atmospheric conditions, iii) it should exploit a strategy that can effectively calculate the differences of feature time series on a yearly basis and, iv) it needs an effective and automatic change detector to locate the changes in space and time.

LC Changes can be divided into three classes [88]: (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 Normalized Difference Vegetation Index (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 be developed with an emphasis on quantification of abrupt/permanent changes since climate change tends to be more abrupt than gradual [89]. The analysis is performed over the products derived from the multitemporal optical merging step, plus the HRLC static and five years regional maps. The general block scheme shown in Figure 33 is used for the generation of HRLC change products. This chapter is organized around two main focuses of CCI HRLC phase 2:

- The first focus involves reprocessing the products from phase one to refine the produced maps.
- The second focus extends production to new areas and clusters of Sentinel-2 tiles, representing each area of interest using a multi-annual, multi-feature land cover change detection approach.

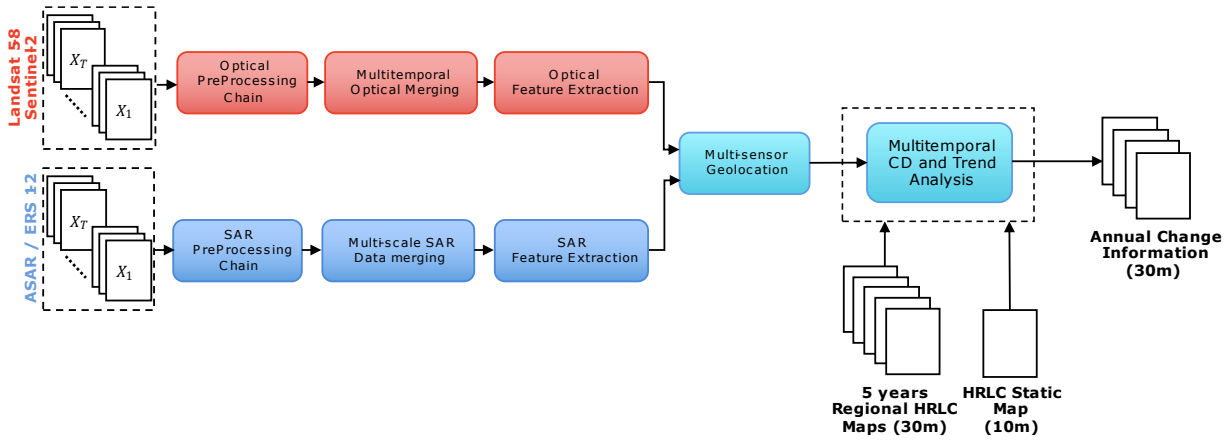


Figure 33. Block-based representation of the processing chain for the multitemporal change detection and trend analysis.

10.1 Reprocessing Phase 1 Historical LC Change Detection and Trend Analysis

For the historical analysis, the products generated in this phase of the project will undergo refinement through the integration of ancillary data and multiple inputs derived from the Phase 1 production, as shown in the Figure 34. This workflow ensures consistency and homogeneity throughout the processing chain, which is critical for accurate detection and mapping of historical changes.

The historical multi-annual CD starts with registered optical images from Landsat 5-8 that are processed in a sequence that involves the identification of changes and the year of interest. The outputs from Phase 1, which include LC classifications maps and auxiliary datasets, are used to improve this analysis. Incorporating these elements guarantees that the historical maps generated are more reliable and consistent across different periods. This approach will also support better tracking of long-term changes by leveraging the temporal depth and spatial consistency provided by Phase 1 data, leading to more accurate historical change detection results.

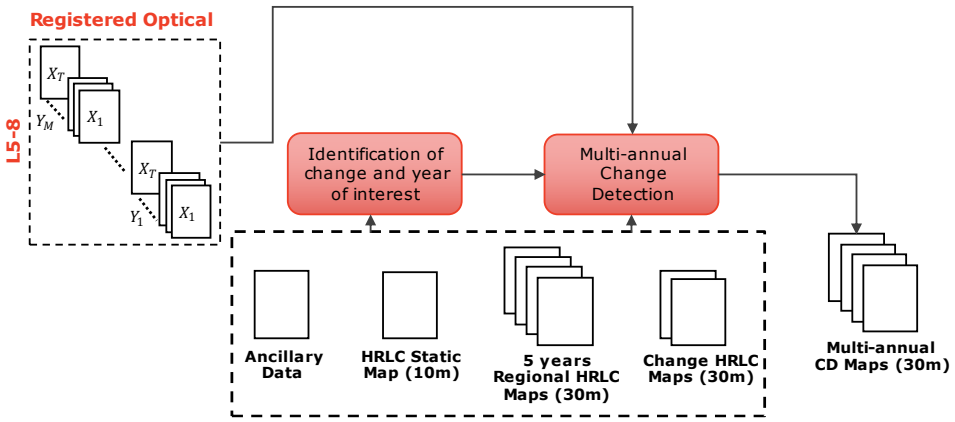


Figure 34. Historical multi-annual change detection and trend analysis

10.2 Multi-annual Multi-feature Change Detection

In phase2, new areas are being introduced for analysis and change detection utilize also Sentinel-2 datasets. Here, we should take into account that the CD processing chain works in pixel level in a yearly basis from 1990 to 2024 considering three subcontinental area in Amazonia, Africa and Siberia [AD3], so the methodologies will be updated considering the higher spatial resolution of Sentinel-2 data, the need to harmonize multi-sensor data across these regions, and the inclusion of advanced techniques for managing temporal and spatial variability in land cover dynamics.

The goal is to apply the same preprocessing stages developed in optical preprocessing, with greater emphasis on creating an integrated and unified preprocessing workflow for both land cover classification and land cover change detection. This will streamline the optical data analysis process, reduce the number of processing steps, and ensure a homogeneous and robust preprocessing pipeline.

Let $SITS = \{Y_1, Y_2, \dots Y_m, \dots, Y_M\}$ be a pre-processed satellite image time series that includes M years of images acquired over the same geographical area. $Y_m = \{X_1, X_2, \dots, X_n, \dots, X_T\}$ is a year in the SITS composed of several satellite images. Let us assume that Y_m has non-uniform time sampling and each image has a total number of P pixels. Given an image $X_t \in SITS$, each pixel value represents the surface reflectance in a given spatial position $p \in [1, P]$ and a temporal instant $t \in [1, T]$. Let $\{b_1, b_2, \dots, b_B\}$ be the set of bands that compose the images and K the total number of bands. In phase 2, instead of using the original time series data, we utilize composite data that can better model LC behavior through time generated in optical preprocessing module (Figure 1). This enhances the ability of the breakpoint detector methodology to identify change dates and probabilities more effectively with less computations. In this stage, considering the LC maps produced every five years, there are years when LC classification is not conducted; in such cases, the composite generation will be implemented for LC change detection.

The input of the processing chain is the pre-processed composites that is employed in the feature selection module to distinguish the spectral trends of different sets of LC changes (in details Figure 35).It is possible to generate the Post Classification Comparison (PCC) maps using LC classification map every five years. These maps have been produced in order to: 1) align the changes that have occurred during five years derived from the LC maps to the changes detected in multitemporal change detection processing chain, and 2) reduce the computational burden. The PCC maps effectively filter out unchanged pixels over the five-year period by comparing LC maps from two different years on a pixel-by-pixel basis. Only the pixels where a change in LC class is detected are marked for further processing, allowing the less computational effort in the multitemporal change detection chain. Cloud/shadow mask is imposed to remove cloudy pixels.

After feature selection, for different sensors or the years with high or less frequent acquisitions the time series reconstruction module will be considered or not:

- For Sentinel-2 years with frequent weekly acquisitions and adequate data, composites are generated to represent the LC behavior without the need for full time series reconstruction. This approach saves processing time and computational resources.

- For Landsat years with less frequent acquisitions, the strategy will shift toward using monthly, bi-monthly, or seasonal composites generated during the pre-processing phase (as outlined in the optical pre-processing block in Figure 1). These composites, instead of the original time series data, are used to properly model LC behavior over time.
- In the case of critical Landsat or Sentinel-2 years where there are data gaps, we reconstruct the time series to ensure a reliable representation of the LC dynamics.

The time series reconstruction generates a continuous and dense feature time series by using a LC map to select a suitable model for different LCs. This ensures reliable CD analysis despite limited data availability. The abrupt change detection is performed considering the proposed Multi-Feature Hyper-temporal Change Vector Analysis (MHCVA) [90] technique that analyzes the differences between the features extracted from two consecutive yearly feature spaces. In phase two, instead of using the original time series data, we utilize composite data that can better model LC behavior through time generated in optical preprocessing module (see Figure 1). This enhances the ability of the breakpoint detector methodology to identify change dates and probabilities more effectively.

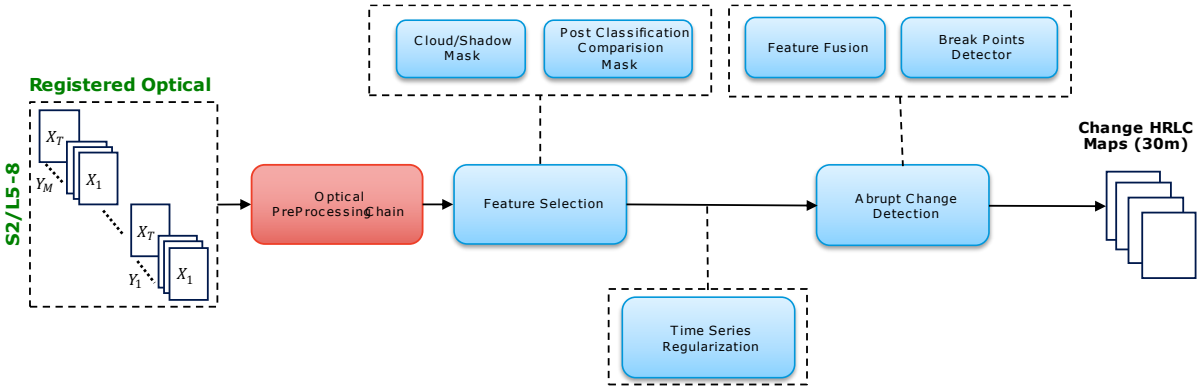


Figure 35. Multi-annual Multi-feature LC change detection.

10.2.1 Feature Selection

The feature space design will focus on developing region-specific feature spaces customized to the unique characteristics of each area by using methods to optimize change detection analysis [91], [92]. For example, the feature space incorporates spatial, temporal, and fine-grained features that capture the detailed information of the LCs and their changes over time in each region. Pretrained deep learning models are evaluated for extracting complex spatial and temporal features, leveraging their capacity to learn intricate patterns from large datasets [93].

The first stage is to determine the suitable features which are the most important factor in distinguishing the spectral trends of various sets of LC changes. Different couples of the available sensor bands are considered to compute a set of Normalized Difference Indices ($NDI_f^{SITS}, f = 1, \dots, F$) of different bands as follows:

$$NDI_f^{SITS} = \frac{b_u - b_v}{b_u + b_v}, f = 1, \dots, F$$

This stage transforms the K-dimensional feature space into a F-dimensional feature space, where b_u and b_v belong to B (the set of bands available in a sensor) and u and $v \in [1,2, \dots, B]$. These different ratios between different spectral bands (e.g., SWIR/NIR, Red/Green) can highlight different changes in land cover types like forests, water, or urban areas, as follows:

- Normalized Difference Vegetation Index (NDVI) is widely used to assess vegetation health and density by leveraging the high reflectance of healthy vegetation in the Near-Infrared (NIR) band and the low reflectance in the Red band. It helps identify changes in vegetation cover over time, such as deforestation, agricultural stress, or regrowth following disturbances [94].
- Normalized Difference Water Index (NDWI) is designed to detect and monitor changes in water bodies

by capitalizing on the high absorption of NIR radiation by water and its high reflectance in the Green band. This index helps in mapping lakes, rivers, wetlands, and coastal zones and can track seasonal fluctuations in water bodies or detect long-term water changes due to drought or urbanization [95].

- Normalized Burn Ratio (NBR) is used for identifying burned areas and assessing wildfire severity by analyzing the changes in NIR and Shortwave Infrared (SWIR) reflectance. Burned areas exhibit lower NIR and higher SWIR reflectance compared to healthy vegetation, making this index effective in mapping fire-affected regions [96]. An enhanced version of NBR is developed that takes into account the reflectance of water, which is called NBR+ [97].
- Soil-Adjusted Vegetation Index (SAVI) modifies the NDVI formula by incorporating a correction factor (L) to minimize the influence of soil brightness, making it more suitable for arid regions or areas with sparse vegetation. The factor L is usually set to 0.5 but can be adjusted depending on soil conditions. SAVI provides more accurate vegetation assessments in areas with exposed soil, such as deserts or grasslands, where traditional NDVI might not perform well [98].
- Normalized Difference Built-up Index (NDBI) is designed to identify urban and built-up areas by analyzing the difference in reflectance between the SWIR and NIR bands. Urban areas typically reflect more SWIR radiation and less NIR radiation compared to vegetation, making NDBI effective for detecting urban expansion and monitoring land use changes [99].
- The Non-Homogeneous Feature Difference (NHFD) is an index designed to detect differences between various spectral features across multiple bands or between two images taken at different times. It works by calculating the absolute differences between corresponding spectral bands (or other relevant features) of two images, summing these differences across all the available bands to produce a single value that represents the degree of change [100].

Each of the provided indices offers valuable information about land cover, aiding in the analysis of changes. However, depending on the specific area of study, certain indices may be more effective than others. Therefore, a thorough analysis of the region is conducted to determine the most relevant indices or a combination of them, ensuring a more reliable change detection process. In this context, temporal features can also be highly useful for capturing how spectral information evolves over time, allowing for the identification of abrupt changes in land cover. Time series analysis of indices like NDVI, NDWI, and NBR provides insights into patterns such as deforestation, regrowth, and seasonal fluctuations in vegetation or water bodies. Phenological metrics, including the start and end of growing seasons or peak biomass, help monitor vegetation dynamics, while seasonal variability analysis distinguishes natural cycles from human disturbances. Multi-year averages of indices offer a stable view of persistent changes over years, reducing noise and revealing long-term trends like continuous deforestation or urban growth.

Methods like Principal Component Analysis (PCA), Scale-Invariant Feature Transform (SIFT), and Local Directional Pattern (LDP) are highly beneficial in different change detection scenarios due to their ability to extract detailed spatial features and patterns from imagery. For instance, PCA is widely used for dimensionality reduction and can highlight the most significant variance in data, making it useful for detecting subtle changes in land cover over time. SIFT is effective for identifying key points and matching features between images, which is particularly valuable for tracking structural or morphological changes in urban areas. LDP captures local texture variations, making it useful for detecting fine-scale changes in land surfaces, such as soil degradation or vegetation shifts [101].

Convolutional Neural Networks (CNNs) [102], for instance, can automatically learn hierarchical features from raw image data, capturing both low-level details like edges and textures as well as high-level abstractions such as shapes and objects. This is particularly useful for complex change detection scenarios, such as urban expansion or deforestation, where both fine-grained and large-scale changes need to be monitored.

Autoencoders and pre-trained deep networks (e.g., ResNet or VGG) are also commonly used for feature extraction, especially when large labeled datasets are unavailable. These networks can be fine-tuned for specific tasks, allowing the extraction of rich, spatial, and temporal features that are crucial for identifying land cover changes. Deep learning techniques have the added advantage of being able to handle diverse input types (e.g.,

optical and radar data) and can adapt to different regions and landscape characteristics, making them more versatile for multi-temporal change detection.

The overall approach for optical feature extraction, as illustrated in Figure 36, begins with the preprocessing of satellite imagery (see Figure 1). The feature extraction involves generating various feature types: normalized difference features, temporal features, and deep network features. Normalized difference features allow for the assessment of vegetation health or land cover changes by calculating different normalized indices. Temporal features leverage the time series aspect of the data, capturing seasonal dynamics and trends over time. Deep network features utilize advanced machine learning techniques to extract complex patterns from the data. After extracting these features, the next steps involve feature selection to identify the most relevant variables, followed by accuracy assessment to evaluate the performance of the feature extraction process. Ultimately, this comprehensive approach culminates in change detection, enabling the identification and analysis of changes in land cover or environmental conditions over the monitored period.

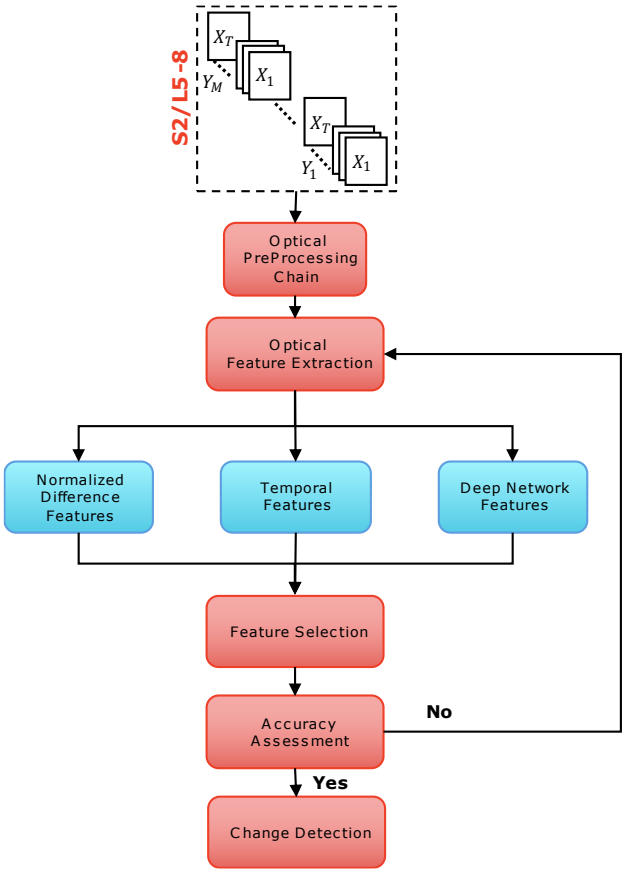


Figure 36. The best feature space selection flowchart.

10.2.2 Time Series Reconstruction

Time series reconstruction will be implemented for the years (sensors) characterized by non-equally distributed temporal sampling and non-continuous trend, also affected by noisy oscillation not corrected in the pre-processing step. In this context, continuous refers to time series data that contains samples at regularly spaced time intervals without significant gaps or missing values.

The proposed time series reconstruction considers two different strategies for the vegetation and non-vegetation samples. To produce reliable and continuous time series for the non-vegetation profiles the strategy is based on two steps: i) for each pixel in the image extract the NDI-SITS, ii) perform NDI data-SITS augmentation by upper envelope and dropout strategy (a piecewise cubic interpolation is used here) [103].

Further details on the augmentation by upper envelope strategy are illustrated as follows:

- Define a NDI-SITS set (NDI_{tr}), corresponding to a year (365 days), plus the two previous and two later months of data;

- For each NDI_{tr}, select the samples that are above a given threshold (defined by trial and error as NDI = 0.4). This threshold identifies when a given $SITS_p$ experiences a significant variability over time;
- Calculate the local maxima (as the points with zero first derivative and negative second derivative) of the selected samples and withdraw the remaining ones (from NDI_{tr}). This leads to the upper envelope of the data;
- Use the samples below the threshold and the local maxima from previous step for data imputation by means of a Piecewise Cubic Hermite Interpolating Polynomial (PCHIP). The selection of PCHIP over other interpolation methods is justified by its characteristic to preserve the shape of the data and respect monotonicity. The combination of these samples is defined as the upper-envelope set;
- Subtract the imputed data from NDI_{tr}. Reinsert the withdrawn samples with a difference greater than zero to the upper-envelope set. This step allows to better follow the shape of the original data;
- Impute the updated upper-envelope set by means of PCHIP;
- Remove the two previous and two later months from NDI_{tr}.
- The definition of NDI_{tr} allows to better model the beginning and the end of the SITS, thus smoothing discontinuities and possible errors in LCCD analysis.

In the case of complex land cover classes like vegetation type (i.e., grass, shrubs, forest and crops) that show strong variabilities over space and time due to intrinsic seasonality and the large amount of species around the world, a third step is added that performs adaptive non-parametric regression of NDI-SITS by considering a General Regression Neural Network (GRNN) by taking inspiration from [104] [105](see Figure 37). The non-parametric regression is used and adapted to produce continuous and regularly sampled temporal signatures for vegetation pixels. 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 produce an increased number of features. The NDI temporal signatures are then interpolated, considering 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 smoother behaviour. Last, a non-parametric regression captures the temporal signatures trend reducing the profile complexity and arithmetic dependency.

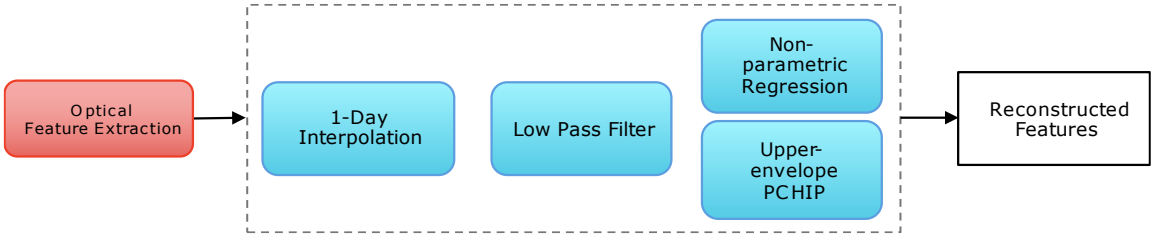


Figure 37. Time series reconstruction step.

10.2.3 Abrupt Change Detection

The proposed CD method for multi-annual SITS detects and characterizes changes occurring between consecutive years. Abrupt change detection incorporate a feature fusion strategy based on the MHCVA [90]. Assuming NDI_f^{SITS} being the feature time series for each feature f and $f = 1, \dots, FR$, MHCVA finds the differences between every couple of consecutive years in the NDI_f^{SITS} by applying a difference feature magnitude calculation. First, the regular NDI_f^{SITS} is divided into the records acquired in the same years $NDI_f^{SITS} = \{NDI_f^{Y_1}, \dots, NDI_f^{Y_m} \dots, NDI_f^{Y_M}\}$. Each $NDI_f^{Y_m}, f = 1, \dots, FR$ is a regularly sampled feature time series. Then, a MHCV is computed between the consecutive years in NDI_f^{SITS} to highlight the temporal differences in the time series. If for a given period data are not available for two neighboring years at the pixel level, the algorithm considers the next year to calculate the MHCV.

Let $NDI_f^{Y_m}$ and $NDI_f^{Y_{m+1}}$, $f = 1, \dots, FR$ be the sets of smooth NDIs of feature f for the years Y_m and Y_{m+1} within a SITS. Both sets have the same length according to the time series reconstruction technique (for a daily and weekly reconstruction it is equal to 365 and 52, respectively). A MHCV between Y_m and Y_{m+1} is performed by subtracting $NDI_f^{Y_m}$ and $NDI_f^{Y_{m+1}}$ for each $f = 1, \dots, FR$ as $NDI^{Y_m,Y_{m+1}} = \{NDI_1^{Y_m,Y_{m+1}}, \dots, NDI_f^{Y_m,Y_{m+1}}\}$. Then, the magnitude of $NDI^{Y_m,Y_{m+1}}$ is calculated as follows:

$$|NDI^{Y_m,Y_{m+1}}| = \sqrt{\sum_{f=1}^{f=FR} {NDI_f^{Y_m,Y_{m+1}}}^2}$$

The resulting magnitude of MHCV represents the variability between the pairs of neighboring years and fuses the selected features to benefit from all the features information. Moreover, it makes the processing time computationally efficient for dealing with a large multi-annual multi-feature dataset. The output is a time series $NDI^{SITS} = \{NDI^{Y_1,Y_2}, \dots, NDI^{Y_m,Y_{m+1}}, \dots, NDI^{Y_{M-1},Y_M}\}$ and it is used as input for the break point detector that detects multiple abrupt changes in the trend component of the NDI^{SITS}.

In phase2, other break point detectors such as Bayesian Estimator of Abrupt change, Seasonality & Trend (BEAST) [106] will be considered. BEAST employs a Bayesian framework to detect abrupt changes in the trend, seasonality, and noise components of a time series, providing probabilistic estimates of breakpoints and quantifying uncertainty. It is highly flexible, capable of modeling complex changes such as gradual shifts, sudden jumps, or variations in seasonality, making it well-suited for noisy, irregular time series where uncertainty quantification is important. However, this flexibility comes at the cost of increased computational demand due to its reliance on Bayesian inference techniques like Markov Chain Monte Carlo (MCMC). While BEAST is ideal for complex, noisy datasets requiring detailed uncertainty assessment, BFAST excels in applications with well-defined, periodic seasonal patterns and is widely used in land cover change detection where seasonal cycles are predictable. Since BEAST is implemented in R, a Python implementation is being considered to evaluate the computational extensiveness of the method; by conducting the following experiments using this breakpoint detector, it will be possible to assess the reliability of the analysis. Another version of the BEAST is the Bayesian Online Change Point Detection (BOCPD) method [107], known for its speed and efficiency, will be considered as alternatives to BFAST and BEAST.

11 References

[1] D. Frantz, "FORCE—Landsat + Sentinel-2 Analysis Ready Data and Beyond," *Remote Sensing*, vol. 11, no. 9, Art. no. 9, Jan. 2019, doi: 10.3390/rs11091124.

[2] 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.

[3] R. Richter and D. Schläpfer, "Atmospheric/Topographic Correction for Satellite Imagery (ATCOR-2/3 User Guide, Version 9.5, August 2023)," Wessling, Germany, DLR-IB 564-01/2023, 2023.

[4] B. Mayer and A. Kylling, "Technical note: The libRadtran software package for radiative transfer calculations - description and examples of use," *Atmospheric Chemistry and Physics*, vol. 5, no. 7, pp. 1855–1877, Jul. 2005, doi: 10.5194/acp-5-1855-2005.

[5] 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.

[6] 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.

[7] 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.

[8] S. Qiu, Z. Zhu, and B. He, "Fmask 4.0: Improved cloud and cloud shadow detection in Landsats 4–8 and Sentinel-2 imagery," *Remote Sensing of Environment*, vol. 231, p. 111205, Sep. 2019, doi: 10.1016/j.rse.2019.05.024.

[9] D. Frantz, E. Haß, A. Uhl, J. Stoffels, and J. Hill, "Improvement of the Fmask algorithm for Sentinel-2 images: Separating clouds from bright surfaces based on parallax effects," *Remote Sensing of Environment*, vol. 215, pp. 471–481, Sep. 2018, doi: 10.1016/j.rse.2018.04.046.

[10] L. Baetens, C. Desjardins, and O. Hagolle, "Validation of Copernicus Sentinel-2 Cloud Masks Obtained from MAJA, Sen2Cor, and FMask Processors Using Reference Cloud Masks Generated with a Supervised Active Learning Procedure," *Remote Sensing*, vol. 11, no. 4, Art. no. 4, Jan. 2019, doi: 10.3390/rs11040433.

[11] G. Mateo-García, L. Gómez-Chova, J. Amorós-López, J. Muñoz-Marí, and G. Camps-Valls, "Multitemporal Cloud Masking in the Google Earth Engine," *Remote Sensing*, vol. 10, no. 7, Art. no. 7, Jul. 2018, doi: 10.3390/rs10071079.

[12] H. Zhai, H. Zhang, L. Zhang, and P. Li, "Cloud/shadow detection based on spectral indices for multi/hyperspectral optical remote sensing imagery," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 144, pp. 235–253, Oct. 2018, doi: 10.1016/j.isprsjprs.2018.07.006.

[13] C. Aybar *et al.*, "CloudSEN12, a global dataset for semantic understanding of cloud and cloud shadow in Sentinel-2," *Sci Data*, vol. 9, no. 1, p. 782, Dec. 2022, doi: 10.1038/s41597-022-01878-2.

[14] D. P. Roy *et al.*, "A general method to normalize Landsat reflectance data to nadir BRDF adjusted reflectance," *Remote Sensing of Environment*, vol. 176, pp. 255–271, Apr. 2016, doi: 10.1016/j.rse.2016.01.023.

[15] D. P. Roy, Z. Li, and H. K. Zhang, "Adjustment of Sentinel-2 Multi-Spectral Instrument (MSI) Red-Edge Band Reflectance to Nadir BRDF Adjusted Reflectance (NBAR) and Quantification of Red-Edge Band BRDF Effects," *Remote Sensing*, vol. 9, no. 12, Art. no. 12, Dec. 2017, doi: 10.3390/rs9121325.

[16] Y. Li, Q. Liu, S. Chen, and X. Zhang, "An Improved Gap-Filling Method for Reconstructing Dense Time-Series Images from LANDSAT 7 SLC-Off Data," *Remote Sensing*, vol. 16, no. 12, Art. no. 12, Jan. 2024, doi: 10.3390/rs16122064.

[17] N. Flood, "Seasonal Composite Landsat TM/ETM+ Images Using the Medoid (a Multi-Dimensional Median)," *Remote Sensing*, vol. 5, no. 12, Art. no. 12, Dec. 2013, doi: 10.3390/rs5126481.

[18] F. Trastour, L. Duan, and M. Swaine, "Sentinel-2 Global Mosaic: Algorithm Theoretical Basis Document," S2GM-ATBD-001-ACR, Mar. 2023. [Online]. Available: https://s2gm.land.copernicus.eu/help/documentation

[19] Barstow, "Format Specification for ERS Products within ENVISAT Format".

[20] R. Barstow, "ENVISAT-1 Products Specifications Volume 8: ASAR Products Specifications," vol. 8, p. 179.

[21] John C. Curlander and Robert N. McDonough, *Synthetic aperture radar*, vol. 11. New York: Wiley, 1991.

[22] D. Small, "Flattening Gamma: Radiometric Terrain Correction for SAR Imagery," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 8, pp. 3081–3093, Aug. 2011, doi: 10.1109/TGRS.2011.2120616.

[23] J.-W. Park, A. Korosov, M. Babiker, S. Sandven, and J.-S. Won, "Efficient Thermal Noise Removal for Sentinel-1 TOPSAR Cross-Polarization Channel," *IEEE Transactions on Geoscience and Remote Sensing*, vol. PP, pp. 1–11, Dec. 2017, doi: 10.1109/TGRS.2017.2765248.

[24] Senbox, "Developer Guide - SNAP - SNAP Wiki." [Online]. Available: https://senbox.atlassian.net/wiki/spaces/SNAP/pages/8847381/Developer+Guide

[25] "Radiometric Calibration of Level-1 Products - Sentinel-1 SAR Technical Guide - Sentinel Online," Sentinel Online. [Online]. Available: https://copernicus.eu/radiometric-calibration-of-level-1-products

[26] F. Filipponi, "Sentinel-1 GRD Preprocessing Workflow," presented at the Proceedings, Jun. 2019, p. 6201. doi: 10.3390/ECRS-3-06201.

[27] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou, "A tutorial on synthetic aperture radar," *IEEE Geoscience and Remote Sensing Magazine*, vol. 1, no. 1, pp. 6–43, Mar. 2013, doi: 10.1109/MGRS.2013.2248301.

[28] C. Oliver and S. Quegan, *Understanding synthetic aperture radar images*. in The SciTech radar und defense series. Raleigh, NC: SciTech Publishing, Inc, 2004.

[29] 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.

[30] O. Rubel, V. Lukin, A. Rubel, and K. Egiazarian, "Selection of Lee Filter Window Size Based on Despeckling Efficiency Prediction for Sentinel SAR Images," *Remote Sensing*, vol. 13, no. 10, Art. no. 10, Jan. 2021, doi: 10.3390/rs13101887.

[31] A. K. Shukla, R. Shree, and J. Narayan, "Combining Fusion-Based Thresholding and Non-Linear Diffusion for Improved Speckle Noise Mitigation in SAR Images," *Applied Sciences*, vol. 14, no. 19, Art. no. 19, Jan. 2024, doi: 10.3390/app14198985.

[32] H. Choi and J. Jeong, "Speckle Noise Reduction Technique for SAR Images Using Statistical Characteristics

of Speckle Noise and Discrete Wavelet Transform," *Remote Sensing*, vol. 11, no. 10, Art. no. 10, Jan. 2019, doi: 10.3390/rs11101184.

[33] P. Kupidura, "COMPARISON OF FILTERS DEDICATED TO SPECKLE SUPPRESSION IN SAR IMAGES," *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. XLI-B7, pp. 269–276, Jun. 2016, doi: 10.5194/isprs-archives-XLI-B7-269-2016.

[34] 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.

[35] H. Cantalloube and C. Nahum, "How to Compute a Multi-Look SAR Image?," Jan. 2000.

[36] 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.

[37] D. M. Olson and E. Dinerstein, "The Global 200: Priority Ecoregions for Global Conservation," *Annals of the Missouri Botanical Garden*, vol. 89, no. 2, pp. 199–224, 2002, doi: 10.2307/3298564.

[38] Gorica Bratic and Maria Antonia Brovelli, "Map Of Land Cover Agreement - MOLCA." Accessed: Jun. 06, 2024. [Online]. Available: https://zenodo.org/records/8071675

[39] A. Sorriso, D. Marzi, and P. Gamba, "A General Land Cover Classification Framework for Sentinel-1 SAR Data," in *2021 IEEE 6th International Forum on Research and Technology for Society and Industry (RTSI)*, Sep. 2021, pp. 211–216. doi: 10.1109/RTSI50628.2021.9597319.

[40] M. Rußwurm and M. Körner, "Self-attention for raw optical Satellite Time Series Classification," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 169, pp. 421–435, Nov. 2020, doi: 10.1016/j.isprsjprs.2020.06.006.

[41] R. Sedona, C. Paris, L. Tian, M. Riedel, and G. Cavallaro, "An Automatic Approach for the Production of a Time Series of Consistent Land-Cover Maps Based on Long-Short Term Memory," in *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, Jul. 2022, pp. 203–206. doi: 10.1109/IGARSS46834.2022.9883655.

[42] G. Perantoni, G. Weikmann, and L. Bruzzone, "Bayesian Modelling of Multi-Year Crop Type Classification Using Deep Neural Networks and Hidden Markov Models," in *IGARSS 2024 - 2024 IEEE International Geoscience and Remote Sensing Symposium*, Jul. 2024, pp. 941–945. doi: 10.1109/IGARSS53475.2024.10642432.

[43] M. Rußwurm and M. Körner, "Temporal Vegetation Modelling Using Long Short-Term Memory Networks for Crop Identification from Medium-Resolution Multi-spectral Satellite Images," in *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Jul. 2017, pp. 1496–1504. doi: 10.1109/CVPRW.2017.193.

[44] A. Vaswani *et al.*, "Attention is All you Need," in *Advances in Neural Information Processing Systems*, Curran Associates, Inc., 2017. Accessed: Oct. 23, 2024. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html

[45] 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, Art. no. 5, Jan. 2019, doi: 10.3390/rs11050523.

[46] R. Interdonato, D. Ienco, R. Gaetano, and K. Ose, "DuPLO: A DUal view Point deep Learning architecture for time series classificatiOn," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 149, pp. 91–104, Mar. 2019, doi: 10.1016/j.isprsjprs.2019.01.011.

[47] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National Science Review*, vol. 5, no. 1, pp. 44–53, Jan. 2018, doi: 10.1093/nsr/nwx106.

[48] L. Bruzzone, "Multisource Labeled Data: an Opportunity for Training Deep Learning Networks," in *IGARSS 2019 - 2019 IEEE International Geoscience and Remote Sensing Symposium*, Jul. 2019, pp. 4799–4802. doi: 10.1109/IGARSS.2019.8898311.

[49] G. Perantoni and L. Bruzzone, "A Novel Technique for Robust Training of Deep Networks With Multisource Weak Labeled Remote Sensing Data," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 60, pp. 1–15, 2022, doi: 10.1109/TGRS.2021.3091482.

[50] G. Perantoni and L. Bruzzone, "Robust Training of Deep Neural Networks with Weakly Labelled Data," in *Signal and Image Processing for Remote Sensing*, 3rd ed., CRC Press, 2024.

[51] G. Algan and I. Ulusoy, "Image classification with deep learning in the presence of noisy labels: A survey," *Knowledge-Based Systems*, vol. 215, p. 106771, Mar. 2021, doi: 10.1016/j.knosys.2021.106771.

[52] G. Perantoni and L. Bruzzone, "A deep multiple instance learning approach based on coarse labels for high-resolution land-cover mapping," in *Image and Signal Processing for Remote Sensing XXIX*, SPIE, Oct. 2023, pp. 127–140. doi: 10.1117/12.2679464.

[53] 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)*, Jul. 2001, pp. 187–189 vol.1. doi: 10.1109/IGARSS.2001.976097.

[54] 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.

[55] 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.

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

[57] 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.

[58] 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*, C. M. Neale and A. Maltese, Eds., Berlin, Germany: SPIE, Oct. 2018, p. 6. doi: 10.1117/12.2325546.

[59] H. Yang, J. Yu, Z. Li, and Z. Yu, "Non-Local SAR Image Despeckling Based on Sparse Representation," *Remote Sensing*, vol. 15, no. 18, Art. no. 18, Jan. 2023, doi: 10.3390/rs15184485.

[60] P. Singh, M. Diwakar, A. Shankar, R. Shree, and M. Kumar, "A Review on SAR Image and its Despeckling," *Arch Computat Methods Eng*, vol. 28, no. 7, pp. 4633–4653, Dec. 2021, doi: 10.1007/s11831-021-09548-z.

[61] "Filters (Spatial): Median - MIPAV." Accessed: Oct. 13, 2024. [Online]. Available: https://mipav.cit.nih.gov/pubwiki/index.php/Filters_(Spatial):_Median

[62] William K. Pratt, *Digital Image Processing: PIKS Scientific Inside*, vol. 4. Hoboken, New Jersey: Wiley-interscience, 2007.

[63] 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.

[64] Rafael Gonzales, *Digital image processing 4th Edition*. London: Pearson.

[65] O. Oktay *et al.*, "Attention U-Net: Learning Where to Look for the Pancreas," May 20, 2018, *arXiv*: arXiv:1804.03999. doi: 10.48550/arXiv.1804.03999.

[66] H. Cao *et al.*, "Swin-Unet: Unet-Like Pure Transformer for Medical Image Segmentation," in *Computer Vision – ECCV 2022 Workshops*, L. Karlinsky, T. Michaeli, and K. Nishino, Eds., Cham: Springer Nature Switzerland, 2023, pp. 205–218. doi: 10.1007/978-3-031-25066-8_9.

[67] D. Marzi, J. I. S. Jara, and P. Gamba, "A 3-D Fully Convolutional Network Approach for Land Cover Mapping Using Multitemporal Sentinel-1 SAR Data," *IEEE Geoscience and Remote Sensing Letters*, vol. 21, pp. 1–5, 2024, doi: 10.1109/LGRS.2023.3332765.

[68] 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.

[69] D. Marzi and P. Gamba, "Inland Water Body Mapping Using Multi-temporal Sentinel-1 SAR Data," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. PP, pp. 1–1, Nov. 2021, doi: 10.1109/JSTARS.2021.3127748.

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

[71] M. L. Fonteh, F. Theophile, M. L. Cornelius, R. Main, A. Ramoelo, and M. A. Cho, "Assessing the Utility of Sentinel-1 C Band Synthetic Aperture Radar Imagery for Land Use Land Cover Classification in a Tropical Coastal Systems When Compared with Landsat 8," *Journal of Geographic Information System*, vol. 8, no. 4, Art. no. 4, Jul. 2016, doi: 10.4236/jgis.2016.84041.

[72] M. Buchhorn, M. Lesiv, N.-E. Tsendbazar, M. Herold, L. Bertels, and B. Smets, "Copernicus Global Land Cover Layers—Collection 2," *Remote Sensing*, vol. 12, no. 6, Art. no. 6, Jan. 2020, doi: 10.3390/rs12061044.

[73] D. Lu and Q. Weng, "A survey of image classification methods and techniques for improving classification performance," *International Journal of Remote Sensing*, vol. 28, no. 5, pp. 823–870, Mar. 2007, doi: 10.1080/01431160600746456.

[74] S. Hao, Y. Zhou, and Y. Guo, "A Brief Survey on Semantic Segmentation with Deep Learning," *Neurocomputing*, vol. 406, pp. 302–321, Sep. 2020, doi: 10.1016/j.neucom.2019.11.118.

[75] N. Teimouri, M. Dyrmann, and R. N. Jørgensen, "A Novel Spatio-Temporal FCN-LSTM Network for Recognizing Various Crop Types Using Multi-Temporal Radar Images," *Remote Sensing*, vol. 11, no. 8, Art. no. 8, Jan. 2019, doi: 10.3390/rs11080990.

[76] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds., Cham: Springer International Publishing, 2015, pp. 234–241. doi: 10.1007/978-3-319-24574-4_28.

[77] T.-T. Tran and V.-T. Pham, "Fully convolutional neural network with attention gate and fuzzy active contour model for skin lesion segmentation," *Multimedia Tools and Applications*, vol. 81, Apr. 2022, doi: 10.1007/s11042-022-12413-1.

[78] Z. Liu *et al.*, "Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows," presented at the Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 10012–10022. Accessed: Oct. 14, 2024. [Online]. Available: https://openaccess.thecvf.com/content/ICCV2021/html/Liu_Swin_Transformer_Hierarchical_Vision_Transformer_Using_Shifted_Windows_ICCV_2021_paper

[79] L. Yu, Z. Li, J. Zhang, and Q. Wu, "Self-attention on Multi-Shifted Windows for Scene Segmentation," Jul. 10, 2022, *arXiv*: arXiv:2207.04403. doi: 10.48550/arXiv.2207.04403.

[80] L. Maggiolo, D. Solarna, G. Moser, and S. B. Serpico, "Optical-SAR Decision Fusion with Markov Random Fields for High-Resolution Large-Scale Land Cover Mapping," in *IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium*, Kuala Lumpur, Malaysia: IEEE, Jul. 2022, pp. 5508–5511. doi: 10.1109/IGARSS46834.2022.9884751.

[81] S. Z. Li, *Markov Random Field Modeling in Image Analysis*. in Advances in Pattern Recognition. London: Springer London, 2009. doi: 10.1007/978-1-84800-279-1.

[82] Swain, "Bayesian Classification in a Time-Varying Environment," *IEEE Trans. Syst., Man, Cybern.*, vol. 8, no. 12, pp. 879–883, 1978, doi: 10.1109/TSMC.1978.4309889.

[83] C. M. Bishop, *Pattern recognition and machine learning*. in Information science and statistics. New York: Springer, 2006.

[84] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. in Adaptive computation and machine learning. Cambridge, Mass: The MIT press, 2016.

[85] M. Alameh, Y. Abbass, A. Ibrahim, G. Moser, and M. Valle, "Touch Modality Classification Using Recurrent Neural Networks," *IEEE Sensors J.*, vol. 21, no. 8, pp. 9983–9993, Apr. 2021, doi: 10.1109/JSEN.2021.3055565.

[86] A. Trucco, A. Barla, R. Bozzano, S. Pensieri, A. Verri, and D. Solarna, "Introducing Temporal Correlation in Rainfall and Wind Prediction From Underwater Noise," *IEEE J. Oceanic Eng.*, vol. 48, no. 2, pp. 349–364, Apr. 2023, doi: 10.1109/JOE.2022.3223406.

[87] S. Jamali, P. Jönsson, L. Eklundh, J. Ardö, and J. Seaquist, "Detecting changes in vegetation trends using time series segmentation," *Remote Sensing of Environment*, vol. 156, pp. 182–195, Jan. 2015, doi: 10.1016/j.rse.2014.09.010.

[88] G. A. Afuye *et al.*, "Global trend assessment of land use and land cover changes: A systematic approach to future research development and planning," *Journal of King Saud University - Science*, vol. 36, no. 7, p. 103262, Aug. 2024, doi: 10.1016/j.jksus.2024.103262.

[89] D. McNeall, P. R. Halloran, P. Good, and R. A. Betts, "Analyzing abrupt and nonlinear climate changes and their impacts," *WIREs Climate Change*, vol. 2, no. 5, pp. 663–686, Sep. 2011, doi: 10.1002/wcc.130.

[90] K. Meshkini, F. Bovolo, and L. Bruzzone, "A Multi-Feature Hyper-Temporal Change Vector Analysis Method For Change Detection In Multi-Annual Time Series Of HR Satellite Images," *IGARSS 2023 - 2023 IEEE International Geoscience and Remote Sensing Symposium*, 2023.

[91] J. Seo, W. Park, and T. Kim, "Feature-Based Approach to Change Detection of Small Objects from High-Resolution Satellite Images," *Remote Sensing*, vol. 14, no. 3, p. 462, Jan. 2022, doi: 10.3390/rs14030462.

[92] J. Li, S. Zhu, Y. Gao, G. Zhang, and Y. Xu, "Change Detection for High-Resolution Remote Sensing Images Based on a Multi-Scale Attention Siamese Network," *Remote Sensing*, vol. 14, no. 14, p. 3464, Jul. 2022, doi: 10.3390/rs14143464.

[93] K. Meshkini, F. Bovolo, and L. Bruzzone, "Multi-Annual Change Detection using a Weakly Supervised 3D CNN in HR SITS," *IEEE Geosci. Remote Sens. Lett*, 2023.

[94] A. K. Bhandari, A. Kumar, and G. K. Singh, "Feature Extraction using Normalized Difference Vegetation Index (NDVI): A Case Study of Jabalpur City," *Procedia Technology*, vol. 6, pp. 612–621, 2012, doi: 10.1016/j.protcy.2012.10.074.

[95] B. Gao, "NDWI—A normalized difference water index for remote sensing of vegetation liquid water from space," *Remote Sensing of Environment*, vol. 58, no. 3, pp. 257–266, Dec. 1996, doi: 10.1016/S0034-

4257(96)00067-3.

[96] M. L. García and V. Caselles, “Mapping burns and natural reforestation using Thematic Mapper data,” *Geocarto International*, vol. 6, no. 1, pp. 31–37, 1991.

[97] E. Alcaras, D. Costantino, F. Guastaferro, C. Parente, and M. Pepe, “Normalized Burn Ratio Plus (NBR+): A New Index for Sentinel-2 Imagery,” *Remote Sensing*, vol. 14, no. 7, p. 1727, Apr. 2022, doi: 10.3390/rs14071727.

[98] A. R. Huete, “A soil-adjusted vegetation index (SAVI),” *Remote Sensing of Environment*, vol. 25, no. 3, pp. 295–309, Aug. 1988, doi: 10.1016/0034-4257(88)90106-X.

[99] C. He, P. Shi, D. Xie, and Y. Zhao, “Improving the normalized difference built-up index to map urban built-up areas using a semiautomatic segmentation approach,” *Remote Sensing Letters*, vol. 1, no. 4, pp. 213–221, Dec. 2010, doi: 10.1080/01431161.2010.481681.

[100] D. Perez, Y. Lu, C. Kwan, Y. Shen, K. Koperski, and J. Li, “Combining Satellite Images with Feature Indices for Improved Change Detection,” in *2018 9th IEEE Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*, New York City, NY, USA: IEEE, Nov. 2018, pp. 438–444. doi: 10.1109/UEMCON.2018.8796538.

[101] J. V. D. Prasad, M. Sreelatha, and K. SuvarnaVani, “V-BANet: Land cover change detection using effective deep learning technique,” *Ecological Informatics*, vol. 75, p. 102019, Jul. 2023, doi: 10.1016/j.ecoinf.2023.102019.

[102] T. Boston, A. Van Dijk, and R. Thackway, “Convolutional Neural Network Shows Greater Spatial and Temporal Stability in Multi-Annual Land Cover Mapping Than Pixel-Based Methods,” *Remote Sensing*, vol. 15, no. 8, p. 2132, Apr. 2023, doi: 10.3390/rs15082132.

[103] Y. T. Solano-Correa, K. Meshkini, F. Bovolo, and L. Bruzzone, “A land cover-driven approach for fitting satellite image time series in a change detection context,” *Proc. SPIE 11533, Image and Signal Processing for Remote Sensing XXVI*, 2020.

[104] 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*, vol. 58, no. 3, pp. 2150–2164, Mar. 2020, doi: 10.1109/TGRS.2019.2953652.

[105] 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*, Jul. 2018, pp. 1946–1949. doi: 10.1109/IGARSS.2018.8519264.

[106] 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, 2019.

[107] G. Yoshizawa, “Bayesian Online Change Point Detection for Baseline Shifts,” *Stat., optim. inf. comput.*, vol. 9, no. 1, pp. 1–16, Dec. 2020, doi: 10.19139/soic-2310-5070-1072.