Reinforcing the AI4EU Platform by Advancing Earth Observation Intelligence, Innovation and Adoption

D4.3: Machine learning models for EO

Grant Agreement ID	101016798	**Acronym**	AI4COPERNICUS
Project Title	Reinforcing the AI4EU Platform by Advancing Earth Observation Intelligence, Innovation and Adoption		
Start Date	01/01/2021	**Duration**	36 Months
Project URL	https://ai4copernicus-project.eu/		
Contractual due date	30/06/2022	**Actual submission date**	30/06/2022
Nature	R = Document, report	**Dissemination Level**	PU = Public
Author(s)	Lorenzo Bruzzone, Giulio Weikmann (UNITN), Mihai Alexe (ECMWF), David Hassan (TAS), Omar Barrilero (SatCen)		
Contributor(s)			
Reviewer(s)	Mihai Alexe (ECMWF)		

This project has received funding from the *European Union's Horizon 2020 research and innovation programme* under grant agreement No 101016798.

Document Revision History *(including peer reviewing & quality control)*

Version	Date	Changes	Contributor(s)
v0.1	17/03/22	Table of Content	
v.02	30/05/22	All sections filled	UniTN, SatCen, Thales, ECMWF
v1	16/06/22	Deliverable ready for QA	
v2	24/06/22	Deliverable submitted (pending review by EC)	

Executive Summary

The following deliverable of WP4 (Implementation, customisation, integration and testing) describes the work done in Task 4.4 (Machine learning models for EO). In particular, we present the machine learning algorithms and models identified and integrated taking into account the inputs from WP2. We describe the architectures tailored to the processing of Copernicus data optimised on the different user cases.

This deliverable presents the services available as bootstrapping services deployed in the AI4Copernicus infrastructure, with a focus on the three machine learning models tailored to Sentinel 2 data: a long short-term memory network, a pixel-level classification deep network on S2 patches, and a probabilistic downscaling of CAMS air quality model data.

Table of Contents

List of Figures

List of Tables

List of Terms & Abbreviations

Abbreviation	Definition
AC	Atmospheric Composition
AQ	Air Quality
CAMS	Copernicus Atmosphere Monitoring Service
EAC4	ECMWF Atmospheric Composition Reanalysis 4
EO	Earth Observation
F1	F-Score
GAN	Generative Adversarial Network
GHG	Greenhouse gas
LSTM	Long Short-Term Memory
OA	Overall Accuracy

S2	Sentinel-2
VAE	Variational Autoencoder
WGAN-GP	Wasserstein Generative Adversarial Network with Gradient Penalty

1 Introduction

This is the third deliverable of WP4 (Implementation, customisation, integration and testing) and, more specifically, Task 4.4 (Machine learning models for EO).

1.1 Purpose and Scope

The purpose of this deliverable is to present the machine learning models for EO developed in WP5 (Bootstrapping AI4Copernicus with high-impact services). The architectures and the algorithms developed are oriented to the processing of Copernicus data, considering both single-date and time series of remote sensing images.

D4.3 analyses the machine learning model and algorithms that have been developed and made available in WP5 for the four high-impact domains (energy, security, agriculture, and health).

1.2 Approach for Work Package and Relation to other Work Packages and Deliverables

Work package WP4 (Implementation, customisation, integration and testing) started on M4 and ends on M24 of the project. It is led by partner CF with the collaboration of partners NCSR-D, UoA, TAS, ECMWF and UNITN. WP4 demonstrates the usability of the solution by the reference test and the use cases selected in the open calls (WP6).

WP4 has the following five tasks:

- Task 4.1 Integration of AI4EU platform with CREODIAS/WEKEO (M4-M12, lead: CF, contributor: TAS). The technical contribution of this task is the configuration of the environment to accommodate the requirements identified in the WP2.
- Task 4.2 Integration of tools for transformation, querying, interlinking and federating big linked geospatial data (M4-M12, lead: UoA). The technical contribution of this task is the integration of the linked data suite, developed by UoA, to the platform.
- Task 4.3 Implementation of the semantic catalogue and the semantic search and discovery functionality (M4-M12, lead: UoA, contributor: NCSR-D). The technical contribution of this task is the implementation of the semantic catalogue designed in Task 3.2.
- Task 4.4 Machine learning models for EO (M4-M12, lead: UNITN, contributors: NCSR-D, ECMWF). The technical contribution of this task is the identification and integration of different supervised machine learning techniques and models, taking into account the inputs from WP2.
- Task 4.5 Testing and operation of bootstrapping services (M7-M18, lead: CF, contributors: NCSR-D, UoA). The technical contribution of this task is the availability of dedicated environments for the use cases.

The present deliverable D4.3 is the third deliverable of WP4 and contains the contributions of the project in Task 4.4.

The machine learning algorithms and models identified in WP2 are designed and implemented in WP5 (Bootstrapping AI4Copernicus with high-impact services). WP5 started in M4 and will end on M30. It is led by SatCen with the participation of partners NCSR-D, UoA, THA, ECMWF, CF, UNITN, Equinor, and Blue-Sight.

The following tasks of WP5 are relevant to WP4:

- Task 5.1 Agriculture bootstrapping services and resources (M4-M30, lead: UniTN, contributors: THA, UoA, NCSR-D, ECMWF, CF). The technical contribution of this task is the development of machine learning algorithms for EO in the agriculture domain.
- Task 5.2 Energy bootstrapping services and resources (M4-M30, lead: Equinor, contributors: NCSR-D, UoA, CF). The technical contribution of this task is the development of machine learning algorithms for EO in the energy domain.
- Task 5.3 Security bootstrapping services and resources (M4-M30, lead: SatCen, contributors: NCSR-D, UoA, CF). The technical contribution of this task is the development of machine learning algorithms for EO in the security domain.
- Task 5.4 Health bootstrapping services and resources (M4-M30, lead: ECMWF, contributors: NCRS-D, CF). The technical contribution of this task is the development of machine learning algorithms for EO in the health domain.

The following deliverable of WP5 is relevant to WP4:

- D5.1 Bootstrapping services and resources I (M12, DEM, PU, SatCen). This deliverable describes the services and the resources developed and implemented in the project. It also provides a documentation of each service referred to the application deployed in the AI4Copernicus infrastructure.

1.3 Organization of the Deliverable

The deliverable is structured in accordance with the template and guidelines provided by the EC and is organised as follows. Section 2 contains the machine learning algorithms and models developed for the first batch of open calls. In particular, subsection 2.1 contains the algorithms used to perform the pre-processing of remote sensing data and non-machine learning models (SatCen). Subsection 2.2 (ECMWF), 2.3 (Thales Six), and 2.4 (University of Trento) focuses on the machine learning models developed within the project. In section 3, the performance of the models are analysed and the validation is performed. Lastly, section 4 draws the conclusions of the D4.3

2 Machine learning models for EO

In this Deliverable, we present different supervised machine learning models integrated by taking into consideration the inputs from WP2. We propose several architectures based on Convolutional Neural Networks, Generative Adversarial Networks, and Recurrent Neural Networks (in particular

Long-Short Term Memory Neural Networks). The proposed architectures are application-oriented, focusing on the four domains of the Open Calls challenges: Security, Agriculture, Health, and Energy, in order to facilitate the implementation and the success of the winning projects selected. The proposed architectures can easily be adapted and used on cross-domain problems, so that the user can exploit the algorithm preferred. The algorithms presented rely on the Copernicus data, with a particular focus on Sentinel-1 and Sentinel-2 data, and will be optimised with respect to the user cases.

2.1 Pre-Processing

The architectures proposed rely on pre-processed Copernicus images, where sensor and platform-specific radiometric and geometric distortions are corrected. Due to the requirements of this pre-processing step, this deliverable will contain the description of the pre-processing deployed and made available to the user-cases. By adopting the pre-processing algorithm, standardised corrected images can be extracted and fed to the different architectures proposed.

2.1.1 Sentinel-1 Pre-Processing

In order to be used as inputs of Machine Learning models, Sentinel-1 data has to be preprocessed to apply mainly radiometric and geometric corrections.

The radiometric corrections have two main objectives: to reduce the noise and to calibrate the pixel values.

Regarding the noise, different source are considered:

- Thermal Noise: Level-1 products of Sentinel-1 provide a noise LUT that can be applied to remove the noise.
- Border Noise: This noise (artefacts at the image borders) is present in GRD L1 products due to the processes carried out by the Instrument Processing Facility when converting RAW data into L1.
- Speckle: It is caused by random constructive and destructive interference of the de-phased but coherent return waves scattered by the elementary scatters within each resolution cell. By applying spatial filters or multilooking, the speckle noise can be reduced.

The calibration of SAR images is important to be able to convert the pixel values (usually optimized to minimize the space required for its storage) to a physical value (radar backscatter of the reflecting surface) that can be used for quantitative processing and for comparison with other images (from the same or even other sensor). Sentinel-1 Level-1 products provide calibration Look Up Tables (LUTs) to produce beta, sigma or gamma nought.

Geometric corrections are needed to compensate distorsions generated by topographical variations. The algorithm exploits available orbit state vector information in the metadata and external precise orbit files, the radar timing annotations, the slant to ground range conversion parameters together with the reference DEM data to derive the precise geolocation information.

2.1.2 Sentinel-2 Pre-Processing

In order to be used as inputs of Machine Learning Models, Sentinel-2 data has to be preprocessed to apply mainly radiometric and geometric corrections.

The radiometric correction includes the conversion from pixel values to radiance/reflectance that can be used in further processing and the atmospheric corrections (in case the input images are S2 L1C products). The L1C products provide the top of atmosphere (TOA) reflectance. TOA reflectance could be enough for certain processes, but when the process requires the use of images acquired at different times (e.g. time series analysis), the atmospheric effects (that will be different due to different atmospheric conditions) on the reflectance has to be minimised. The Sen2Cor tool can be used to apply the atmospheric corrections obtaining bottom of atmosphere reflectances (BOA). This product (L2A) is the same that the one generated by ESA (there are small differences due to the use of a different Digital Elevation Model and if the user selects a configuration different to the default one).

The geometric correction performed in the S2 pre-processing includes the resample of the S2 bands. Some Machine Learning Models require input images at the same resolution, but the S2 bands are at 10, 20 or 60 metres.

Finally, as some models are specifically designed for land or sea applications (e.g. crop classification, ship detection), the pre-processing can also perform some filters to remove land/sea pixels and cloud pixels.

2.1.3 Sentinel-2 Change Detection

Most change detection processing chains can be simplified as shown in the figure below (Fig. 2.1):

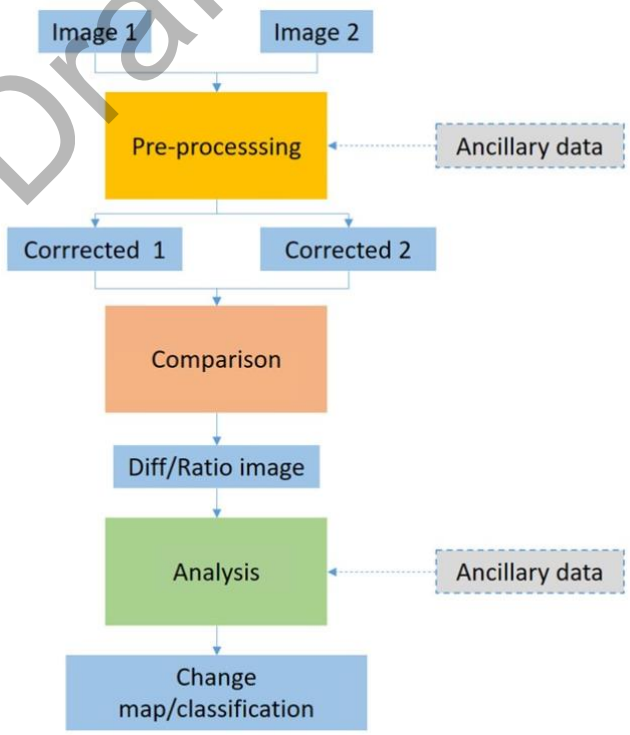

Figure 2.1. Generic workflow for change detection.

The first step consists of the preprocessing of Sentinel-2 data to prepare the data for the comparison. The main substeps are:

- Resample selected bands and convert the DN to reflectance
- Resample scene classification to same resolution
- Relative radiometric correction (to mitigate differences in light and atmospheric conditions):
  - Use scene classification (from Sentinel-2 products) to use only pixels that have the same classification in both images (and are not snow, clouds, shadows...)
  - Differences of those pixels are computed and normalized for all the selected bands o Computation of amplitude of the vector of differences
  - Selection of 50% of pixels with less changes -> Mask for relative correction o Use of that mask to perform a linear regression of the input images
- Subset the images to AoI (this is not done before in order to have more info to perform the relative correction)

In the "Comparison" step two main substeps are carried out:

- Generate cloud mask: we rely on scene classification from S2 products (but other approaches are also possible). A merge of both cloud masks (image 1 and image 2) is done.
- Change Vector Analysis I: Differences are computed and normalized.

For the "Analysis" step, using the normalized differences, amplitude and the "direction" (using a reference vector) of the change are computed (Change Vector Analysis).

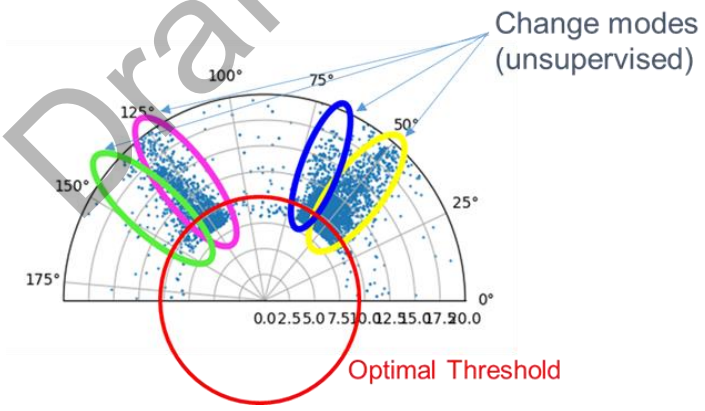

Figure 2.2. Example of polar representation of changes. The points above a threshold in amplitude are classified as changes. Different types of changes are assigned according to the different angles.

After that, there are two substeps: the thresholding to identify changes and finally the classification of these changes.

In the thresholding step, they are classified as "change" the pixels with big changes in only one band and pixels with not a big change in an specific band but with small-medium changes in most of the bands:

- Changes in any of the bands:
  - For every band of differences, it is considered that they follow normal distribution: It is classified as change based on the threshold computed from the confidence level
- Total change considering all the selected bands o For the total change (amplitude of change considering all the bands) it is considered that they follow a chi-squared distribution (considered independent variables). The threshold is computed using the confidence level

Finally, the k-means clustering algorithm is used to classify changes based on the "direction" obtained with CVA.

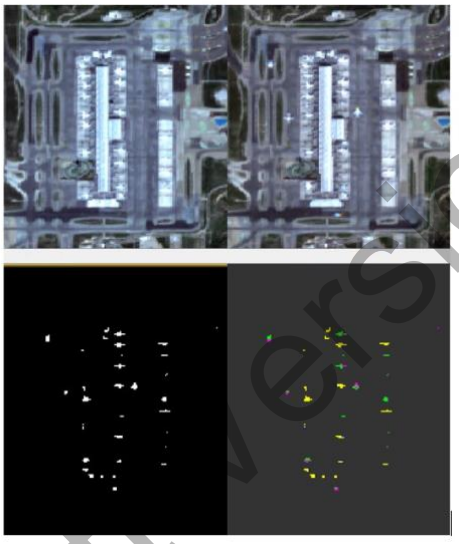

Figure 2.3. Example of S2 change detection over an airport.

2.1.4 Harmonization of Sentinel-2 data

Sentinel-2 image time series allow the analysis of the phenological evolution of the targets considered. This becomes particularly effective when the seasonality of the target allows the discrimination or the detection of certain physical characteristics based on a particular date. However, analysing time series of images introduces several challenges, mainly consisting in: (i) the cloud coverage problem that hampers the time series introducing missing information and reducing the usability of the images, and (ii) the spatial and temporal inconsistency of the remote sensed time series. To efficiently train a model on such time series, an harmonisation step is required to reproject the irregular data into a homogeneous grid.

Starting from Sentinel 2 images pre-processed using the algorithm described in Subsection 2.1.2, the harmonization algorithm aims at converting the irregular time series into a 12 monthly composites time series considering a pixel-based statistic-based approach for each tile analysed. Due to the high revisit time of Sentinel 2, enough images can be collected for each month, allowing an accurate reconstruction of the temporal signature for different cultivations.

Let us focus the attention on the set of *Q* sentinel images acquired within the *i*th month, with i ∈

[1,...,12].2 image ofLetthe$X$considered$_{ij}$ = dataset,[$x_{ij,1}$, x*i*-$_{ji,2}$th,where.month, $x_{ij,Q}$]$x_{iT,1}$ =be[the$x_{ij,,11}$,ofmultitemporalx_{ji}N=9$_{,,12}$, ., $x$spectral$_{ij,,1N}$] is thechannels.spectralspectralmatrixForvectoreachassociatedofspectralthe firsttoband,Sentinelthethe*j*th

labeled pixel of the

$_{j}$made up

Q reflectance values are collapsed into a single one by computing their median. Let $\mathbb{M}\{\cdot\}$ be the median operator, the computation of the *i*th monthly composite is as follows:

$$x^{i}_{j,1} = \mathbb{M}\{x^{i}_{j,}{}^{1}1, x_{j}{}^{i,}{}^{2}1, ., x^{i}_{j,}{}^{Q}1\}$$

$$x^{i}_{j,2} = \mathbb{M}\{x^{i}_{j,}{}^{1}2, x_{j}{}^{i,}{}^{2}2, ., x^{i}_{j,}{}^{Q}2\}$$

$$\vdots$$

$$x^{i}_{j,N} = \mathbb{M}\{x^{i}_{j,}{}^{1}{}_{N}, x_{j}{}^{i,}{}^{2}{}_{N}, ., x^{i}_{j,}{}^{Q}{}_{N}\}$$

where $x_{ij,1}$ obtained= [$x_{ij,,11}$, x_{ji}multitemporal$_{,,12}$, ., $x_{ij,,1N}$] is thespectralspectral vectorvector ofis

themade*i*th monthlyup of NxMcomposite.features.AtMoreover,the end of thisthe step, the

median computation ignores cloudy, snowy and shadowy samples, according to the pre-processing script described above. If no cloud-free images are available within the month, the reflectance value is set to zero, which will be handled by the LSTM model after [1]. The harmonized monthly composite retrieved can be used for training the Long Short-Term Memory neural network described furtherly in the deliverable.

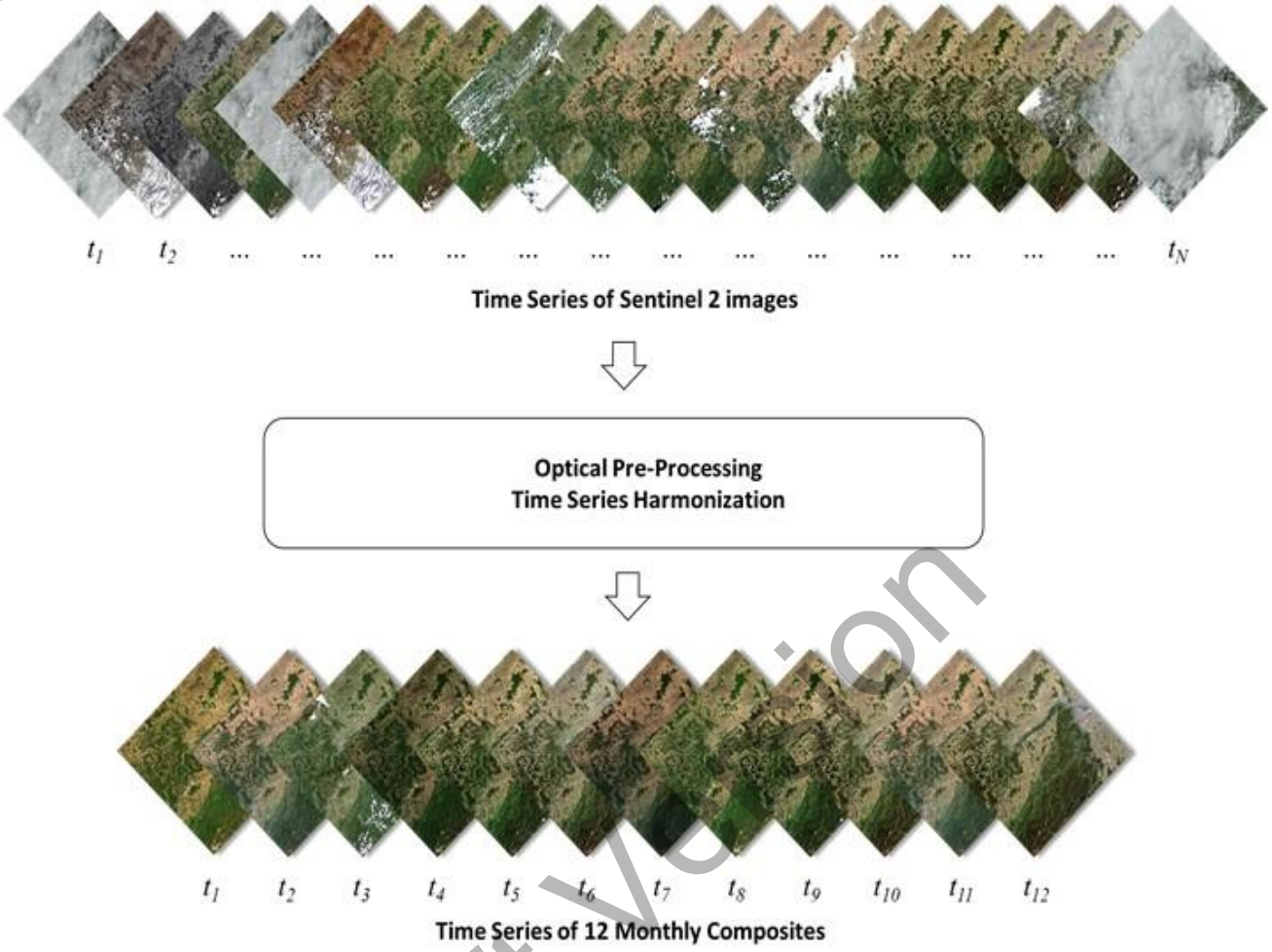

Figure 2.4. Optical Pre-Processing step which converts the initial irregular time series of Sentinel 2 images into an harmonized time series of 12 monthly composites.

2.2 Probabilistic downscaling of CAMS air quality model data using GANs

The AI4Copernicus Health Bootstrapping service and resources have been developed to address current public health and air pollution challenges using Earth observation data. The services are focused on probabilistic downscaling (super-resolution) of air quality (AQ) and atmospheric composition (AC) model output. Current AC / AQ models output forecasts at relatively low-resolution - e.g., ca. 80 km for the CAMS global reanalysis and 10 km for the CAMS-Regional analysis / forecast products. We note that the CAMS service is operated by ECMWF and CAMS output is freely available for download through the Copernicus Atmosphere Data Store.

Previous research has demonstrated that it is possible to make use of state-of-the-art deep learning architectures to downscale (i.e., increase the spatial or temporal resolution of) model output, thus allowing the identification of pollution or greenhouse gas (GHG) emission hotspots.

The probabilistic downscaling engine is based on a Wasserstein generative adversarial network with gradient penalty (WGAN-GP). Recent downscaling studies [2,3] have used WGAN-GP architectures

with considerable success, achieving high-resolution outputs of remarkable quality and diversity compared to alternative techniques such as variational autoencoders (VAEs) [4].

GAN-based models have achieved impressive performance in artificial high-resolution "image" generation for the Earth sciences. Conditional GANs [5] allow the generation of pseudo-ensembles (many high-resolution realisations that correspond to the same low-resolution input) that can be used to quantify the uncertainty in the high-resolution reconstructions. In essence, the trained GAN generator defines a *probabilistic* mapping between the low-resolution EAC4 input and the high-resolution CAMS-regional input.

The structure of the WGAN used by the downscaling service is inspired by the works of Leinonen et al. (2019) [6] and Price and Rasp (2022).

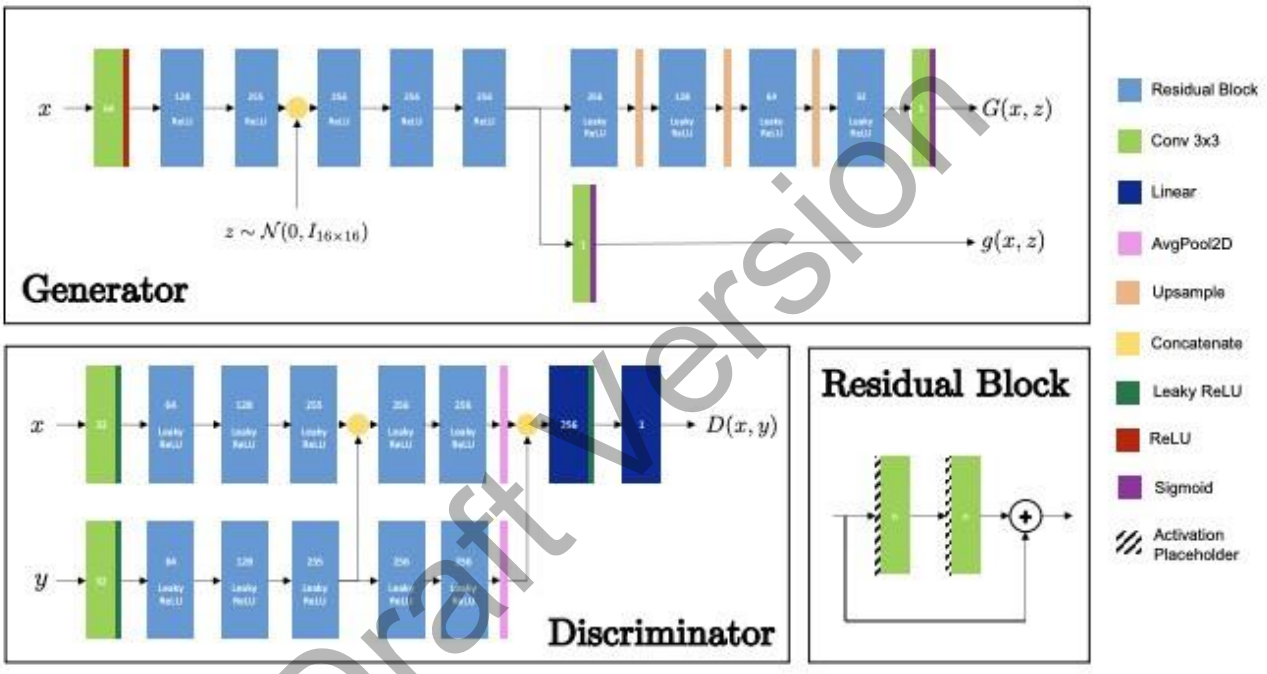

Figure 2.5. Structure of the GAN generator and discriminator used for downscaling (Price and Rasp, 2022).

Above we reproduce Figure 1 from the appendix of Price and Rasp (2022) showing the WGAN-GP architecture that we adopted for the CAMS downscaling service. The generator (top) outputs a high-resolution reconstruction **G(x, z)** of the CAMS-Regional fields (NO2, O3 or PM2.5) conditioned on the input x. The input vector x contains CAMS global reanalysis (EAC4), mid-resolution weather (ERA5) and high-resolution "static" field data (orography, built-area fraction). Here, z is the noise vector that is fed to the generator, whereas **g(x, z)** is a low-resolution output signal that is used to spatially bias-correct the output (EAC4 low-res vs. CAMS-regional hi-res) during pre-training. For more details on the pre-training scheme, see Price and Rasp (2022). The WGAN discriminator attempts to "separate" generated ("fake") images **G(x, z)** from the ground truth ("real") **y**, through the Wasserstein loss.

The generator and discriminator are trained adversarially:

- The discriminator minimises the WGAN-GP loss [7]:

$$L_D = D(x, G(x, z)) - D(x, y) + \lambda(\|\nabla_{\tilde{y}} D(x, \tilde{y})\|_2 - 1)^2,$$

where

$$\tilde{y} = \epsilon y + (1 - \epsilon) G(x, z), \quad \epsilon \sim U(0, 1),$$

- The generator loss incorporates a L1 penalty on the low- and high-res generator outputs (the so-called "content-losses" $\mathbf{L_{HR}}$ and $\mathbf{L_{LR}}$) plus the adversarial loss:

$$L_G = \mathbb{E}_{x,y}[\mathbb{E}_z[-D(x, G(x, z))] + \gamma_1 L_{LR} + \gamma_2 L_{HR}].$$

The WGAN-GP model code has been made available on Github: https://github.com/mishooax/ai4cop-health-cams. Interested users are encouraged to examine the code and suggest extensions / changes to the authors through the "Github issues" page.

2.3 Long Short-Term Memory Neural Network

The spatial, spectral, and temporal characteristics of the Sentinel 2 sensors allow for precise seasonal trend analysis, especially in the crop type mapping field. Moreover, specific bands are dedicated to the monitoring of the Red-Edge spectral range which can be exploited to extract extremely informative features for agricultural monitoring.

Long Short-Term Memory (LSTM) neural networks can store huge amounts of memory from previous samples and accurately model the seasonal trend of the targets, making them suitable for the crop type mapping problem. Moreover, LSTMs do not suffer from the vanishing gradient problem that affects vanilla Recurrent Neural Network (RNN), allowing the analysis of longer time series without losing information of the oldest samples. However, due to the complexity of the LSTMs with respect to the RNNs, the training phase is significantly heavier, with more parameters that need to be estimated.

In order to apply the LSTM to the EO problem, two main challenges have been identified: (1) time series acquired over different areas have to be harmonized from the temporal viewpoint, and (2) handle a severely imbalanced classification problem. The first challenge has been addressed considering the "Harmonization of Sentinel 2 data" algorithm defined in Section 2.1.4 of the Deliverable, where different time series that are hampered by cloud coverage, or that may change in length due to the overlapping orbits of acquisition, are compressed into a standardised time series of 12 monthly composite, each representing a month in the agronomic year. The second challenge is specific for the crop type mapping problem, where agricultural areas are typically dominated by few common crops that are extensively cultivated. Minor crop types are less common, and their weights have to be tuned accordingly to their prior probabilities by the model.

In order to deal with the second challenge identified, the cost function of the LSTM is modified to take into account the prior probabilities of the different semantic classes. In particular, the cross-entropy loss function is calculated at each training step between the predicted and the ground truth

class. When the predicted probability diverges from the actual label, the cross-entropy loss increases, penalising the choice of the model. Considering a binary classification problem, let *y* represent whether a sample is correctly classified or not, and $\hat{y}$ the predicted probability of a correct classification (the output of the softmax function). We can define the binary cross-entropy H_b as:

$$H_b(z, \hat{y}) = -\left[y\ log(\hat{y}) + (1 - y)\ log(1 - \hat{y})\right]$$

In a multiclass classification problem, the binary cross-entropy loss function is generalised considering separate loss for each class label per observation. The cross-entropy loss function can be rewritten as a sum of the separate losses averaged over the *N* samples considered. H_b

$$(w) = -\frac{1}{N}\sum_{n=1}^{N} H(p_n, o_n) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{i}$$

of the true predicted $p_i, log(o_i)$

be optimised.p^i The loss is then backpropagatedo^i at each training step considering a RMSprop where is the probability label, the value and *w* the vector of weights to optimizer [8]. To handle the imbalanced classification challenge described before, the cross-entropy loss function is modified. Such modified cost function can be rewritten as: $H' = \sum U \quad nn_{max}\, u=1$

$_u H_u$

(with the higher number of training samples)n^{max} and the number of samples associated to the *u*th Where *U* is the number of classes and the number of samples associated to the dominant class class . Due to the modification of the cost function,n^u the output represents an approximation of a posterioriw_u probability. After the training of the LSTM, the model parameters are saved and another training of the networks occurs using the standard cross entropy loss.

The structure of the multitemporal deep learning architecture is summarised in Figure 2.5. The proposed architecture is a multi-layer LSTM, being able to exploit longer time series data than a single-layer one. In particular, the network consists of three layers with 200, 125, and 100 hidden units respectively. After the three LSTM hidden layers, a fully connected layer followed by a softmax layer provides the classification at pixel level. The structure of the network has been defined considering the TimeSen2Crop train, test and validation sets. TimeSen2Crop [9] is a pixel-based dataset containing more than 1 million crop samples of Sentinel 2 time series. The dataset provides information related to an agronomic year (September 2017 to August 2018) of the Austrian country, reporting information regarding the snow, shadows, and cloud information. TimeSen2Crop is a resource made available in the AI4Copernicus bootstrapping services and resources, allowing the user to develop its neural network using the training dataset provided.

In order to define the structure of the architecture, we followed a standard grid-search approach by testing the different combination of multiple network layers {2, 3, 4} and a varying number of cells per layer {100, 125, 200, 300}. The setup that obtained a higher accuracy value in the validation set has been implemented.

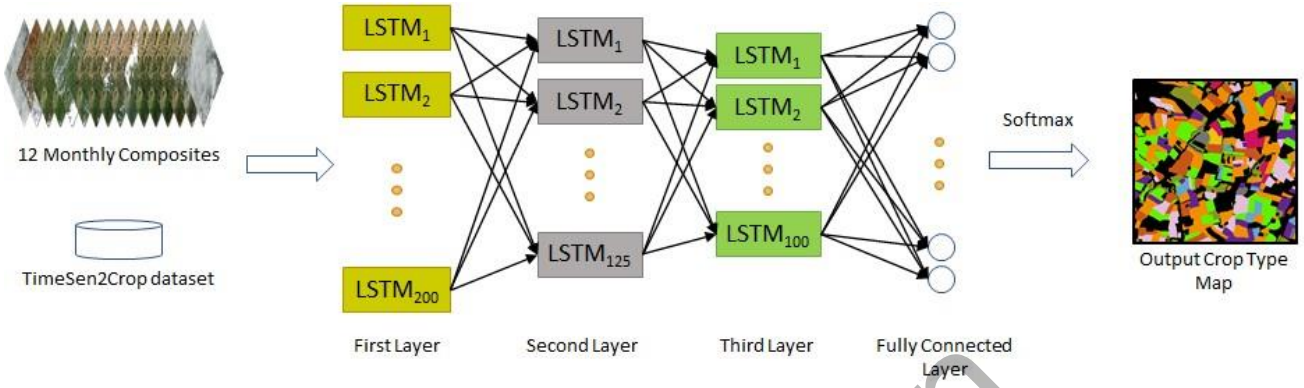

Figure 2.6. Architetcure of the LSTM neural network defined in the agriculture domain.

The user can train the architecture from scratch, either providing its own training dataset or deploying the TimeSen2Crop dataset. However, to allow the user to test the architecture on the user-case without the need of training from scratch, a pre-trained version of the architecture for the agricultural domain is available as a bootstrapping service. The pre-trained model has been trained on the TimeSen2Crop dataset considering one million labelled crop type samples subdivided into 15 different crop type classes. The classification scheme is reported in Table 2.1.

	ID	Class Name
	1	Legumes
	2	Grassland
	3	Maize
	4	Potato
	5	Sunflower
	6	Soy
	7	Winter Barley

	8	Winter Caraway
	9	Rye
	10	Rapeseed
	11	Beet
	12	Spring Cereals
	13	Winter Wheat
	14	Triticale
	15	Permanent Plantations

Table 2.1. The classification scheme of the pre-trained LSTM neural network available as a bootstrapping service.

2.4 Deep Network for pixel-level classification of S2 patches

The Sentinel-2 datasets provide multispectral images (13 bands) which contain far more information than the usual RGB images. While this huge amount of information expands the possibilities in terms of fine-grained detection and segmentation, it also considerably increases the complexity of the deep learning models required to perform segmentation on such data.

The model we use is based on a Unet architecture. Unet is a fully convolutional neural network that performs pixel-level image segmentation [10]. It takes the original image as input and outputs a segmented image.

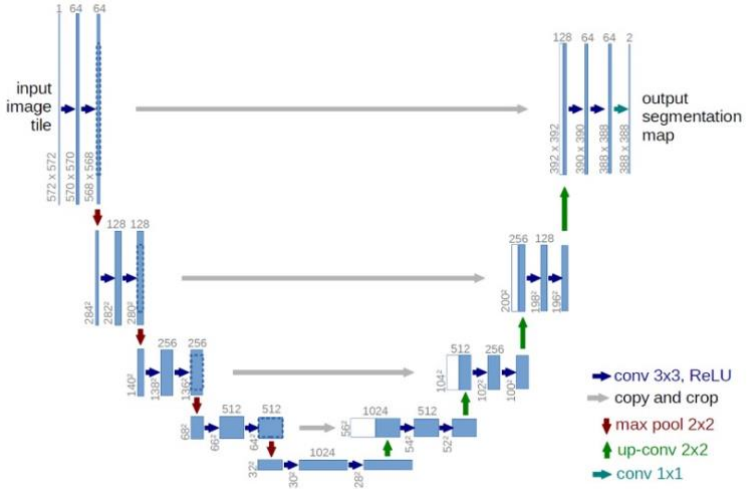

Figure 2.7. Unet architecture - "U-Net: Convolutional Networks for Biomedical Image Segmentation", University of Freiburg

The core of this auto-encoder architecture is a succession of convolutional networks. The first half of the model is an encoder, where features are extracted from the input data. The second part of the model is the decoder where the final segmentation image is constructed using the encoded data. Standard implementation of the Unet network deals with RGB and grayscale images. It usually outputs a binary image, which suits with a one class segmentation use case. However, it does not match with the requirements of the current use case. We modified the Unet to be able to process multi-spectral images as input. The exact number of input layers of the model is defined during the creation of the docker depending on the arguments specified by the user. We also modify the output of the Unet model to perform multi-label segmentation. In this case, the output is an image with the same height and width as the input, each pixel value corresponding to the classified label.

Concerning the loss function, we use categorical crossentropy to fit our multi-label segmentation use case. Loss function can be described as:

$$\ell(x, y) = L = \{l_1, \dots, l_N\}^\top, \quad l_n = -w_{y_n} \log \frac{\exp(x_{n,y_n})}{\sum_{c=1}^{C} \exp(x_{n,c})} \cdot 1\{y_n \neq \text{ignore\_index}\}$$

Where x is the input, y the target, w the weight, C the number of classes and N the minibatch dimension. Users can also select Focal loss instead of categorical crossentropy, which aims to focus on hard examples.

Using multi-spectral images as input complicates the training process. Same goes for the multi-label segmentation task compared to a binary segmentation. In a deep learning context, a more complex task can imply an increase in training data needs, an increase of the training time, a deeper network or even a drop in accuracy. To solve this issue, one solution is to use a model that has already been trained on a different set of data that appears to be close enough from the target use case. Doing so,

the weights of the model are not random and more likely to converge to an optimized solution while retraining it with the data provided by the user.

Inside our solution, users can choose to replace the encoder part of the model with a pretrained backbone. This backbone has been created using a self-supervised method, meaning that no annotated data has been required to generate it. Instead, a huge amount of unlabeled satellite images has been used during that self-supervised training phase, which is based on a siamese network approach [11] : the algorithm automatically creates positive and negativs samples and compares them to calculate a contrastive loss. This loss is further used to update model weights.

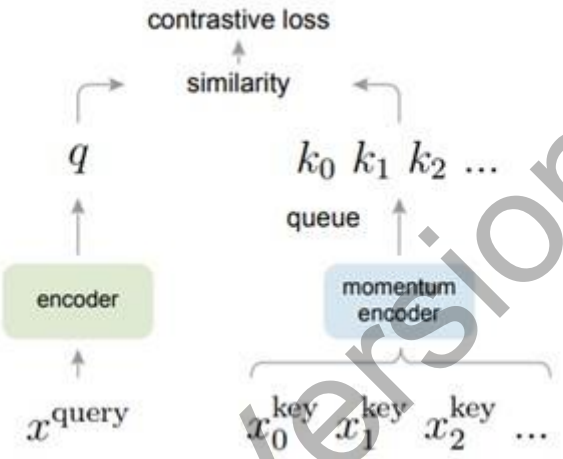

Figure 2.8. MoCo algorithm, "Momentum contrast for unsupervised visual representation learning" - Facebook AI Research

3 Performance Evaluation

3.1 Probabilistic downscaling of CAMS air quality model data (GAN)

The figure below shows the areal extent of the input datasets, as follows. Individual 128 x 128 CAMS-regional high-resolution data “patches” are shown in red (with overlap). The region covered by the CAMS-regional data is bounded by the green rectangle, while EAC4 data covers the blue rectangular area; note that the EAC4 data domain includes a “buffer” zone that is used to provide spatial “context” to the downscaling algorithm and help with EAC4-to-CAMS-regional bias-correction. The WGAN-GP generator receives low-resolution EAC4 data (16x16 or 32x32 image patches, the latter including spatial “context”), medium-res ERA5 weather data (64 x 64) and high-resolution orography + “built-fraction” data (512 x 512 pixels) as conditioning information. It produces 128 x 128 output fields (at the CAMS-regional spatial resolution of approximately 10km).

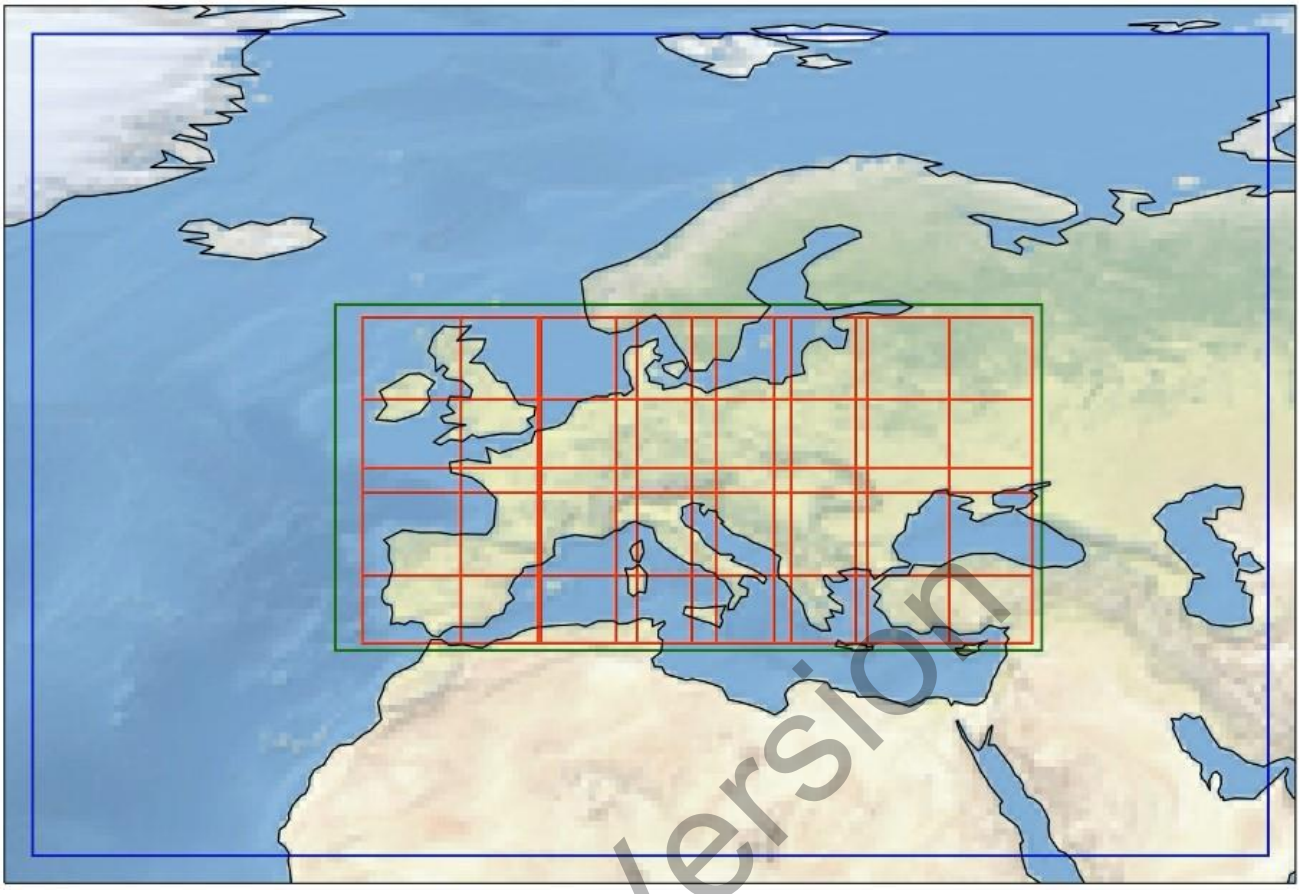

Fig 3.1. Input / output data domain for the CAMS downscaling service. EAC4 data covers the blue rectangular area; it includes a spatial "buffer" that provides the generator with "context" to allow bias correction between the EAC4 and CAMS-regional model fields. CAMS-Regional data lies inside the green rectangle, whereas the red squares (128 x 128 pixels) show the high-resolution patches ("regions") that the generator is trained on.

The data covers the period of Jan 1, 2014 through June 30, 2021 (as EAC4 is currently only available until June 2021). We split this range into training (2015 - 2019), validation (2020) and test (2021) datasets.

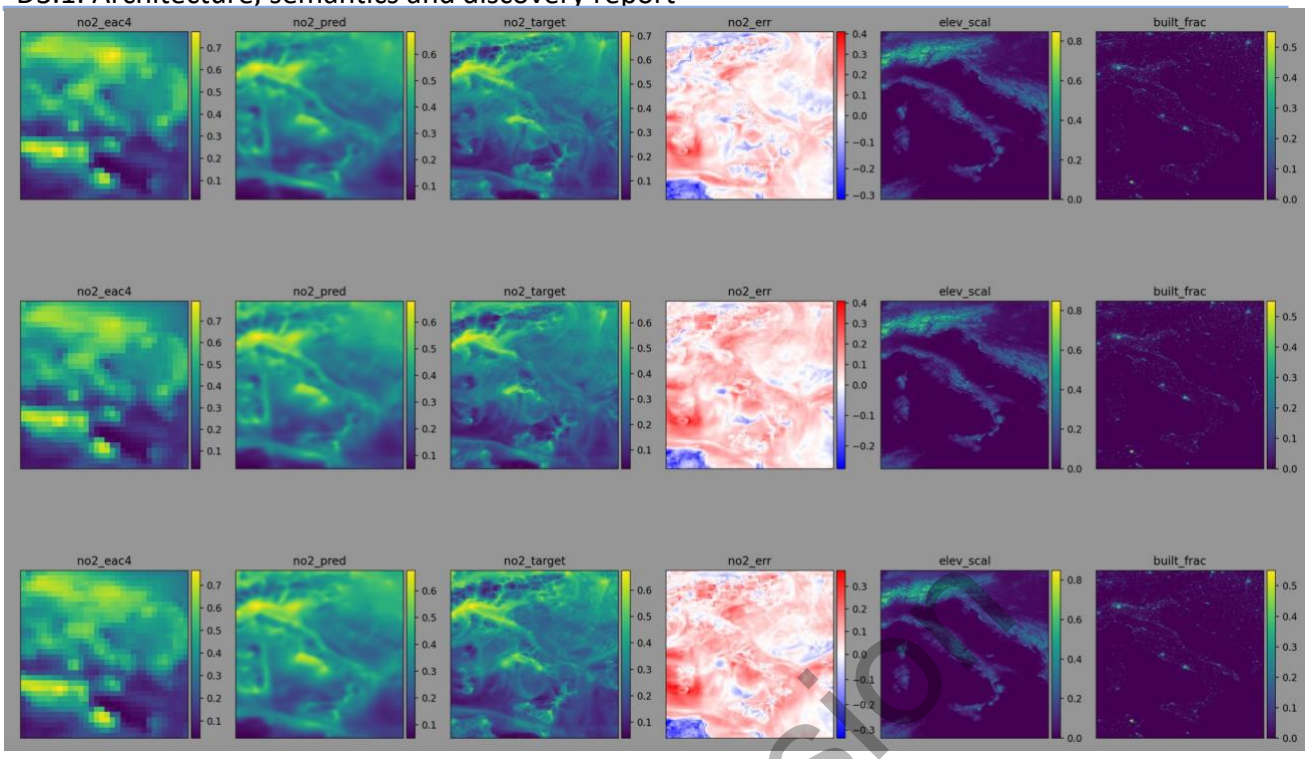

Fig 3.2. Inputs and outputs for the WGAN-GP downscaling model. The target is the high-resolution CAMS-Regional surface NO2 field over the Italian Peninsula (3rd column from left).

Sample WGAN-GP generated reconstructions of nitrogen dioxide (NO2) fields over the Italian Peninsula. Leftmost panels show the low-resolution EAC4 data - this includes a halo around the region of interest for spatial "context". The scaled elevation and built-fraction fields are shown in the rightmost columns. We can see that the generator is able to reproduce realistic, high-resolution features in the target NO2 fields.

The present service is meant only as a proof-of-concept, and we have not yet performed an exhaustive performance evaluation study. Several avenues of improvement exist, including the use of (1) proper score losses [10] for pre-training and/or adversarial training and (2) of self-supervised techniques for deep representation learning to construct custom content losses [12]. These extensions will be described in a future publication (in preparation). Potential users are encouraged to use the pre-trained WGAN-GP architecture for transfer learning, e.g. evaluate performance on the North-American domain using high-resolution data (e.g., AirNow) available there.

3.2 Long Short-Term Memory Neural Network

The accuracy of the Long Short-Term Memory neural network described in Section 2.3 has been assessed considering the TimeSen2Crop dataset provided as a resource in the bootstrapping services of WP5. The TimeSen2crop dataset is a pixel-based dataset made up of more than 1 million samples of Sentinel 2 time series associated with 16 different crop types. The dataset represents crop type samples acquired over 15 tiles in the Austrian

country, with acquisitions ranging from September 2017 to August 2018. In order to test the generalisation capabilities of the network, the 15 considered tiles have been subdivided into training set, validation set, and test set. The subdivision performed can be seen in Fig. Y. By employing such subdivision, we are performing analysis on statistically independent samples, since no spatial overlapping is considered between the different sets.

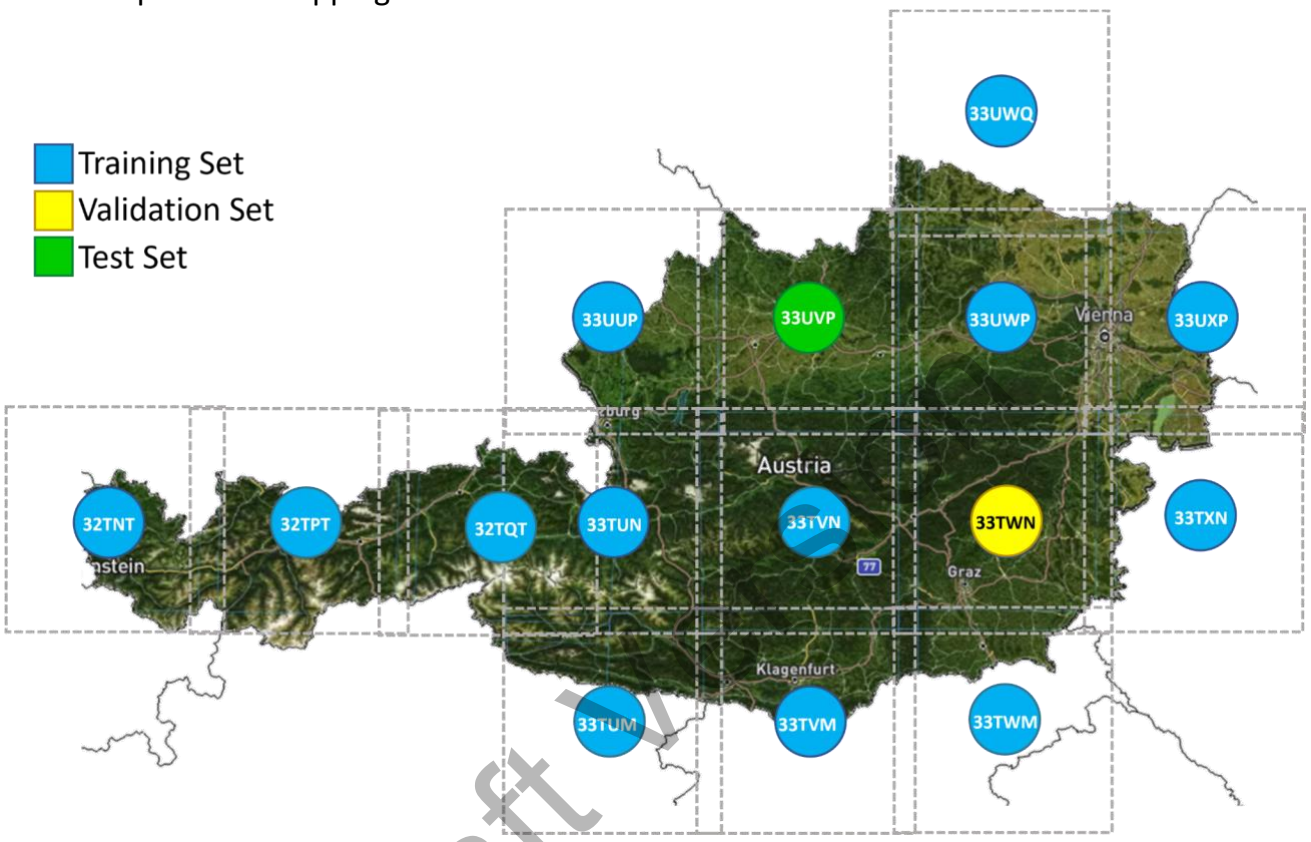

Fig 3.3. Selected tile to assess the performance of the LSTM network.

To further validate the results of the LSTM, we considered a second publicly available dataset, the Lucas database, and compared the results obtained over the Austrian country with the samples of the second dataset collected over 35 tiles. Fig. 3.3 shows the samples extracted from the Lucas Database that were used to validate the architecture.

Table 3.1 shows the numerical results obtained on the two dataset. From the table one can see that the value of Overall Accuracy (OA) and F score (F1) are similar in both datasets, showing that the architecture is able to generalise over large scale areas. The architecture performs well on the considered area that includes a large part of the Danube basin, with an OA% of 85.20% and a median FScore of 82.71%. Analysing the single classes, both the tests show a decrease of performance associated with the rye (confused with its wheat-rye hybrid triticale), and the permanent plantation class, which shows a high variability across the study area. Apart from the minoritarian classes, the results show that the different crop types are classified with good Overall Accuracy and Fscore. Moreover, as can be seen from the Lucas Database test considered, the network is able to accurately generalise the results obtained in the Austrian country to the neighbouring tiles without a huge decrease in terms of performance.

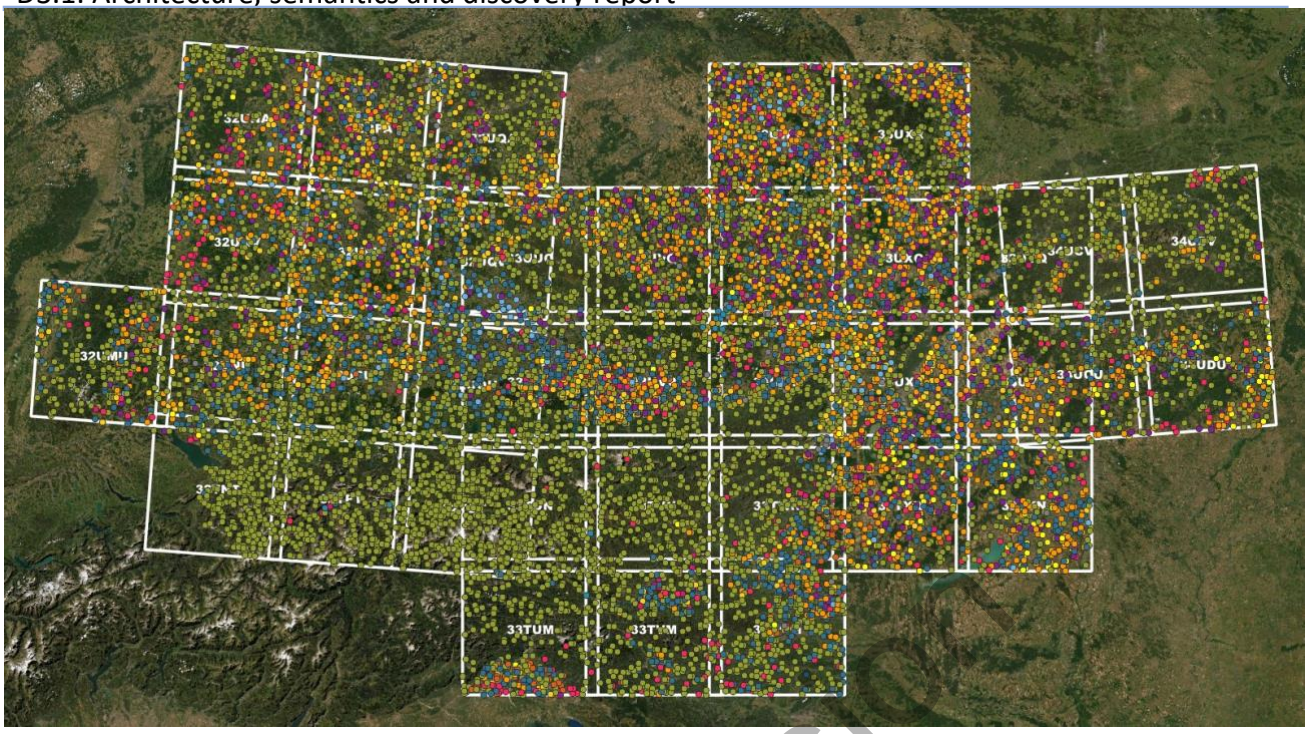

Fig 3.4. The samples extracted from the Lucas Database over 37 tiles in the Danube Basin area.

		TimeSen2Crop TestSet		LUCAS (Danube Basin)	
	Crop Type	#Samples	F1%	#Samples	F1%
	Legumes	2031	83.23	-	-
	Grassland	15080	82.99	2832	92.72
	Maize	15001	97.94	1661	95.94
	Potato	4015	84.93	88	72.20
	Sunflower	240	70.89	140	85.82
	Soy	10712	96.15	131	84.44
	Barley	15001	91.92	817	74.75
	Winter Caraway	577	42.95	-	-
	Rye	9701	73.96	142	40.38
	Rapeseed	5086	96.98	733	92.03
	Beet	4212	97.04	181	92.05

	Spring Cereals	11987	89.49	-	-
	Winter Wheat	15001	95.44	1705	80.98
	Triticale	14363	75.80	117	18.32
	Perm. Plantations	411	26.52	170	61.49
	OA%		85.39		85.20
	Median F1%		84.09		82.71

Table 3.1. Numerical results obtained considering the TimeSen2Crop test set (tile 33UVP) and the samples extracted from the Lucas Database (37 tiles in the Danube basin).

3.3 Deep Network for pixel-level classification of S2 patches

The pixel-level classification models are tested using the Sen12MS dataset [13]. This dataset is made of 180,662 Sentinel-2 satellite patches. Each of these patches is multi-spectral, composed of 13 channels corresponding to specific bandwidths. While several types of annotations are available in this dataset, we focus on the labels corresponding to the Geosphere-Biosphere Programme (IGBP), which in this dataset classify every pixel among one of the 17 available classes (Table 3.2). The SEN12MS research team proposes a simplified class clustering (Simplified Class Name) with 10 classes. We choose this data classification for our tests.

	IGBP class description	IGBP class id	Simplified id
	Evergreen Needleleaf Forest	1	1
	Evergreen Broadleaf Forest	2	
	Deciduous Needleleaf Forest	3	
	Deciduous Broadleaf Forest	4	
	Mixed Forest	5	
	Closed Shrublands	6	2
	Open Shrublands	7	
	Woody Savannas	8	3
	Savanna	9	

	Grasslands	10	4
	Permanent Wetlands	11	5
	Croplands	12	6
	Cropland / Natural Vegetation Mosaics	14	
	Urban and Built-up Lands	13	7
	Permanent Snow and Ice	15	8
	Barren	16	9
	Water Bodies	17	10

Table 3.2. Class description of the SEN12MS dataset

According to (Sulla-Menashe et al., 2019) the average accuracy of the IGBP global land cover map is around 67% [14]. This fixes a limitation in terms of pixel-segmentation performances, as our models have to learn some wrong features during the training process and cannot achieve fine-grained detection.

As described in the previous section, our model performs segmentation for each pixel of the input image. We split the Sen12MS dataset into training, validation and test sets. For each class that we want to detect, we create a new sub-dataset. Each of this dataset is balanced in order to have four times more images containing the targeted class than images that do not contain these annotations. This aims to tackle the imbalanced class distribution among the original dataset. Keeping some images with no annotations during the training process aims to reduce the amount of false positive results when the trained model is tested with real-world data.

To facilitate the evaluation process, one model per class is trained. Users can also choose to perform multi-label classification if required, knowing that it will impact segmentation performances. This can be interesting in embedded contexts where there are some constraints on data storage, training or inference time. To evaluate models, we choose to use the Precision, Recall and F1-score metrics.

$$Precision = TP^{TP + FP}$$

$$Recall = TP^{TP + FN}$$

$$F1\ Score = 2 * Precision\underline{Precision} + * \underline{Recall}Recall$$

With *TP* for True Positive, *FP* for False Positive, *TN* for True Negative and *FN* for False Negative. Precision measures the amount of good detection among detected areas, while recall measures the amount of good detections among all areas. For instance, in a situation where we want to segment trees from a background, if we only detect 1% of the trees but that all the segmented trees are effectively trees, we would have a very high precision score but a very low recall score. On the contrary, if all the trees are well classified but with most of the background also misclassified as trees, it would lead to a low precision score and a high recall score. The F1-score allows us to find a compromise between these two values and evaluate overall model performances.

During the preprocessing phase, we apply rotation and mirroring as data augmentation to avoid overfitting during the training process. We decide to apply the same training parameters for all the classes: We use all the 13 bands of the original data as input for our model. In a user-defined process, fine-grained selected hyperparameters may lead to an increase in terms of segmentation performances. Other parameters are described in Table 3.3. For each model, the output is thresholded in order to maximise the F1-score.

Selected bands	Patch size	Loss	Batch size	Learning rate
All	256x256	crossentropy	4	10e-4

Table 3.3. Parameters used during the Pixel-level classification service training

Simplified class id (Table 3.2)	Precision %	Recall %	F1-score %
1	55.09	63.66	53.83
2	32.92	31.86	31.65
3	38.37	44.78	32.90
4	32.03	52.99	32.60
5	34.07	57.49	36.75
6	57.39	61.54	54.91
7	51.42	59.81	50.90

8	5.794	74.39	9.93
9	24.50	23.21	21.47
10	64.22	69.61	61.58

Table 3.4. Numerical results from the training of the image segmentation models on the SEN12MS dataset

The variation of results highlights the imbalanced number of samples among classes and the disparity in terms of segmentation difficulty among classes (data variance). This reinforces the need of context-defined parameters. Discriminating features hide in specific bandwidths depending on the targeted label, and removing bandwidths that appear to be useless for some classes may decrease the data complexity and ease the convergence of the model. Hyperparameters used during training have also to be adjusted

Since the ground-truth provided by SEN12MS only reaches 67% of pixel accuracy compared to real world land covering, pixel-level metrics may not be the best way to evaluate these models. Indeed, even if areas are well classified from a high-level perspective, the lack of precision of the ground-truth highly penalizes those metrics which expect a complete overlay between predictions and labeled data to reach high scores. To get around this issue, one could consider some other metrics based on higher level consideration (for instance per patch classification). Another solution would be to switch to a different dataset that would provide fine-grained satellite annotations.

4 Conclusions

In this Deliverable, the machine learning models for EO developed in the WP5 (Bootstrapping AI4Copernicus with high-impact services) have been presented and described in detail. The architectures can be used by the open call winners of the four different domains (energy, agriculture, health, and security) to support their application scenario by significantly reducing the implementation time and improving the quality and the number of tests performed. In this context D4.3 presents a deep analysis of the architectures in the AI4Copernicus environment, allowing the user to fully understand the pipeline and workflow of the resources provided as bootstrapping services.

5 References

[1] Podsiadlo, I., Paris, C., & Bruzzone, L. (2020, September). A study of the robustness of the longshort-term memory classifier to cloudy time series of multispectral images. In Image and Signal Processing for Remote Sensing XXVI (Vol. 11533, p. 115331O). International Society for Optics and Photonics.

[2] Price, I., & Rasp, S. (2022). Increasing the accuracy and resolution of precipitation forecasts using deep generative models. arXiv. https://doi.org/10.48550/ARXIV.2203.12297

[3] Ravuri, S., Lenc, K., Willson, M. *et al.* Skilful precipitation nowcasting using deep generative models of radar. *Nature* **597,** 672–677 (2021). https://doi.org/10.1038/s41586-021-03854-z

[4] Harris, L., McRae, A. T. T., Chantry, M., Dueben, P. D., & Palmer, T. N. (2022). A Generative DeepLearning Approach to Stochastic Downscaling of Precipitation Forecasts. arXiv. https://doi.org/10.48550/ARXIV.2204.02028

[5] Mirza, M., & Osindero, S. (2014). Conditional Generative Adversarial Nets. arXiv.https://doi.org/10.48550/ARXIV.1411.1784

[6] J. Leinonen, D. Nerini and A. Berne, "Stochastic Super-Resolution for Downscaling Time-EvolvingAtmospheric Fields With a Generative Adversarial Network," in IEEE Transactions on Geoscience and Remote Sensing, vol. 59, no. 9, pp. 7211-7223, Sept. 2021, doi: 10.1109/TGRS.2020.3032790.

[7] Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., & Courville, A. (2017). Improved Training ofWasserstein GANs. arXiv. https://doi.org/10.48550/ARXIV.1704.00028

[8] Geoffrey, H., Nitish, S., & Kevin, S. (n.d.). Overview of mini- -batch gradient descent . In Lecture notes in Neural Networks for Machine Learning.

[9] G. Weikmann, C. Paris and L. Bruzzone, "TimeSen2Crop: A Million Labeled Samples Dataset ofSentinel 2 Image Time Series for Crop-Type Classification," in IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing, vol. 14, pp. 4699-4708, 2021, doi: 10.1109/JSTARS.2021.3073965.

[10] O. Ronneberger, P. Fisher, T. Brox. (2015). U-Net: Convolutional Networks for Biomedical ImageSegmentation. arXiv. https://doi.org/10.48550/arXiv.1505.04597

[10] Pacchiardi, L., Adewoyin, R., Dueben, P., & Dutta, R. (2021). Probabilistic Forecasting withGenerative Networks via Scoring Rule Minimization. arXiv. https://doi.org/10.48550/ARXIV.2112.08217

[11] K. He, H. Fan, Y. Wu, S. Xie, R. Girshick. (2019). Momentum Contrast for Unsupervised Visual Representation Learning. arXiv. https://doi.org/10.48550/arXiv.1911.05722

[12] Hoffmann, S., & Lessig, C. (2022). AtmoDist: Self-supervised Representation Learning forAtmospheric Dynamics. arXiv. https://doi.org/10.48550/ARXIV.2202.01897

[13] M. Schmitt, L. H. Hughes, C. Qiu, X. X. Zhu. (2019). SEN12MS -- A Curated Dataset ofGeoreferenced Multi-Spectral Sentinel-1/2 Imagery for Deep Learning and Data Fusion. arXiv. https://doi.org/10.48550/arXiv.1906.07789

[14] Sulla-Menashe, D., Gray, J. M., Abercrombie, S. P. and Friedl, M. A., 2019. Hierarchical mappingof annual global land cover 2001 to present: The MODIS Collection 6 Land Cover product. Remote Sens. Env. 222, pp. 183–194.