Towards a Massive Sentinel-2 LAI Time-Series Production Using 2-D Convolutional Networks

Mohanad Albughdadi, Guillaume Rieu, Sylvie Duthoit, Mohammed Alswaitti

Highlights

Towards a Massive Sentinel-2 LAI Time-Series Production Using 2-D Convolutional Networks

Mohanad Albughdadi,Guillaume Rieu,Sylvie Duthoit,Mohammed Alswaitti

- A fast procedure to estimate LAI products from Sentinel-2 images is proposed based on 2-D ConvNets.
- The model confidence interval is introduced as a quality indicator for LAI estimates.
- The proposed method is adapted to massive production of LAI using Sentinel-2 images.

Towards a Massive Sentinel-2 LAI Time-Series Production Using 2-D Convolutional Networks

Mohanad Albughdadi[a], Guillaume Rieu[a], Sylvie Duthoit[a] and Mohammed Alswaitti[b,*]

[a]TerraNIS SAS., 12 avenue de l'europe 31520 Ramonville-Saint-Agne,France.

[b]School of Information Science and Technology, Xiamen University Malaysia, Jalan Sunsuria, Bandar Sunsuria, 43900 Sepang Selangor Darul Ehsan, Malaysia.

ARTICLE INFO

Keywords:
biophysical parameter
LAI
CNN
Sentinel-2

ABSTRACT

Biophysical parameters and more specifically the leaf area index provide an absolute quantification of the biomass of vegetation allowing an overview of the development status of a plant. However, the estimation of the leaf area index requires sophisticated and complex algorithms. This paper proposes a new procedure to estimate the leaf area index using Sentinel-2 data. The proposed procedure relies on the 2-D convolutional network known as the UNet algorithm for regression. The architecture of the UNet algorithm is adapted to account for the processing of large chunks of Sentinel-2 data. Moreover, the adopted procedure makes use of the dropout as a Bayesian approximation at the inference step in order to allow estimating the algorithm confidence interval, which is a very important quality indicator for the production of biophysical parameters. The proposed procedure is validated on multiple Sentinel-2 tiles and years and compared to the multilayer perceptron algorithm and the Sentinel Application Platform of the European Space Agency, also known as SNAP. The UNet and multilayer perceptron algorithms provide coherent results when compared to the results obtained using the SNAP software with an average correlation of 0.99 for both algorithms. However, the UNet algorithm provides better results in terms of average Euclidean distance, mean squared error and R^2 score. One main advantage of the UNet algorithm is the vast reduction of inference time when compared to the SNAP software and the multilayer perceptron

regressor. The estimation of the leaf area index of a Sentinel-2 tile at 20 m requires 18 seconds, 13.5 minutes and 15 minutes using the UNet, multilayer perceptron and SNAP, respectively. This advantage allows a massive production of temporal sequences of leaf area index based on Sentinel-2 images that will be ready to use for land cover/use applications. Furthermore, experiments conducted on multiple crop types prove that the proposed approach can serve as a generic procedure to estimate the leaf area index regardless of the crop type.

1. Introduction

Biophysical parameters derived from remote sensing images have well defined signature dependent on the growth stage of a crop. Examples of these parameters are the fraction of green vegetation cover (fCover), the fraction of absorbed photosynthetically active radiation (FAPAR), the chlorophyll content (CHL) and leaf area index (LAI). The characteristics of these biophysical parameters have made them of great interest for many agricultural and land use applications such as crop growth monitoring (Albughdadi et al., 2017) and crop/land cover classification (Waldner et al., 2015). Furthermore, the fact that the biophysical parameters are not sensor-specific allows increasing the satellite observation frequency by using data extracted from different sensors (Waldner et al., 2015). The free access to Sentinel-2 (S2) images, characterized by a spectral richness and a fine temporal and spatial resolution has fostered the development of image processing applications, in particular those related to crop development. Nonetheless, this vast stream of data requires the existence of efficient, fast and accurate techniques to estimate the corresponding biophysical parameters.

LAI is a biophysical parameter that measures the total area of leaves per unit ground area and is directly correlated with the amount of intercepted light by the plant. This parameter has many uses

*Corresponding author
mohanad.albughdadi@terranis.fr (M. Albughdadi); guillaume.rieu@terranis.fr (G. Rieu); sylvie.duthoit@terranis.fr (S. Duthoit); alswaitti.mohammed@xmu.edu.my (M. Alswaitti)
ORCID(s):

such as the prediction of photosynthetic primary production, monitoring crop growth and yield estimation (Waldner et al., 2019). Moreover, the LAI is required by many global models of climate (Bonan et al., 2002), ecosystem productivity and ecology (Asner et al., 2003; Running and Coughlan, 1988; Sellers et al., 1997; Yan et al., 2016). Hence, this paper focuses on the retrieval of the LAI biophysical parameter using an efficient approach that allows fast estimation of this parameter at the level of S2 tile.

Different approaches have been studied to estimate biophysical parameters in general and the LAI in particular. One can divide them into three main approaches, namely, physical, parametric and non-parametric machine learning techniques (Verrelst et al., 2015).

The physical methods apply physical laws and specific knowledge to infer model variables. A well-known example of these models is the inversion of radiative transfer models (RTMs) (Knyazikhin et al., 1998; Weiss et al., 2000). The estimation of the biophysical parameters is then reduced to an ill-posed inverse problem which is non-trivial to be solved. Two popular approaches have been used to solve this inverse problem. The first approach relies on lookup-tables (LUT) inversion strategies. These strategies require the simulation of spectral reflectance for a large range of RTM variable values. Hence, the inversion problem is transformed into searching the set of simulated reflectance set for the value that most resembles the measured one (Liang, 2007). Additional challenges might be faced when using LUT-based physical models such as selecting the cost function to perform LUT queries. A thorough comparison was performed in (Verrelst et al., 2015) between the different methods in terms of accuracy, computational time and the ability to provide model uncertainty. This study concluded that physical-based models relying on LUT are cumbersome in terms of computational time. The second approach to perform the inversion of the RTMs is through using neural networks. Contrary to the LUT approach, the simulated database is used only once during the training phase where the weights of the network are estimated (Bacour et al., 2006). Indeed, neural networks are computationally efficient algorithms that aim at approximating the non-linear relations between input and output variables. It

is worth mentioning that the performance of a neural network-based approach relies on the quality of the simulated database and their architecture. In this context, the SNAP software (Weiss and Baret) of the European Space Agency includes operational modules that estimate biophysical parameters such as the LAI, FAPAR and fCover from S2 images. The training of these neural networks is performed offline and the operational models are only integrated to the SNAP software. Although the neural networks are used to generate these modules, they work at the pixel-level, *i.e.,* the estimation of the biophysical parameters is performed for each pixel separately, which makes the production of periodic biophysical products a time consuming process.

The parametric methods aim at explicitly parameterizing the relationship between spectral bands and the biophysical parameters (Glenn et al., 2008). This is mainly performed using vegetation indices regressed with the biophysical parameters using a regression function. An extensive review on the selection of spectral bands and vegetation indices is provided in (Le Maire et al., 2004). This kind of methods has been the choice in remote sensing community since it is very fast. However, the performance of such methods is influenced by the choice of spectral bands, vegetation indices and the fitting function (Rivera et al., 2014; Verrelst et al., 2015).

Finally, the non-parametric methods make use of machine learning regression techniques to establish a mapping function from input variables represented in the spectral bands and the output biophysical parameters through a training phase (Verrelst et al., 2011, 2012). These models use the full spectral bands to infer the non-linear relationship between these bands and the biophysical parameters. The comparison provided in (Verrelst et al., 2015) found that these models are effective in terms of accuracy, inference time and the ability of some of them (those based on Bayesian inference) to provide the model uncertainty. These models are very adapted to operational production since they are also trained offline on ground truth measurements and then used for production. However, their performance is affected by the scarcity of ground measurement and the fact that they also work at the pixel-level, which means they are not very adapted to large scale production.

The need of the scientific and industrial communities to have access to temporal LAI products has been first answered with the MODerate resolution Imaging Spectroradiometer (MODIS) instruments (Yan et al., 2016; Yang et al., 2006). The latest collection (C6) contains the LAI products from February 2000 to this day at a spatial resolution of 500m (Myneni et al.). Additionally, the LAI is retrieved using Terra MODIS, Aqua MODIS and Terra MODIS+Aqua MODIS allowing a high temporal frequency of 4 and 8 days. In this context the S2 mission comprises a constellation of two polar-orbiting satellites over land and costal water with a high visit frequency. Additionally, the spectral and spatial resolutions of S2 images are higher than those associated with MODIS data. These advantages of S2 images can be exploited to generate temporal LAI products similar to the MODIS ones that cover large parts of the globe and will be ready to use for land cover applications.

The S2 toolbox in the SNAP software provides many features such as atmospheric correction and the biophysical parameter processor. This biophysical parameter processor was tested and validated in many studies (Brown et al., 2019; Djamai et al., 2019; Vinué et al., 2018; Weiss and Baret) using ground truth measurements of biophysical parameters. These studies proved that the biophysical parameter processor in the SNAP software is able to provide comparable results with ground truth measurements and can serve as a generic model for biophysical parameter estimation without any calibration using the crop type. Nonetheless, estimating LAI products using the SNAP software on a large scale is not practical as the biophysical parameter processor takes around 15 to 20 minutes to estimate the LAI of one S2 tile at 20m resolution. The bottleneck of the procedure adopted in the SNAP software is that it relies on pixel-wise processing to estimate the biophysical variables, which limits its performance. One solution to this performance challenge is to adopt a strategy that allows processing large amount of data in a very short time. Recent advances in machine learning and more specifically deep learning have fostered computer vision applications. Deep learning techniques allow machines to understand images, extract pertinent features and hence perform sophisticated computer

vision tasks such as semantic classification, object detection and regression among others.

The first contribution of this paper is to provide a scalable, fast and accurate estimation of LAI products at the S2 tile-level using 2-D convolutional neural networks. The use of these networks speeds up the processing of a whole tile to multiple seconds. Since the biophysical processor in the SNAP software has been validated multiple times and it provides a good performance in estimating the LAI biophysical parameter (Brown et al., 2019; Djamai et al., 2019; Vinué et al., 2018; Weiss and Baret), LAI products derived from the SNAP software along with the corresponding S2 images are used at the training phase (which is performed offline). Once the model is trained, it is used at the operational mode to provide the estimates. It is worth noting that solutions to problems faced by 2-D convolutional networks are adopted in the proposed structure such as the border problem and overfitting. The second contribution of this work is the proposition of using dropout as a Bayesian approximation, which allows providing a confidence interval representing the model uncertainty. This an important aspect in the estimation of biophysical parameters as it is considered as a quality indicator. Additionally, as in the procedure adopted in the SNAP software, quality indicators from the acquired image can also be propagated to the final product (Weiss and Baret).

The rest of the paper is organized as follows. Section 2 describes the training and testing datasets used through this paper. The competing algorithms including the adopted 2-D convolutional networks are illustrated in Section 3. The adopted strategy of generating the training datasets, training the algorithms and testing them for operational uses are discussed in Section 4. Results and discussions are depicted in Section 5. Finally, some conclusions and future work are presented in Section 6.

2. Datasets

This paper studies the estimation of LAI biophysical parameter using S2 Copernicus data. The S2 mission aims at acquiring high resolution optical images (10 m to 60 m) over land and coastal waters. This mission consists of two constellation satellites (Sentinel-2A and Sentinel-2B), hence allowing a high temporal resolution of the acquired sites. S2 data are characterized by a fine spectral resolution as the acquired multispectral images consist of 13 bands in the visible, near infrared (NIR) and short wave infrared (SWIR) part of spectrum. Table 1 summarizes the different characteristics of each of these bands.

Table 1
S2 band characterstics. CWL and BW denote the central wavelength and bandwidth, respectively and are expressed in nm.

Band	Sentinel-2A		Sentinel-2B		Resolution
	CWL	BW	CWL	BW	
Band 1-Coastal aerosol	442.7	21	442.2	21	60
Band 2-Blue	492.4	66	492.1	66	10
Band 3-Green	559.8	36	559.0	36	10
Band 4-Red	664.6	31	664.9	31	10
Band 5-Vegetation red edge	704.1	15	703.8	16	20
Band 6-Vegetation red edge	740.5	15	739.1	15	20
Band 7-Vegetation red edge	782.8	20	779.7	20	20
Band 8-NIR	832.8	106	832.9	106	10
Band 8A-Narrow NIR	864.7	21	864.0	22	20
Band 9-Water vapour	945.1	20	943.2	21	60
Band 10-SWIR-Cirrus	1373.5	31	1376.9	30	60
Band 11-SWIR	1613.7	91	1610.4	94	20
Band 12-SWIR	2202.4	175	2185.7	185	20

2.1. Training Data

Multi-temporal sequences of S2 satellite images were used to train the algorithms in this study. More specifically, 14, 5 and 5 images of T30TYS, T30TYQ and T31TCJ S2 tiles, respectively, were used resulting in a dataset of 24 S2 images. These images were acquired in the period between October 2016 and November 2017 and characterized by a low cloud cover. The dates of image acquisition associated with each tile are summarized in Table 2. Moreover, the training S2 tiles are depicted in red in Fig. 1. The selected S2 tiles contain a variety of crops, which allows the algorithm to capture different vegetation characteristics. For instance, the dominant vegetation surfaces in the T30TYS are

Table 2
Acqusition dates of the training dataset.

T30TYS	T30TYQ	T31TCJ
11th October 2016	6th April 2017	16th May 2017
30th November 2016	26th May 2017	26th May 2017
10th December 2016	5th July 2017	5th July 2017
19th January 2017	14th August 2017	14th August 2017
18th February 2017	22nd November 2017	24th August 2017
27th March 2017		
30th March 2017		
9th April 2017		
19th April 2017		
29th April 2017		
18th June 2017		
18th July 2017		
7th August 2017		
27th August 2017		

associated with grassland with around 28% of the parcels. The tile also contains varieties of wheat, corn, sunflower, rapeseed, barley and fallow, which represent around 50% of the agricultural parcels in this tile. The geographical zone covered by the T30TYQ S2 tile is known for its vineyards, where they occupy around 17% of the agricultural surfaces. The T30TYQ tile also contains multiple crop types such as wheat, corn, sunflowers, grassland and fallow. Finally, the T31TCJ tile contains similar crop types such as wheat, barley, soy beans, sunflower, fruits and vegetables. For all these images, the pixel-wise LAI biophsyical parameters were estimated using the SNAP software in order to train the regression algorithms. Note that the parameter extraction for the training dataset is explained in detail in Section 4.2.

2.2. Testing Data

Images associated with multiple S2 tiles (T31TCK, T30TYM and T30TYT) were used test the capability of the competing models in estimating the pixel-wise LAI values (depicted in blue in Fig. 1). The selection of the images insures the variability of the acquisition period as it covers multiple months and years. Table 3 summarizes the acquisition dates of the testing images. Note that the LAI estimates derived from the SNAP software (see Section 4.2) were considered as ground truth data in order to be compared to the LAI estimations obtained using the competing regression models.

Table 3
Acqusition dates of the testing dataset.

T31TCK	T30TYM	T30TYT
16th May 2017	6th April 2017	14th January 2018
26th May 2017	16th April 2017	20th March 2018
25th June 2017	6th May 2017	19th April 2018
5th July 2017	26th May 2017	24th April 2018
4th August 2017	15th June 2017	23rd June 2018
14th August 2017	5th July 2017	18th July 2018
	14th August 2017	2nd August 2018
	13th September 2017	22nd August 2018
		21st October 2018
		20th December 2018

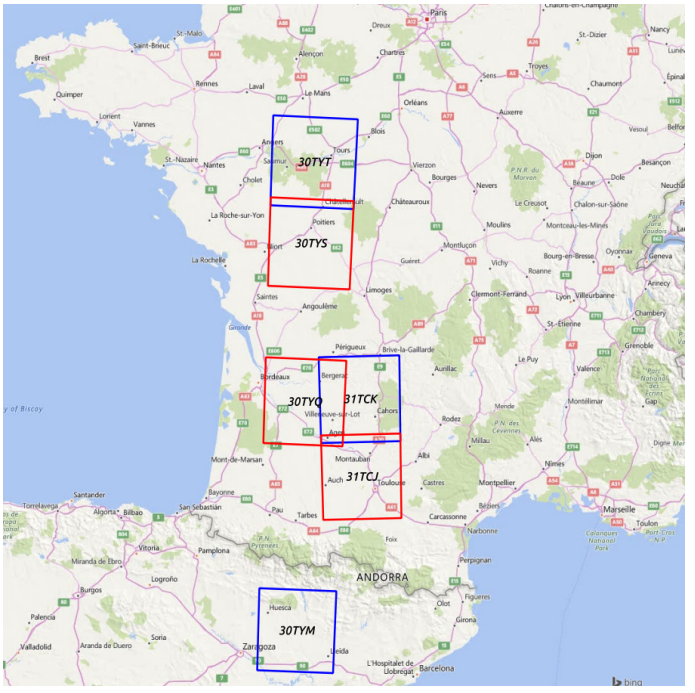


Figure 1: The training (red) and testing (blue) S2 tiles used in the conducted experiments.

3. Network Architecture

3.1. Convolutional Neural Network Regression

The convolutional neural network (CNN) presented in this paper adopts the UNet architecture introduced in (Ronneberger et al., 2015) for biomedical image segmentation. The architecture of this network allows working with few training samples and yielding more precise results in a classification task. The UNet architecture uses a contracting network followed by successive layers that replace

pooling operators with upsampling ones with a large number of feature maps that allow enhancing the output resolution and propagating context information to higher resolution layers. The localization is improved by concatenating high resolution features from the contracting path with upsampled feature maps. Fig. 2 shows the adopted UNet architecture used for the LAI regression problem with some modifications introduced to adapt to the LAI estimation problem. Similar to the original UNet architecture, the left and the right sides represent the contracting and expansive paths, respectively. On the one hand, the contracting path consists of multiple contraction blocks. Each block comprises two 3×3 convolution layers that use a rectified linear activation (ReLU) followed by a 2×2 max pooling. In this modified network, a batch normalization layer is added after each convolutional layer in order to speed-up the training process and add some regularization to the network. After each block, a spatial dropout layer is added in order to avoid overfitting and allow estimating the model uncertainty (see Section 4.5.1). Each of the aforementioned blocks double the number of feature maps and reduces the $x - y$ size by half. The resulting architecture of this contracting path allows the network to learn complex structures. On the other hand, the expansive path consists of multiple expansion blocks. Each block consists of two 3×3 convolutional layers that uses a ReLU activation followed by 2×2 upsampling layer. Similar to the contracting path, a batch normalization layer after each convolutional layer and a spatial dropout layer after each block are added. After each block, the number of feature maps reduces to the half and the $x - y$ size doubles. Additionally, a concatenation step is applied with the corresponding feature maps in the contracting path. This concatenation procedure insures that the features learned during the contracting path are used to reconstruct the image. The last hidden layer is cropped to avoid the local boundary effect of the CNN that perturbs predictions at the image border. Finally, a 1×1 convolution is used to map the 32-channel feature vector to one-channel output that corresponds to the estimated LAI values of the input image. The output layer uses a linear activation function for this regression problem.

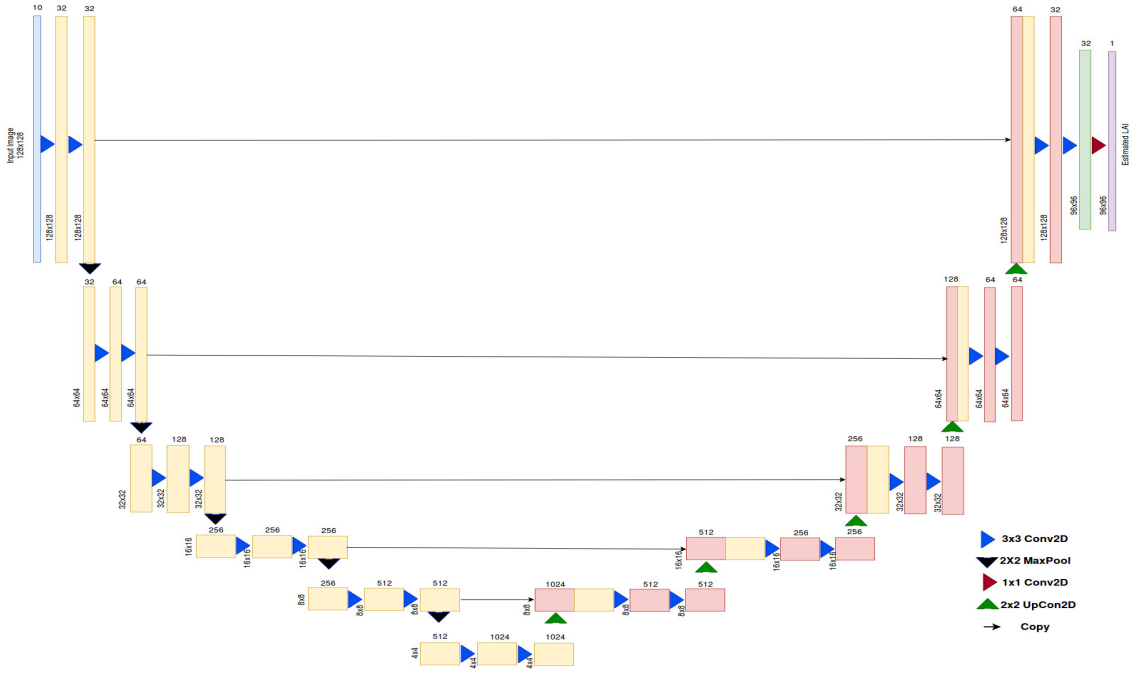


Figure 2: The UNet architecture used in this paper to estimate pixel-wise LAI values. Boxes represent multichannel feature maps. The number of feature maps is provided at the top of each box while the $x-y$ dimensions are provided at the lower left side of each box.The right blue, down black, right red, green up and gray arrows represent 3×3 conv2D, maxpool2D, 1×1 conv2D, upsampling2D and copy operations, repecively.

3.2. Multilayer Perceptron Regression

Multilayer perceptron (MLP) networks (Rumelhart et al., 1986; Werbos, 1974) are the simplest kind of feed-forward networks that consist of neuron-like processing units. These units are arranged into a set of layers such that each layer contains a certain number of identical neurons. Each neuron in the layers is an input to every neuron in the proceeding layer, *i.e.,* fully connected network. In MLP networks, the first layer is the input one where the input features are fed to the neural network. Hence, this layer has the size of the input features. The last layer of the MLP networks is the output layer where each output value is associated with one neuron. In the case of regression problems, the output layer has one neuron. The layers between the input and output layers are known as the hidden layers. The depth of an MLP network is defined by the number of layers in the network. Given a set of features $x = x_1, ..., x_b$ and an output y, the MLP networks approximates a function $f(.) : R^b \rightarrow R^o$ using x and y, where b is the number of input features and o is the output dimension. Neurons in the hidden

layers transform the values from the previous layer with a weighted linear summation $\omega_1 x_1 + ... + \omega_b x_b$ followed by a non-linear activation function $g(.)R \rightarrow R$. Finally, the output layer receives the values of the last hidden layer and transforms them into output values. The MLP network used in this paper consists of 5 layers, namely, the input layer, three hidden dense layers and the output layer. The input layer consists of 10 units that correspond to the dimension of the input feature vector (see Section 4.4.2). The first, second and third hidden layers consist of 256, 128 and 64 units, respectively with a ReLU activation for the first two hidden layers and a linear activation for the last one. A dropout is also used in the MLP network after each hidden layer to avoid overfitting. The output layer maps a feature vector of size 64 to a feature vector of size 1, which corresponds to the estimated LAI value.

4. Adopted Strategy

The adopted strategy is a five-step procedure that includes all the steps required to reproduce this work. These steps are detailed in what follows.

4.1. Data Preprocessing

S2 level-2A products were downloaded using the Peps (d'Etudes Spatiales , CNES) platform of the French National Center of Space Studies (CNES). The L2A products provide Bottom of Atmosphere (BOA) reflectance images derived from the associated Level-1C products. These products were then resampled to a 20 m resolution to be further processed. To be more specific, bands 2, 3, 4 and 8 were downsampled using the mean method while bands 1, 9, and 10 were upsamlped using the bilinear interpolation method. Bands 5, 6, 7 and 8A kept the same resolution (20 m). The cloud and shadow masks provided with the L2A products were used to mitigate cloud and shadow pixels.

4.2. Pixel-wise LAI Estimation Using SNAP

The SNAP software uses neural networks to estimate multiple biophysical variables (Weiss and Baret). These networks are trained using simulated data generated by radiative transfer models. Once these models are trained, they are provided in the operational mode to compute LAI estimates and

some quality assessment indicators. These indicators include the consistency of the input reflectance values with those used in the training dataset, the consistency of the outputs generated using the neural networks with those biophysical parameters in the training datasets and quality indicators that include information regarding the reliability of the atmospheric correction and cloud filtering. The latter quality indicators are replicated from the input images.

The SNAP software was used to generate pixel-wise LAI estimates associated with the S2 tiles described in Section 2.1. The resulting LAI products and their corresponding S2 images and were then used to train and test the competing algorithms. Note that the cloud and shadow masks of the S2 images were used to mask invalid pixels in the obtained LAI products before further processing.

4.3. Training Data Preparation

Temporal sequences of 3 S2 tiles, namely, T30TYS, T30TYQ and T31TCJ and their corresponding LAI products were used to generate the training samples (See Table 2). At this stage, bands 1, 9 and 10 of each image were discarded and only bands with a potential of giving information on vegetation were used. In a second step, the 2nd and 98th percentile pixel values of each band in each image were calculated resulting in a set of pseudo-minimum and pseudo-maximum for each band. Using the percentile instead of the minimum and maximum allows to discard pixels with erroneous values after the atmospheric correction. The minimum and maximum values of these sets were then used to normalize the training images between 0 and 1 using the following equation

$$x^l_{i,j} = \frac{x^l_{i,j} - \min_l}{\max_l - \min_l}, \tag{1}$$

where a pixel in the i-th row and j-th column of band l is denoted as $x^l_{i,j}$, $\max_l$, $\min_l$ denote the minimum and maximum values, respectively and $l = \{1, \ldots, b\}$ where b is the number of bands. Once the images were normalized, patches of size 128×128 were cropped of the S2 images and the corresponding SNAP LAI estimates. It is worth noting that when estimating the LAI using SNAP, a

flag raster is also generated that indicates the validity of the pixel-wise LAI estimates. This flag raster was used to discard erroneous SNAP estimates from the training dataset.

The aforementioned steps resulted in a dataset of size $d \times m \times n \times b = 17554 \times 128 \times 128 \times 10$ S2 patches and their corresponding LAI estimates of size $17554 \times 128 \times 128 \times 1$. This dataset was then used to train the two algorithms described in Section 3.

4.4. Training Procedure

The UNet and MLP algorithms were trained on a machine with Ubuntu 16.04 LTS OS, 32 GB of RAM, Intel Xeon W-2123 CPU with 8 processors that clock at 3.60GHz and a GeForce RTX 2080 GPU associated with 8 GB of memory. One can notice that the training of the UNet algorithm was much faster than the training of the MLP regressor on the same dataset. Additionally, the conducted experiments in Section 5 show the huge gain of performance when using the UNet algorithm to estimate the pixel-wise LAI values.

4.4.1. UNet training

The UNet algorithm is a 2D convolutional network which means that its input is a 4-D dataset organized as $d \times m \times n \times b$. The dataset described in Section 4.3 was divided into two subsets, *i.e.,* 75% and 25% of the total number of patches were used for training and validation, respectively. The training dataset described in Section 4.3 was fed to the algorithm as an input and the corresponding LAI dataset was used as an output. Nonetheless, a cropping step to the LAI dataset was added in order to adapt for the cropping layer added to the UNet architecture. This cropping step discards 16 pixels from the borders of the patches resulting in an LAI patch of size $96 \times 96 \times 1$. The RMSprop optimization algorithm proposed in (Hinton et al.) with a learning rate of 0.001 was used to estimate the parameters of this algorithm. The batch size was set to 64 and the maximum number of epochs to 500. Additionally, the dropout ratio was set to 0.5. Since the estimation of the LAI is a regression problem to estimate a continuous variable, the mean square error (MSE) was used as a loss function.

Fig. 3 depicts the training and validation losses of the UNet algorithm for 100 epochs. It is clear that both the training and validation losses stabilize around 40 epochs. The model used on the test data is the one that minimized the validation loss.

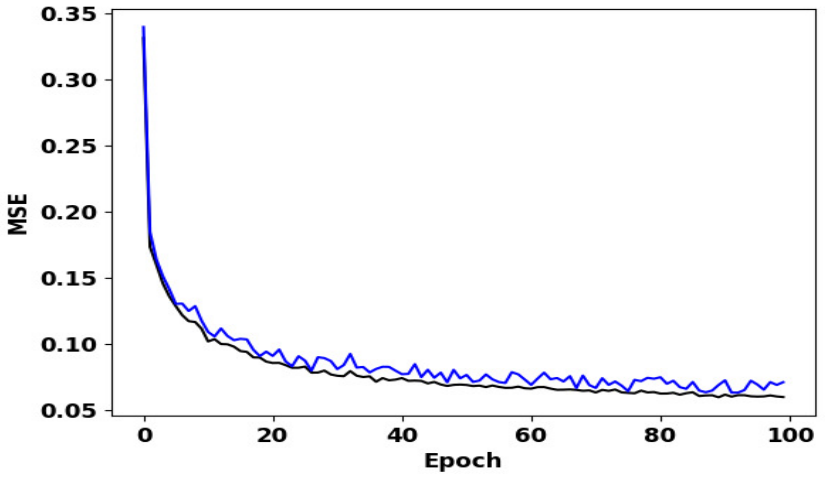


Figure 3: Training and validation loss of the UNet algorithm for 100 epochs.

4.4.2. MLP training

On the contrary of 2-D convolutional networks that are well adapted for image data, the MLP regressor is a generic algorithm that works with all kinds of data and does not consider the spatial properties of an image. Hence, the dataset described in Section 4.3 was flattened such that the size of the dataset is $p \times b = 287604736 \times 10$ where p and b denote the number of pixels and bands, respectively. Similar to the training of the UNet algorithm, 75% of the pixels were considered for training and 25% of them for validation. The RMSprop optimization algorithm was also used to estimate the model parameters with a learning rate of 0.001. The dropout was set to 0.5 and the MSE was used as a loss function. The number of epochs was set to 500. Fig. 4 shows the training and validation loss of the MLP regressor for 100 epochs. Note that the training and validation losses stabilize around 40 epochs.

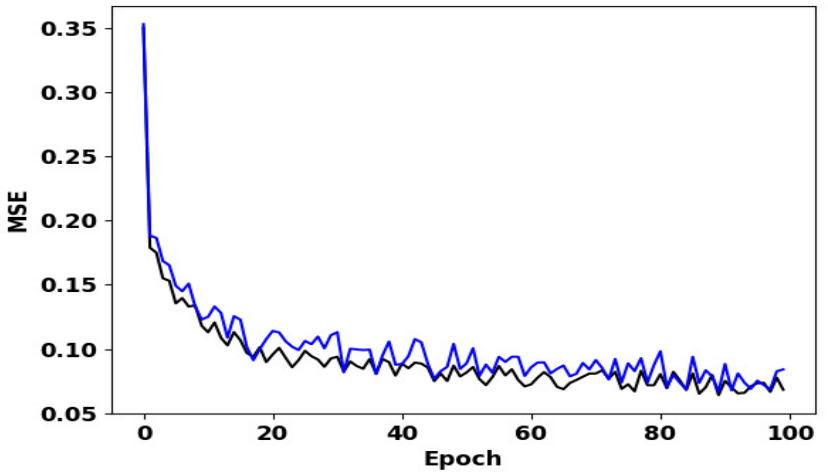


Figure 4: Training and validation loss of the MLP algorithm for 100 epochs.

4.5. Prediction Procedure

4.5.1. UNet prediction

The prediction procedure using the UNet algorithm took the full S2 image as an input. The input image was then normalized using the minimum and maximum values of each band derived from the training dataset. The normalized image was cut to patches of size $96 \times 96 \times 10$, and a padding strategy was adopted by adding a mirror reflection to the borders of the patches resulting in a patch size of $128 \times 128 \times 10$. The performed padding allowed adapting to the added 2-D cropping layer in the architecture of the UNet. The prediction was then performed on $d \times 128 \times 128 \times 10$ dataset, where d is the number of patches in an S2 image. After the inference step, the cutting and padding processes were inversed in order to construct the full LAI product of the input S2 image. This strategy allows a fast estimation of LAI values for S2 images in terms of seconds (see the performance analysis in Section 5). It is worth noting that the dropout was enabled during the inference step. Indeed, deep neural networks with dropout cast as approximate Bayesian inference in deep Gaussian processes (Gal and Ghahramani, 2016; Kwon et al., 2018). This allows capturing model uncertainty when performing classification and regression tasks using deep neural networks. When dropout is allowed at the inference step, randomly

selected neurons in the architecture of the UNet do not fire, hence Monte Carlo simulations can be performed. The mean of the simulations was then computed along with the standard deviation in order to compute the confidence interval of the algorithm. Denoting the number of Monte Carlo simulations as T, the mean of these runs as $\bar{y}$, the standard deviation as σ, a 95% confidence interval can be constructed using

$$\bar{y} \pm t\frac{\sigma}{\sqrt{T}} \tag{2}$$

where t is derived from the T-Distribution table. Knowing that $T = 30$, the degrees of freedom $df = 30 - 1$ and $\alpha = 1 - 0.95 = 0.05$, then $t = 1.699$.

4.5.2. MLP prediction

The prediction procedure of the MLP regression algorithm consisted of flattening the S2 images into a 2-D array of size $p \times b$. The 2-D array was then normalized using the minimum and maximum values of the training dataset. After the inference, the output LAI values were then reshaped to the original size of the S2 image. This procedure requires longer time when compared to the prediction procedure of the modified UNet algorithm as it requires flattening the input image and hence working on image pixels instead of patches (see Section 5). It should be pointed out that the model uncertainty can also be captured using the MLP algorithm and the strategy adopted in the UNet algorithm (Section 4.5.1). However, this would require a very long time.

5. Results and Discussion

The competing algorithms were evaluated on real S2 images using different evaluation metrics. These metrics as well as the experiments are described and discussed in what follows.

5.1. Evaluation Metrics

- **Pearson Correlation Coefficient** is a measure of the linear correlation between two variables (Benesty et al., 2009). This coefficient has a value between +1 and −1. On the one hand, a coefficient of 1 indicates a total positive linear correlation. On the other, a coefficient of −1 indicates a total negative correlation. A coefficient of 0 implies no linear correlation between the two variables. Denoting the estimated and SNAP LAI vectors as $\boldsymbol{y}^{\text{es}}$ and $\boldsymbol{y}^{\text{snap}}$, respectively, the Pearson correlation r coefficient can be estimated using

$$r = \frac{\sum_c \left(y_c^{\text{snap}} - \bar{y}^{\text{snap}}\right)\left(y_c^{\text{es}} - \bar{y}^{\text{es}}\right)}{\sqrt{\sum_c \left(y_c^{\text{snap}} - \bar{y}^{\text{snap}}\right)^2}\sqrt{\sum_c \left(y_c^{\text{es}} - \bar{y}^{\text{es}}\right)^2}} \tag{3}$$

where $\bar{y}_c^{\text{snap}}$ and $\bar{y}_c^{\text{es}}$ are the mean values of $\boldsymbol{y}_{\text{snap}}$ and $\boldsymbol{y}_{\text{es}}$, respectively and $c \in [1, p]$ is an index that runs over p pixels in a flattened image array.

- **Euclidean Distance** is a similarity measure of two variables (Danielsson, 1980). Assuming that each pixel estimation is a point vector, the Euclidean distance $Dist_p$ is calculated for two corresponding pixel values using

$$Dist_c = \sqrt{\left(y_c^{\text{snap}} - y_c^{\text{es}}\right)^2}. \tag{4}$$

The mean of these distances is then calculated to estimate a similarity measure between LAI estimates using SNAP and one of the competing algorithms, *i.e.,*

$$AvgDist = \frac{1}{p}\sum_{c=1}^{p}\sqrt{\left(y_c^{\text{snap}} - y_c^{\text{es}}\right)^2}. \tag{5}$$

- **Mean Squared Error** (MSE) evaluates the quality of an estimator by measuring the average of squared difference between the estimated values y_c^{es} and the actual ones y_c^{snap} for $c \in [1, p]$ and

p is the total number of pixels in an image using

$$MSE = \frac{1}{p}\sum_{c=1}^{p}\left(y_c^{\text{snap}} - y_c^{es}\right) \tag{6}$$

- **Determination Coefficient** (R^2) measures how observed variables are replicated by the model using the proportion of total variation of variables explained by the model. This score takes a value in the range $[0, 1]$, where it equals 1 when the model predictions perfectly fit the data. Using the same notion as before and knowing that the mean of the LAI estimates using SNAP is $\bar{y}^{\text{snap}}$, the variability of the dataset can be measured using

$$R^2 = 1 - \frac{SS_{\text{res}}}{SS_{\text{tot}}}, \tag{7}$$

where

$$SS_{\text{res}} = \sum_{c=1}^{p}\left(y_c^{\text{snap}} - y_c^{\text{es}}\right)^2, \tag{8}$$

and

$$SS_{\text{tot}} = \sum_{c=1}^{p}\left(y_c^{\text{snap}} - \bar{y}^{\text{snap}}\right)^2. \tag{9}$$

5.2. Experiments

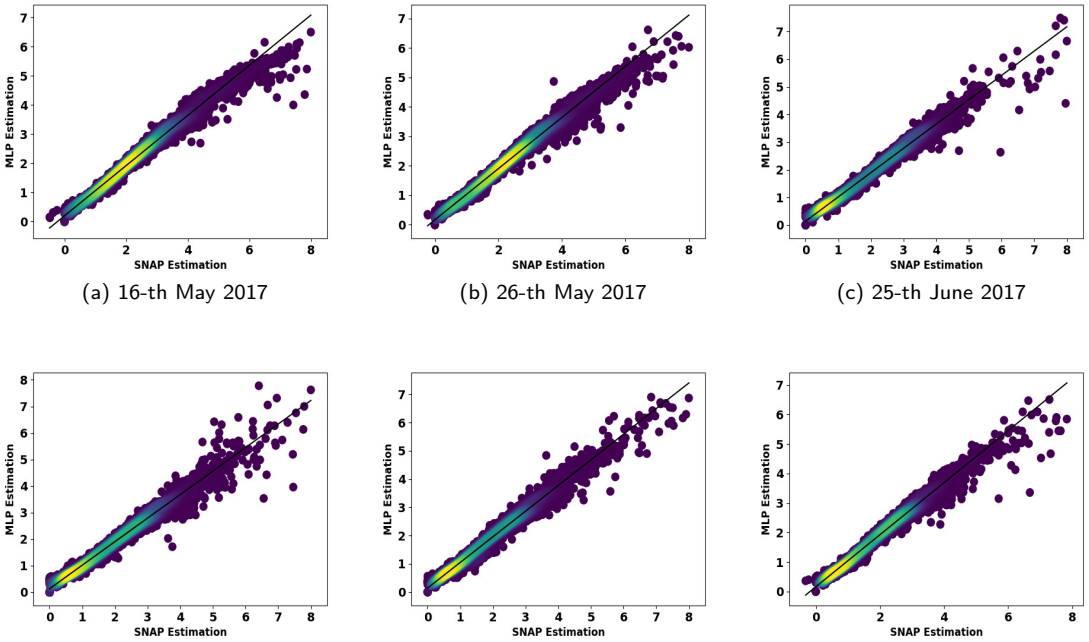
This section consists of four main experiments. The first one aims at comparing the UNet and MLP algorithms to the SNAP software. The second experiment further investigates the performance of the UNet algorithm compared to SNAP using more datasets acquired on different dates and geographical coverage. Additionally, the experiment demonstrates the performance of the UNet algorithm on different crop types. Then, the third experiment compares the LAI estimates obtained using the UNet

algorithm to ground truth data obtained using field measurements. Finally, the last experiment demonstrates the importance of considering the spatial characteristics of the image when estimating the LAI provides some insights on the feature extracted using the UNet algorithm.

5.2.1. UNet and MLP Compared to SNAP

A first analysis was conducted by comparing the LAI estimates obtained using the MLP and UNet regressors to those obtained using the SNAP software on a temporal series of images associated with the T31TCK S2 tile (see Section 2.2). Firstly, 20000 randomly selected pixel-wise LAI estimates from each image in the temporal series were used to compare the performance of these algorithms to the SNAP software. Figs 5 and 6 show the scatter plots associated with the selected pixels from each image for the MLP and UNet compared to SNAP, respectively. In these figures, the x-axis represents SNAP LAI estimates while the y-axis represents the MLP/UNet LAI estimates. These plots show a good agreement between the results obtained using the MLP and UNet algorithms and those obtained using the SNAP software.

To further investigate these results, the LAI estimates obtained using the MLP and UNet algorithms were compared to those obtained using the SNAP software in terms of their correlation, Euclidean distance, MSE, R^2 and inference time metrics (see Table 4). Although the reported results are close for the two algorithms, it is clear that the UNet algorithm obtained better results in all the evaluation metrics when compared to the MLP algorithm. Additionally, there is a noticeable difference between the MLP and the UNet algorithms in terms of inference time. Indeed, the UNet algorithm took less than 2 minutes to process all the temporal series of the T31TCK tile. On the other hand, around 14 minutes were needed to obtain the LAI estimates of a single S2 image using the MLP algorithm. This comparison demonstrates that the UNet algorithm is very adapted to tackle the challenge of processing large amounts of S2 data in order to estimate the corresponding LAI.



(a) 16-th May 2017 (b) 26-th May 2017 (c) 25-th June 2017

(d) 5-th July 2017 (e) 4-th August 2017 (f) 14-th August 2017

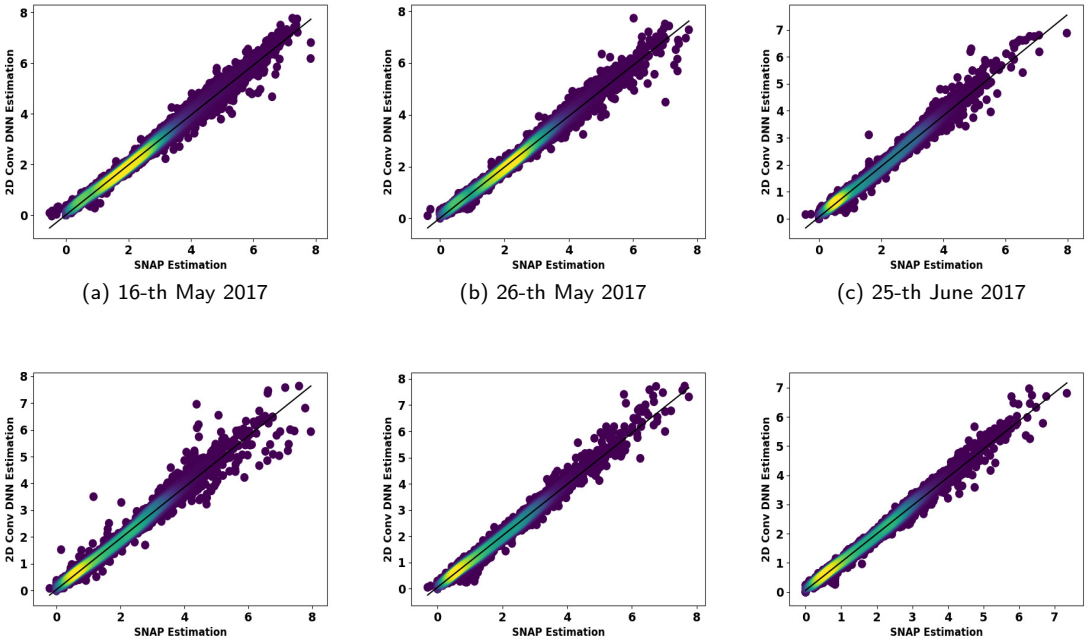
Figure 5: Scatter plot of the LAI estimates obtained using the SNAP software (x-axis) and the MLP algorithm (y-axis) of images of the T31TCK tile. The gradient colors show the concentration of the points on the scatter plot.

Table 4
Evalution metrics of the MLP and UNet algorithms compared to the SNAP software using pixel-wise LAI estimates derived from the images of the T31TCK tile.

Image	Correlation		Euclidean distance		MSE		R^2		Inference time (s)	
	MLP	UNet	MLP	UNet	MLP	UNet	MLP	UNet	MLP	UNet
16-th May 2017	0.992	**0.995**	0.140	**0.089**	0.053	**0.017**	0.961	**0.988**	809.9	**16.9**
26-th May 2017	0.992	**0.995**	0.156	**0.089**	0.065	**0.021**	0.950	**0.983**	810.0	**16.8**
25-th June 2017	0.993	**0.994**	0.146	**0.107**	0.044	**0.022**	0.961	**0.982**	809.9	**16.9**
5-th July 2017	0.991	**0.994**	0.138	**0.100**	0.062	**0.034**	0.951	**0.973**	810.8	**16.8**
4-th August 2017	0.991	**0.995**	0.104	**0.078**	0.034	**0.017**	0.968	**0.984**	812.2	**16.9**
14-th August 2017	0.993	**0.996**	0.106	**0.069**	0.030	**0.011**	0.960	**0.988**	812.8	**16.7**

5.2.2. UNet Compared to SNAP

This section is dedicated to further explore the performance of the UNet algorithm when compared to the SNAP software. Indeed, two types of comparisons were carried out. The first one is a pixel-level comparison where the LAI estimates obtained using the UNet are compared to those obtained using the SNAP software for multiple S2 tiles and different years. This comparison aims at evaluating the model for different geographical locations and image acquisition periods. The uncertainty of the



(a) 16-th May 2017 (b) 26-th May 2017 (c) 25-th June 2017

(d) 5-th July 2017 (e) 4-th August 2017 (f) 14-th August 2017

Figure 6: Scatter plot of the LAI estimates obtained using the SNAP software (x-axis) and the UNet algorithm (y-axis) of images of the T31TCK tile. The gradient colors show the concentration of the points on the scatter plot.

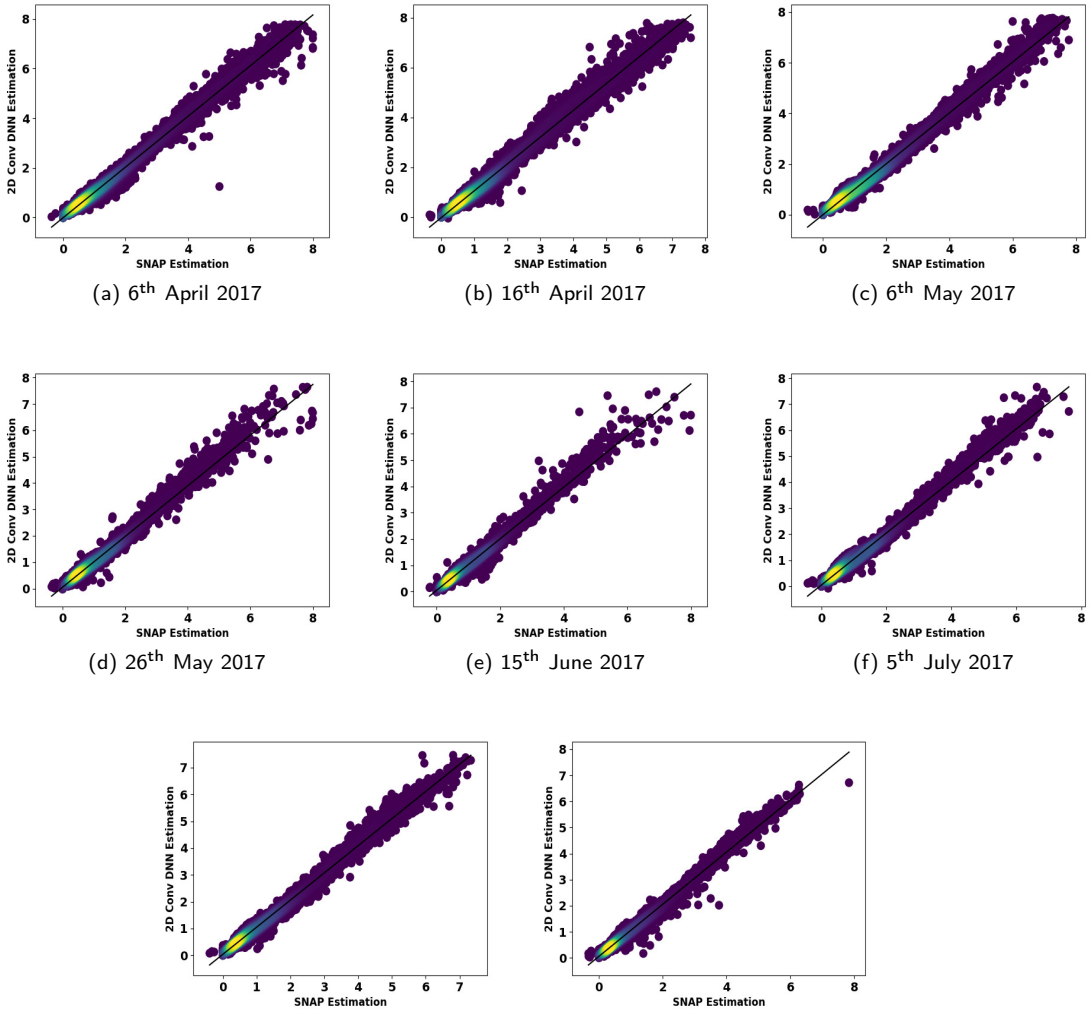
model is also introduced in this comparison. The second comparison aims at investigating the UNet algorithm performance at the parcel-level where different crop types are studied.

- **Pixel-level comparison**

  Two additional S2 tiles were used for this comparison, namely, T30TYT and T30TYM (see Section 2.2). In order to compare the performance of the UNet on these tiles with the performance of the SNAP software, scatter plots of LAI estimates of 20000 randomly selected pixels were used. Figs 7 and 8 show these scatter plots for the T30TYT and T30TM tiles, respectively. One can notice the good agreement of the LAI estimates obtained using the SNAP and the UNet algorithm. In these figures, the LAI estimates range between 0 and 8, which is reasonable from an agronomic point of view. One can notice that there is some level of disagreement between the estimates of SNAP and UNet algorithms on the 14th January and 20th December 2018 (see

Fig. 8). Moreover, SNAP produced some negative LAI values for these two days. Indeed, the two images are associated with a high cloud cover, which may perturb the estimation of LAI. A further analysis was conducted by estimating the evaluation metrics described in Section 5.1 as well as the inference time and the confidence interval of the algorithm. For instance, the correlation, Euclidean distance, MSE and R^2 criteria confirm the ability of the UNet model to estimate LAI values that are very similar to those obtained using the SNAP software (see Tables 5 and 6). Additionally, these results are coherent with the observation derived from the scatter plots in Figs 7 and 8, where a relatively low R^2 values were obtained for 14$^{\text{th}}$ January and 20$^{\text{th}}$ December 2018.

As discussed in Section 4.5.1, activating dropout during the inference step allows estimating the standard deviation (σ) and the confidence interval using Monte Carlo simulations. In these experiments, 30 Monte Carlo simulations were run where σ and the confidence interval were estimated. This experiment shows that the proposed strategy is able to provide the model uncertainty as a quality indicator for the obtained LAI products. Additionally, the reported values of σ are relatively low, which confirms the repeatability of the obtained results. Moreover, the inference time, which was estimated for a single Monte Carlo run, is also very short and coherent with the results obtained in Section 5.2.1.

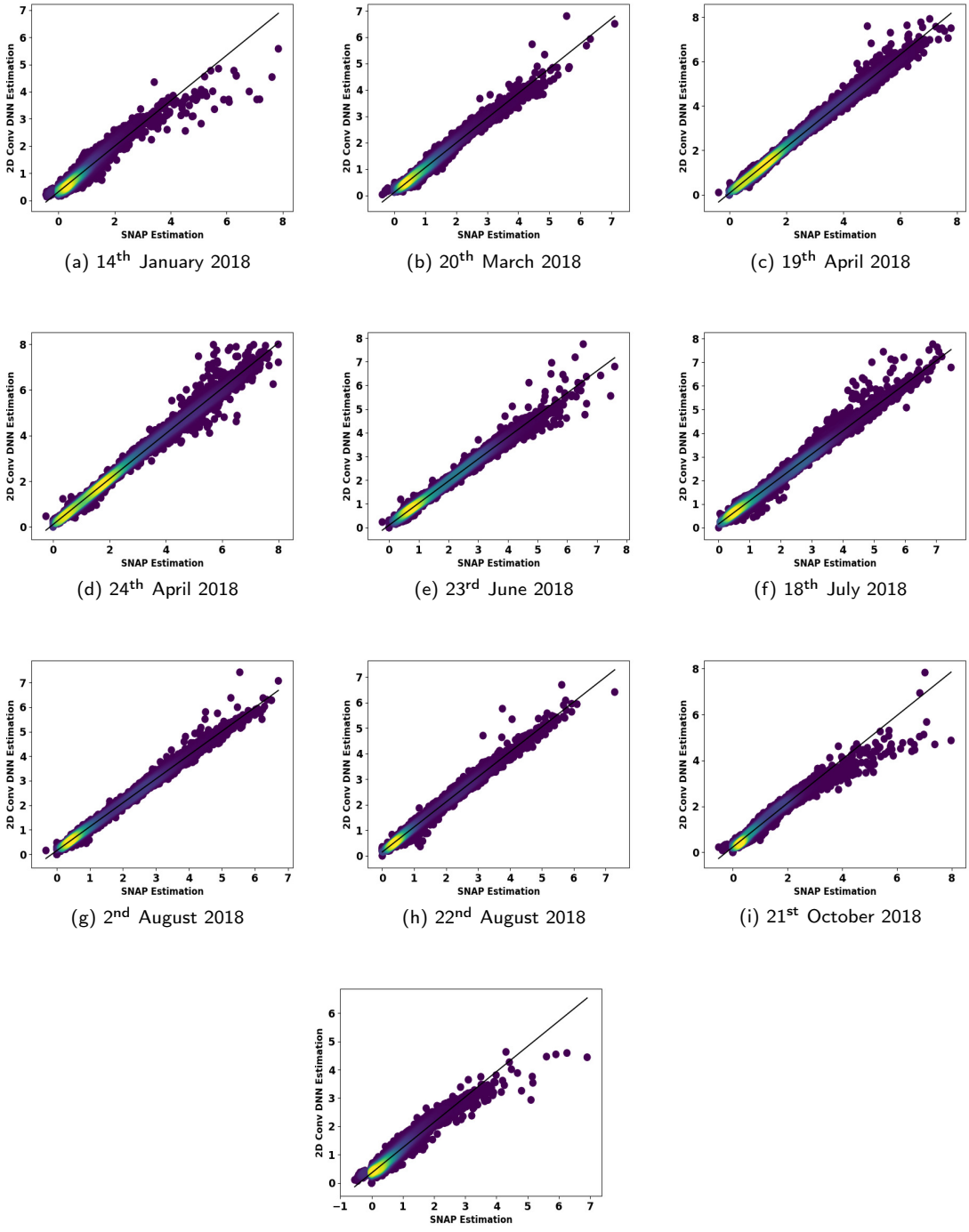


(a) 6th April 2017 (b) 16th April 2017 (c) 6th May 2017

(d) 26th May 2017 (e) 15th June 2017 (f) 5th July 2017

(g) 14th August 2017 (h) 13th September 2017

Figure 7: Scatter plot of the LAI estimates obtained using the SNAP software (x-axis) and the UNet algorithm (y-axis) of images of the T30TYM tile. The gradient colors show the concentration of the points on the scatter plot.



(a) 14th January 2018 (b) 20th March 2018 (c) 19th April 2018

(d) 24th April 2018 (e) 23rd June 2018 (f) 18th July 2018

(g) 2nd August 2018 (h) 22nd August 2018 (i) 21st October 2018

(j) 20th December 2018

Figure 8: Scatter plot of the LAI estimates obtained using the SNAP software (x-axis) and the UNet algorithm (y-axis) of images of the T30TYT tile. The gradient colors show the concentration of the points on the scatter plot.

Table 5
Evalution metrics of UNet algorithm compared to the SNAP software on using pixl-wise LAI estimates derived form the images of the T30TYM tile.

Image	Correlation	Avg. Dist.	MSE	R^2	Confidence Interval	σ	Inference time (s)
6th April 2017	0.995	0.106	0.035	0.983	±0.043	0.140	16.2
16th April 2017	0.993	0.134	0.049	0.972	±0.040	0.137	16.3
6th May 2017	0.994	0.108	0.029	0.984	±0.037	0.131	16.3
26th May 2017	0.988	0.096	0.044	0.952	±0.029	0.110	16.3
15th June 2017	0.988	0.082	0.017	0.970	±0.032	0.115	16.2
5th July 2017	0.990	0.101	0.020	0.976	±0.036	0.119	16.3
14th August 2017	0.994	0.091	0.017	0.984	±0.039	0.126	16.3
13th September 2017	0.993	0.081	0.013	0.979	±0.037	0.122	16.4

Table 6
Evalution metrics of UNet algorithm compared to the SNAP software on using pixl-wise LAI estimates derived form the images of the T30TYT tile.

Image	Correlation	Avg. Dist.	MSE	R^2	Confidence Interval	σ	Inference time (s)
14th January 2018	0.946	0.212	0.077	0.823	±0.034	0.130	18.3
20th March 2018	0.994	0.213	0.110	0.953	±0.053	0.194	21.1
19th April 2018	0.995	0.074	0.014	0.988	±0.038	0.128	18.4
24th April 2018	0.997	0.128	0.029	0.977	±0.038	0.139	18.4
23rd June 2018	0.985	0.141	0.044	0.950	±0.033	0.137	18.3
18th July 2018	0.988	0.164	0.040	0.935	±0.037	0.124	18.7
2nd August 2018	0.978	0.175	0.048	0.928	±0.037	0.134	18.4
22nd August 2018	0.988	0.088	0.0155	0.967	±0.028	0.115	18.2
21st October 2018	0.996	0.227	0.118	0.936	±0.049	0.164	18.3
20th December 2018	0.979	0.214	0.097	0.912	±0.037	0.155	18.3

- **Parcel-level comparison**

Although the previous sections proved that the UNet algorithm is able to provide comparable pixel-wise LAI estimates to those obtained using the SNAP software, studying the performance of the algorithm on different crop types is still mandatory. This comparison was conducted at the parcel-level for different crop types extracted from the French Parcel Registration (RPG) database (Géographique National , IGN). The RPG is an open access database that identifies agricultural parcels covering the French territory of metropolitan France and overseas. This database is useful for agricultural applications, land management services, etc.

For this comparison, the S2 images associated with the T31TCK tile were used (see Section 2). These images were associated with low cloud covers in order to avoid missing data in the LAI estimates and the perturbation caused by clouds. Additionally, the parcels of interest were extracted from the RPG database such that they intersect with the region of interest (ROI). To be more specific, 17 crop types resulting in a total of 52154 parcels, were studied. The distribution of these parcels by crop type is depicted in Table 7. The selection of these crop types was not random. Indeed, these types cover summer, winter, spring and permanent crop types. Table 8 shows the distribution of crop types on annual seasons.

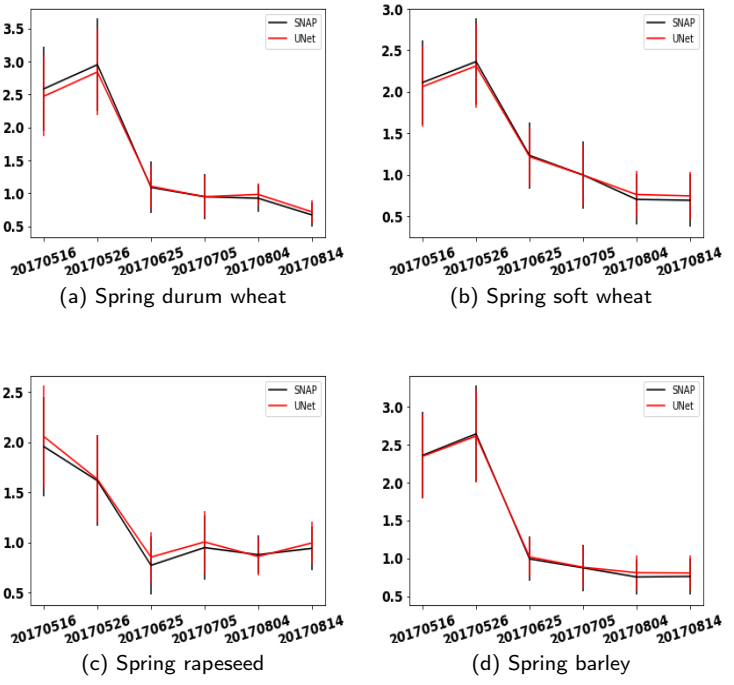
Table 7
Distribution of the analyzed crop types.

Crop	#parcels	Crop	#parcels	Crop	#parcels
Winter durum wheat	85	Sweet corn	36	Winter rye	37
Spring durum wheat	5	Silage corn	1659	Sorghum	691
Winter soft wheat	5609	Corn	4199	Soybeans	849
Spring soft wheat	13	Winter barley	3556	Sunflower	2326
Winter rapeseed	553	Spring barley	435	Fallow	4002
Spring rapeseed	5	Grassland	28094		

Table 8
Distribution of the crop types by season.

Spring	Summer	Winter	Permanent
Spring durum wheat	Sweet corn	Winter durum wheat	Grassland
Spring soft wheat	Silage corn	Winter soft wheat	Fallow
Spring rapeseed	Corn	Winter rapeseed	
Spring barley	Sorghum	Winter barley	
	Soybeans	Winter rye	
	Sunflower		

To analyze the results at the parcel-level, the mean statistic was calculated using the vector dataset containing the parcel delineations and the LAI estimates of the T31TCK tile obtained using the UNet and the SNAP software. This allowed obtaining a parcel-wise LAI estimate at a given time instant. Concatenating all the time instances of these LAI estimates for each parcel resulted in temporal LAI indicators that can be compared. Figs 9, 10, 11 and 12 show the resulting temporal LAI indicators using the UNet algorithm and the SNAP software for spring, summer, winter and permanent crops, respectively. A visual comparison between these parcel-wise LAI estimates proves the good performance of the UNet algorithm as a generic model for LAI estimation on different crop types.

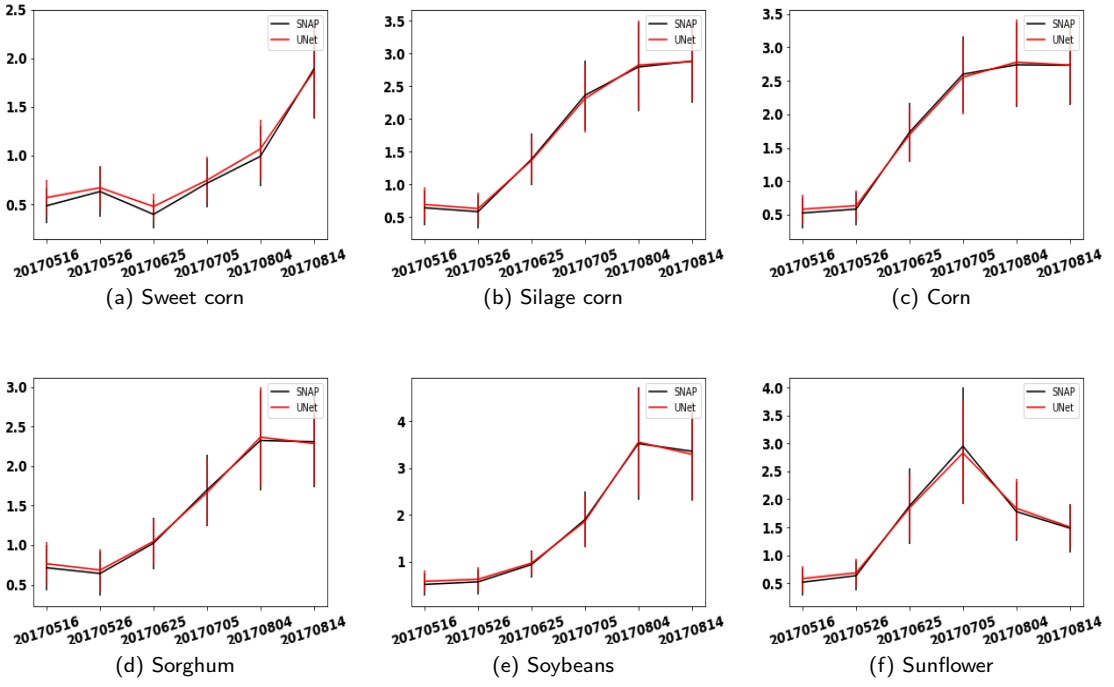


(a) Spring durum wheat (b) Spring soft wheat

(c) Spring rapeseed (d) Spring barley

Figure 9: Parcel-wise temporal LAI indicators of the spring crops estimated using the UNet algorithm(red) and the SNAP software (black).

The parcel-wise LAI estimates were further compared by computing the evaluation metrics illustrated in Section 5.1. Table 9 reports these evaluation metrics for each crop type. It is worth

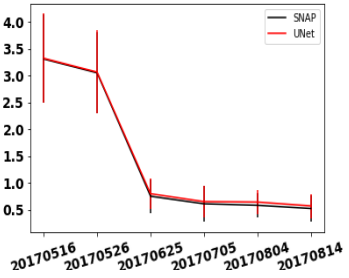


(a) Sweet corn (b) Silage corn (c) Corn

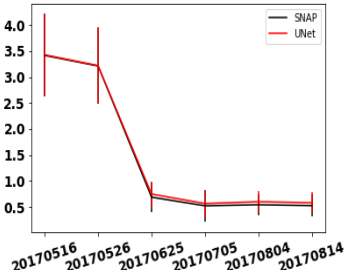
(d) Sorghum (e) Soybeans (f) Sunflower

Figure 10: Parcel-wise temporal LAI indicators of the summer crops estimated using the UNet algorithm(red) and the SNAP software (black).

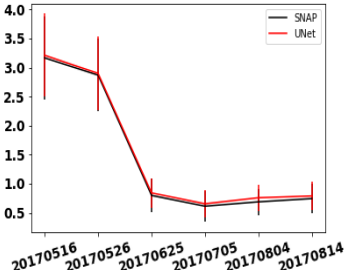
noting that the metrics were calculated for each parcel separately and then averaged for each crop type. One can notice that the UNet algorithm provided very similar results when compared to those obtained by the SNAP software. Moreover, it is clear the UNet algorithm can perform well for different crop types regardless of the season, which is consistent with the previous visual observation.



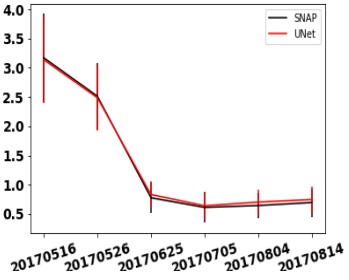
(a) Winter durum wheat



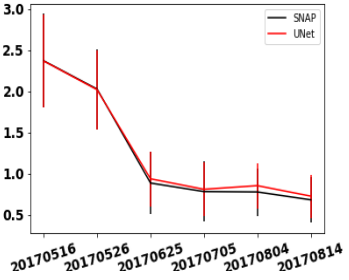
(b) Winter soft wheat



(c) Winter rapeseed

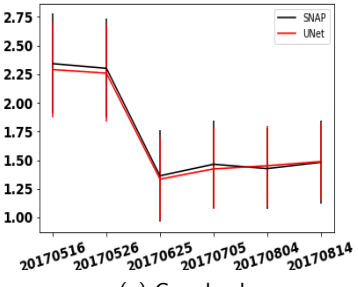


(d) Winter barley

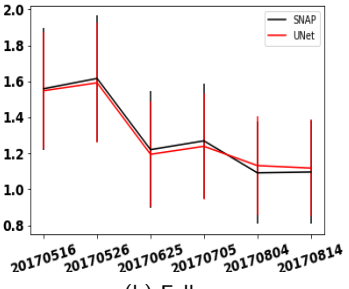


(e) Winter rye

Figure 11: Parcel-wise temporal LAI indicators of the winter crops estimated using the UNet algorithm(red) and the SNAP software (black).



(a) Grassland



(b) Fallow

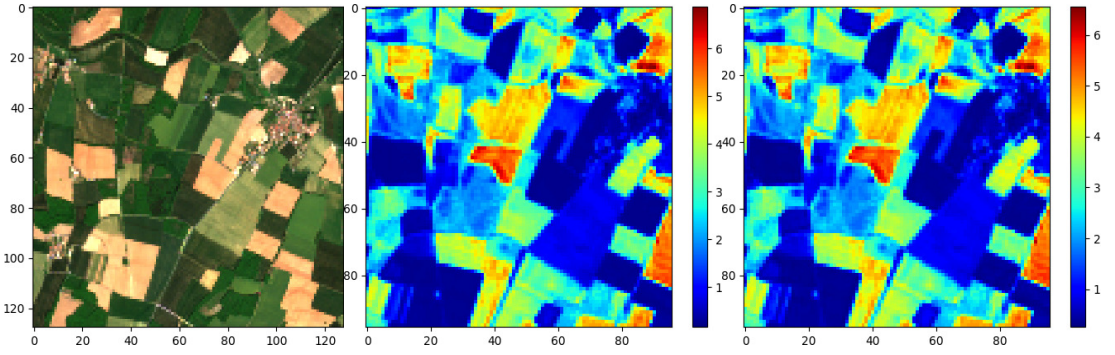
Figure 12: Parcel-wise temporal LAI indicators of the permanent crops estimated using the UNet algorithm(red) and the SNAP software (black).

Table 9
Evalution metrics of UNet algorithm compared to the SNAP software using parcel-wise LAI indicators derived for different crop types.

Crop type	Correlation	MSE	R^2	Euclidean distance					
				16^{th} May 2017	26^{th} May 2017	25^{th} June 2017	5^{th} July 2017	4^{th} Aug 2017	14^{th} Aug 2017
Spring soft wheat	0.997	0.006	0.966	0.085	0.089	0.043	0.043	0.083	0.052
Spring rapeseed	0.994	0.007	0.958	0.099	0.065	0.081	0.056	0.071	0.053
Spring barley	0.996	0.006	0.969	0.067	0.072	0.066	0.043	0.064	0.051
Spring crops	0.996	0.007	0.971	0.091	0.084	0.061	0.042	0.069	0.051
Sweet corn	0.998	0.005	0.978	0.084	0.053	0.084	0.038	0.076	0.063
Silage corn	0.998	0.007	0.990	0.060	0.056	0.080	0.07	0.081	0.063
Corn	0.997	0.009	0.97	0.063	0.058	0.085	0.079	0.078	0.064
Sorghum	0.996	0.009	0.977	0.063	0.057	0.070	0.068	0.078	0.066
Soybeans	0.996	0.035	0.976	0.070	0.061	0.075	0.097	0.187	0.129
Sunflower	0.996	0.059	0.965	0.068	0.059	0.110	0.221	0.087	0.065
Summer crops	0.997	0.021	0.976	0.068	0.057	0.084	0.096	0.098	0.075
Winter durum wheat	0.998	0.006	0.992	0.072	0.080	0.064	0.055	0.072	0.053
Winter soft wheat	0.999	0.007	0.991	0.086	0.077	0.069	0.054	0.063	0.056
Winter rapeseed	0.998	0.009	0.988	0.109	0.090	0.056	0.052	0.073	0.057
Winter barley	0.998	0.007	0.985	0.089	0.077	0.067	0.040	0.065	0.056
winter rye	0.997	0.005	0.986	0.043	0.064	0.062	0.038	0.078	0.050
Winter crops	0.998	0.007	0.988	0.080	0.077	0.064	0.048	0.070	0.054
Grassland	0.992	0.007	0.944	0.071	0.066	0.073	0.059	0.052	0.044
Fallow	0.982	0.006	0.848	0.05	0.057	0.073	0.056	0.058	0.046
Permanent crops	0.987	0.006	0.896	0.062	0.062	0.073	0.058	0.055	0.045

5.3. Visualization of of LAI estimates and the UNet feature maps

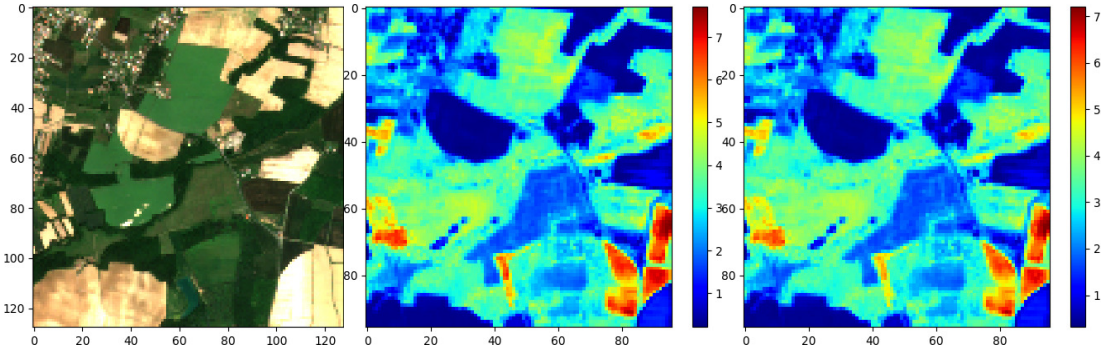
This section is dedicated to visualize the LAI estimates of two image patches from the validation dataset as well as their corresponding feature maps extracted from the different layers of the UNet algorithm, where we try to explain how filters in the layers of the UNet architecture react to specific land cover classes in the input image and the resulting activation. For instance, Figs 13 and 14 show two image patches [a], the corresponding SNAP LAI estimates [b] and UNet LAI estimates [c]. One can notice the similarity between the LAI and UNet estimates, which is consistent with the previous comparisons. Note that there is a difference between the size of the input image and the size of the UNet LAI estimates due to the added cropping layer before the output layer in the proposed architecture.



(a) Image patch (b) LAI estimates using SNAP (c) LAI estimates using the UNet

Figure 13: An image patch (patch#2) from the validation datasets [a] along with its corresponding LAI estimates using the SNAP software [b] and the UNet algorithm [c].

The first convolutional and activation layers when estimating the LAI using these two patches are depicted in Figs 15 and 16. Visualizing the UNet layers allows understanding how the UNet algorithm learns the different features from the input image. For instance, the first filter in the convolutional layer (at row 1 and column 1 in Figs 15[a] and 16[a]) reacts to the soil land cover in these patches. This becomes clearer in the output of the first ReLU activation layer (at row 1 and column 1 in Figs 15[b] and 16[b]). Similarly, another convolutional filter (at row 3 and column 8) reacts to vegetation in these patches. It is worth noting that these filters do not only learn features associated with the land cover class but also border features (*e.g.,* at row 3 and column 2 in Figs 15 and 16). These learned features



(a) Image patch (b) LAI estimates using SNAP (c) LAI estimates using the UNet

Figure 14: An image patch (patch#2) from the validation datasets [a] along with its corresponding LAI estimates using the SNAP software [b] and the UNet algorithm [c].

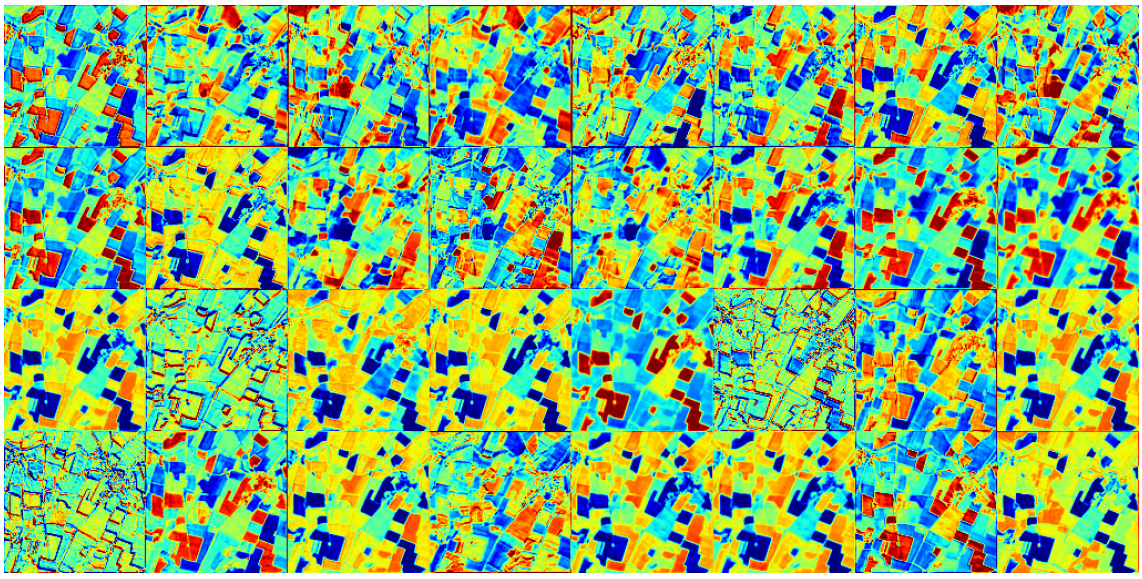
for different land covers can be of interest to develop corp-specific algorithm for LAI estimation.

6. Conclusion and Future Work

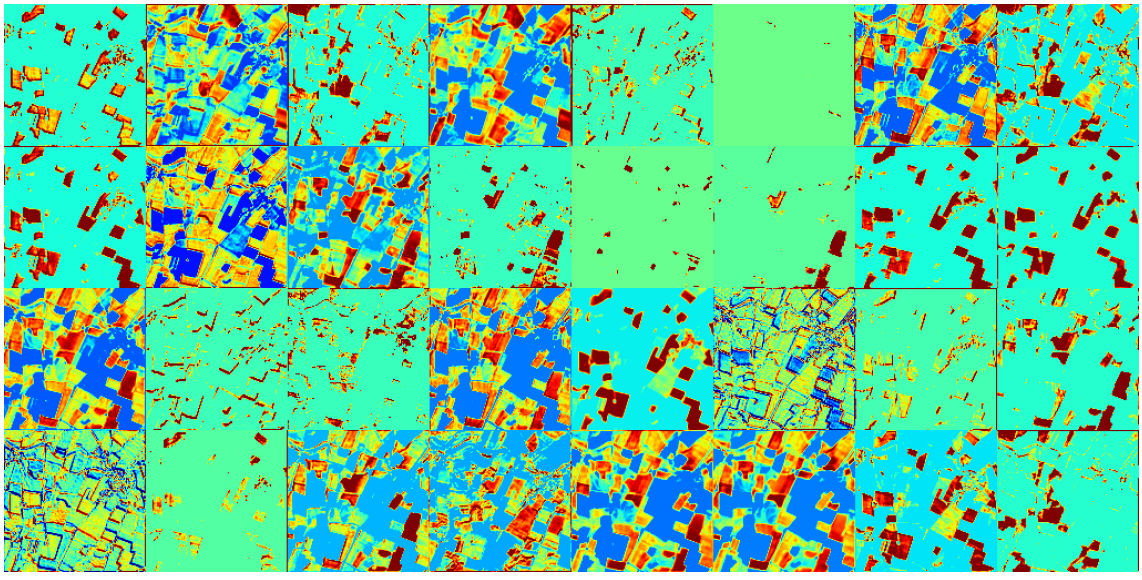
In this paper, we proposed a new procedure to estimate the LAI using S2 images at 20 m resolution. This procedure is based on the UNet algorithm, a 2-D convolutional network architecture. The proposed procedure allows an accurate, fast and scalable estimation of the LAI. The architecture of UNet was modified to include dropout layers that allow overcoming overfitting. Moreover, these dropout layers can be activated at the inference step in order to be used as a Bayesian approximation. Hence, the uncertainty of the model can be estimated and provided as a quality indicator of the obtained results.

It is worth noting that the proposed procedure was trained using LAI estimates of the SNAP software, where the quality indicators provided with these estimates were used for the selection of the valid samples.

In order to justify the algorithm selection, two competing algorithms, the UNet and the MLP, were compared to the SNAP software to estimate the LAI in terms of the MSE, Euclidean distance, R^2 score and inference time. Although the values of some of these metrics were close for the competing algorithms, the UNet algorithm obtained better values in all metrics. Moreover, the UNet algorithm



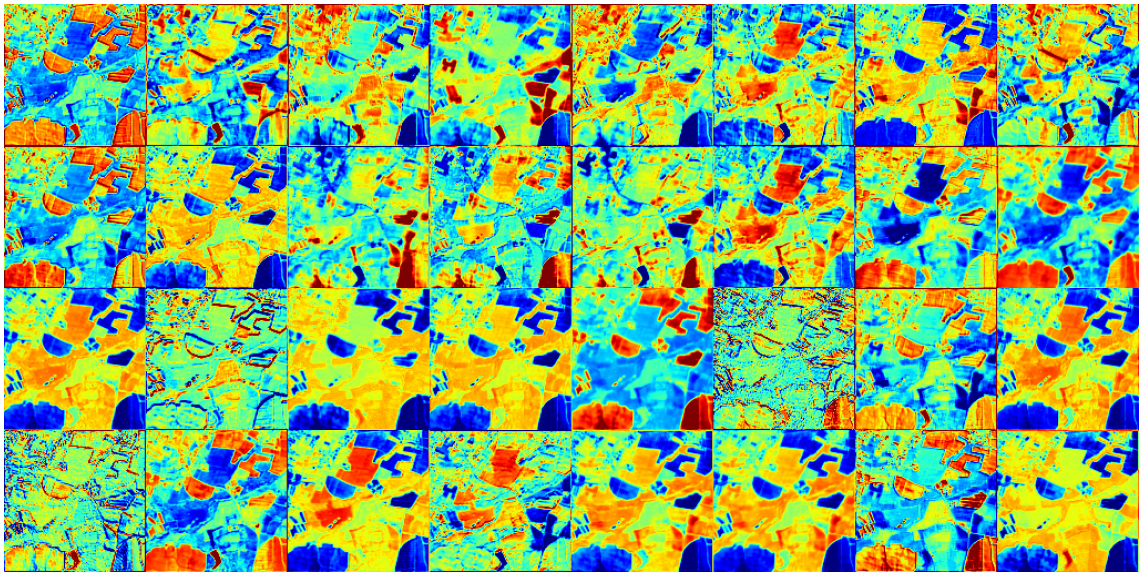
(a) 1st convolutional layer



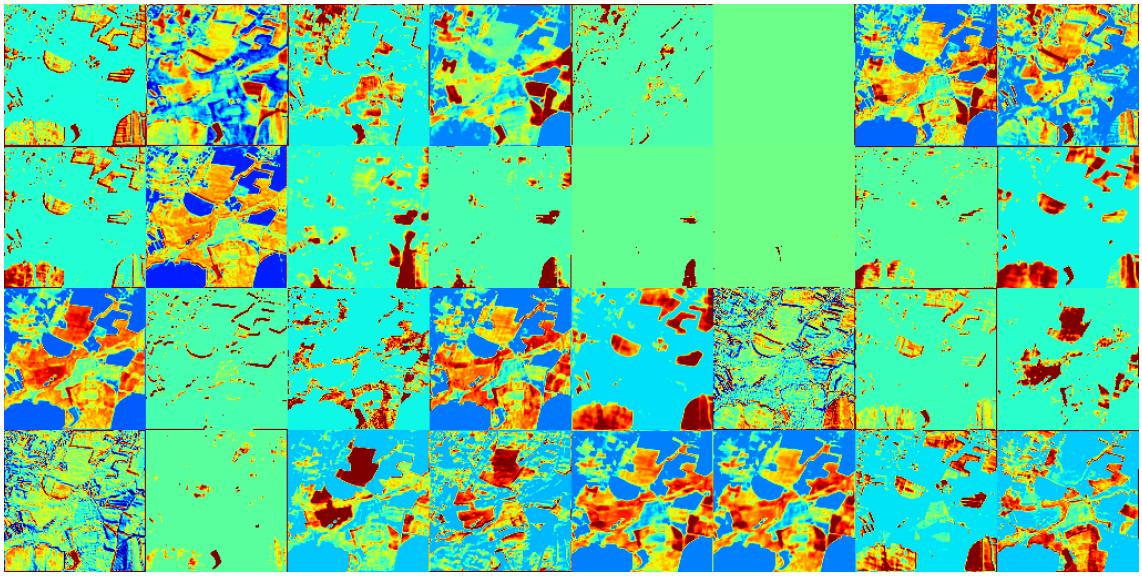
(b) 1st activation layer

Figure 15: Feature maps of the first layers for a patch from the validation dataset.

458 reduced the inference time to 18 seconds when compared to the MLP regressor (13.5 minutes) and the
459 SNAP software (15 minutes). This time reduction makes the UNet algorithm a good fit for processing
460 large amount of S2 images in order to provide LAI products that are ready to use for other applications.
461 The proposed procedure was also validated on multiple crop types that contain Spring, Summer, Win-



(a) 1st convolutional layer



(b) 1st activation layer

Figure 16: Feature maps of the first layers for a patch from the validation dataset.

ter and permanent crops. To carry out the latter validation, parcel delineations from the French Parcel Registration database were used to estimate the mean statistic of the LAI at the parcel-level, which allows constructing a temporal LAI indicator for each parcel. The obtained temporal LAI indicators using the UNet algorithm were compared to those obtained using the SNAP software, where the MSE

values between these estimates were very low. This confirmed the capacity of the proposed procedure as a generic LAI estimator regardless of the crop type. Future work will focus on enriching the LAI training dataset by including other geographical extents that contain other vegetation and soil types, which will improve the performance of the model. In a second step, additional models can be developed for other biophysical parameters such as the fCover. It can also be of interest to investigate a deep learning model with multiple outputs that allows estimating multiple biophysical parameters at once.

References

Albughdadi, M., Kouamé, D., Rieu, G., Tourneret, J.Y., 2017. Missing data reconstruction and anomaly detection in crop development using agronomic indicators derived from multispectral satellite images, in: Proc. IGARSS, IEEE, Texas, USA.

Asner, G., Scurlock, J., A. Hicke, J., 2003. Global synthesis of leaf area index observations: implications for ecological and remote sensing studies. Global Ecology and Biogeography 12, 191–205.

Bacour, C., Baret, F., Béal, D., Weiss, M., Pavageau, K., 2006. Neural network estimation of LAI, fAPAR, fCover and LAI× Cab, from top of canopy MERIS reflectance data: Principles and validation. Remote sensing of environment 105, 313–325.

Benesty, J., Chen, J., Huang, Y., Cohen, I., 2009. Pearson correlation coefficient, in: Noise reduction in speech processing. Springer, pp. 1–4.

Bonan, G., Oleson, K., Vertenstein, M., Levis, S., Zeng, X., Dai, Y., Dickinson, R., Yang, Z., 2002. The land surface climatology of the community land model coupled to the ncar community climate model. Journal of climate 15, 3123–3149.

Brown, L., Ogutu, B., Dash, J., 2019. Estimating forest leaf area index and canopy chlorophyll content with Sentinel-2: An evaluation of two hybrid retrieval algorithms. Remote Sensing 11, 1752.

Danielsson, P., 1980. Euclidean distance mapping. Computer Graphics and image processing 14, 227–248.

Djamai, N., Fernandes, R., Weiss, M., McNairn, H., Goïta, K., 2019. Validation of the Sentinel simplified level 2 product prototype processor (SL2P) for mapping cropland biophysical variables using Sentinel-2/MSI and Landsat-8/OLI data. Remote sensing of environment 225, 416–430.

Gal, Y., Ghahramani, Z., 2016. Dropout as a Bayesian approximation: Representing model uncertainty in deep learning, in: international conference on machine learning, pp. 1050–1059.

Glenn, E., Huete, A., Nagler, P., Nelson, S., 2008. Relationship between remotely-sensed vegetation indices, canopy attributes and plant physiological processes: What vegetation indices can and cannot tell us about the landscape. Sensors 8, 2136–2160.

Hinton, G., Srivastava, N., Swersky, K., . Geoffrey Hinton neural networks for machine learning. http://www.cs.toronto.edu/ tijmen/csc321/slides/lecture_slides_lec6.pdf.

Géographique National (IGN), I., . Le Registre Parcellaire Graphique. http://professionnels.ign.fr/rpg.

Knyazikhin, Y., Kranigk, J., Myneni, R., Panfyorov, O., Gravenhorst, G., 1998. Influence of small-scale structure on radiative transfer and photosynthesis in vegetation canopies. Journal of Geophysical Research: Atmospheres 103, 6133–6144.

Kwon, Y., Won, J., Kim, B., Paik, M., 2018. Uncertainty quantification using Bayesian neural networks in classification: Application to ischemic stroke lesion segmentation .

Le Maire, G., Francois, C., Dufrene, E., 2004. Towards universal broad leaf chlorophyll indices using PROSPECT simulated database and hyperspectral reflectance measurements. Remote sensing of environment 89, 1–28.

Liang, S., 2007. Recent developments in estimating land surface biogeophysical variables from optical remote sensing. Progress in Physical Geography 31, 501–516.

Myneni, R., Knyazikhin, Y., Glassy, J., Votava, P., Shabanov, N., . User's guide FAPAR LAI (ESDT:MOD15A2 8-day composite NASA MODIS land algorithm) .

Rivera, J., Verrelst, J., Delegido, J., Veroustraete, F., Moreno, J., 2014. On the semi-automatic retrieval of biophysical parameters based on spectral index optimization. Remote Sensing 6, 4927–4951.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation, in: International Conference on Medical image computing and computer-assisted intervention, Springer. pp. 234–241.

Rumelhart, D., Hinton, G., McClelland, J., et al., 1986. A general framework for parallel distributed processing. Parallel distributed processing: Explorations in the microstructure of cognition 1, 26.

Running, S., Coughlan, J., 1988. A general model of forest ecosystem processes for regional applications I. hydrologic balance, canopy gas exchange and primary production processes. Ecological modelling 42, 125–154.

Sellers, P., Dickinson, R., Randall, D., Betts, A., Hall, F., Berry, J., Collatz, G., Denning, A., Mooney, H., Nobre, C., et al., 1997. Modeling the exchanges of energy, water, and carbon between continents and the atmosphere. Science 275, 502–509.

d'Etudes Spatiales (CNES), C.N., . French Access to the Sentinel Products. https://peps.cnes.fr.

Verrelst, J., Alonso, L., Camps-Valls, G., Delegido, J., Moreno, J., 2011. Retrieval of vegetation biophysical parameters using Gaussian process techniques. IEEE Transactions on Geoscience and Remote Sensing 50, 1832–1843.

Verrelst, J., Muñoz, J., Alonso, L., Delegido, J., Rivera, J., Camps-Valls, G., Moreno, J., 2012. Machine learning regression algorithms for biophysical parameter retrieval: Opportunities for Sentinel-2 and-3. Remote Sensing of Environment 118, 127–139.

Verrelst, J., Rivera, J., Veroustraete, F., Muñoz-Marí, J., Clevers, J., Camps-Valls, G., Moreno, J., 2015. Experimental Sentinel-2 LAI estimation using parametric, non-parametric and physical retrieval methods: A comparison. ISPRS Journal of Photogrammetry and Remote Sensing 108, 260–272.

Vinué, D., Camacho, F., Fuster, B., 2018. Validation of Sentinel-2 LAI and FAPAR products derived from SNAP toolbox over a cropland site in Barrax and over an agroforested site in Liria (Spain). Fifth recent advances in quantitative remote sensing , 248.

Waldner, F., Horan, H., Chen, Y., Hochman, Z., 2019. High temporal resolution of leaf area data improves empirical estimation of grain yield. Scientific reports 9, 1–14.

Waldner, F., Lambert, M.J., Li, W., Weiss, M., Demarez, V., Morin, D., Marais-Sicre, C., Hagolle, O., Baret, F., Defourny, P., 2015. Land cover and crop type classification along the season based on biophysical variables retrieved from multi-sensor

high-resolution time series. Remote Sensing 7, 10400–10424.

Weiss, M., Baret, F., . S2ToolBox level 2 products: LAI, FAPAR, FCOVER .

Weiss, M., Baret, F., Myneni, R., Pragnère, A., Knyazikhin, Y., 2000. Investigation of a model inversion technique to estimate canopy biophysical variables from spectral and directional reflectance data .

Werbos, P., 1974. Beyond regression: New tools for prediction and analysis in the behavior science. Unpublished Doctoral Dissertation, Harvard University .

Yan, K., Park, T., Yan, G., Chen, C., Yang, B., Liu, Z., Nemani, R., Knyazikhin, Y., Myneni, R., 2016. Evaluation of MODIS LAI/FPAR product collection 6. part 1: Consistency and improvements. Remote Sensing 8, 359.

Yang, W., Tan, B., Huang, D., Rautiainen, M., Shabanov, N., Wang, Y., Privette, J., Huemmrich, K., Fensholt, R., Sandholt, I., et al., 2006. Modis leaf area index products: From validation to algorithm improvement. IEEE Transactions on Geoscience and Remote Sensing 44, 1885–1898.