doi:10.20944/preprints201906.0270.v1

Article

Land Cover Maps Production with High Resolution Satellite Image Time Series and Convolutional Neural Networks: Adaptations and Limits for Operational Systems

Andrei Stoian[1], Vincent Poulain[2], Jordi Inglada[3,4]*, Victor Poughon[4], Dawa Derksen[3]

[1] Thales/SIX/ThereSiS, Palaiseau, France;
[2] Thales Services, Toulouse, France;
[3] Centre d'Etudes Spatiales de la BIOsphere (CESBIO), Université de Toulouse, CNES/CNRS/IRD/UPS/INRA, Toulouse, France;
[4] Centre National d'Etudes Spatiales (CNES), Toulouse, France.
* Correspondence: jordi.inglada@cesbio.eu

Abstract: The Sentinel-2 satellite mission offers high resolution multispectral time series image data, enabling the production of detailed land cover maps globally. At this scale, the trade-off between processing time and result quality is a central design decision. Currently, this machine learning task is usually performed using pixelwise classification methods. The radical shift of the computer vision field away from hand engineered image features and towards more automation by representation learning comes with many promises, including higher quality results and less engineering effort. In this paper we assess fully convolutional neural networks architectures as replacements for a Random Forest classifier in an operational context for the production of high resolution land cover maps with Sentinel-2 time series at the country scale. Our contributions include a framework for working with Sentinel-2 L2A time series image data, an adaptation of the U-Net model for dealing with sparse annotation data while maintaining high resolution output, and an analysis of those results in the context of operational production of land cover maps.

Keywords: Land cover mapping; Convolutional neural networks; UNET; Sentinel-2.

1. Introduction

1.1. The need for land cover mapping

Land cover information is a crucial asset for many scientific and operational applications as for instance climate modeling, vegetation monitoring, urban management, etc. Many of these applications need high resolution up to date land cover information. Land cover databases are freely available over wide areas, for example Corine Land Cover (CLC) over the European Union [1] or ESA's CCI land cover products globally available. However they often fail to meet the needs of many users: CLC has a minimum mapping unit of 25 ha and is only updated every 6 years, and ESA's Globcover cover maps have a resolution of 300 m [2].

Until recently, automatic land cover mapping using remote sensing imagery was only performed at coarse resolution (CCI, for instance). Higher resolution imagery was only used through photo-interpretation (as for CLC) leading to high costs and long production delays.

With the availability of Landsat and more recently Sentinel [3] image time series, the possibility of accurate automatic land cover mapping at high resolution has become a reality [4]. Some one-shot initiatives have been published, as for instance the GLC-30 [5]. To the best of our knowledge, the only recurrent operational (large scale, recurrent production) automatic high resolution land cover mapping is the French Theia Land Data Centre which delivers a 10 m resolution national land cover map with

17 classes and an accuracy of 89% [6,7]. The approach has to be fully automatic in order to allow the delivery of a map every year less than 3 months after the reference period.

The approach used by Theia is described in [8] and uses pixel-based supervised classification, leveraging existing (and therefore out of date) data bases as annotations for training and all available Sentinel-2 imagery acquired during the mapping period.

This approach allows to deal with the following crucial aspects in order to achieve efficient and robust large scale land cover mapping:

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

Although the approach presented in [8] is used for the production of annual land cover maps of metropolitan France, the recognition of some classes of the nomenclature, mainly those where pixel context is important, is not satisfactory and new classification approaches need to be explored.

1.2. The state of the art is supervised classification

The literature on land cover mapping has established supervised classification methods as the most efficient way of performing the task. Most papers in the literature work on small areas, particular sets of classes (non exhaustive nomenclatures) and perfect reference data for training. In this paper, we concentrate on operational approaches, that is the development and delivery of reliable and recurrent data products within a pre-defined time schedule [5]. This is different from experimental approaches for large area land cover mapping and monitoring [9], [10], which focus on the development and performance testing of novel algorithms and models.

One of the main challenges for scaling land cover mapping to very large areas is dealing with the following issues:

1. Reference data is costly to obtain and one has to rely on existing data bases which contain errors or are out of date.
2. Reference data is sparse, only a small amount of the pixels are annotated.
3. Large areas may present climatic and topographic variabilities introducing different behaviours for the same class across different landscapes.

1.3. The promise of deep learning

Although the overall accuracy achieved with the methodology of [8] is very high and most of the classes are very well recognized, some important confusions remain, mostly for classes where the pixel context may be important (discontinuous urban fabric, for instance). Other classes are very difficult to distinguish from each other because of their similarity or because of the inherent intra-class variability.

These limits of the current machine learning approach used (pixel-based, engineered features) may be overcome by the use of more powerful approaches recently made available. The recent developments in the field of deep learning (DL) and more precisely convolutional neural networks (CNN) need to be assessed as candidates for the improvement of fully automatic land-cover mapping using high resolution image time series as those acquired by Sentinel-2.

The complexity and intra-class variability in the problem could be handled by the extraction of features of increasing semantic level in the DL structure. The spatial context of pixels could be taken into account by the convolutional structure of the networks.

For a comprehensive review of DL applied to Remote Sensing, we refer the reader to [11].

doi:10.20944/preprints201906.0270.v1

1.4. Semantic segmentation vs. patch based methods

Two kinds of CNN can be used for land cover mapping. Patch-based approaches assign a class to the center pixel of an image patch and are a straightforward application of image classification networks to the task of semantic segmentation (i.e. the classification of every pixel in an image). Semantic segmentation can also be addressed by specific network architectures (Segnet [12], U-Net [13]) which are more computationally efficient and can also learn spatial dependencies between classes. Segmentation networks, especially U-Net, have another advantage over patch methods, since the latter have discontinuities in their prediction which appear as label noise in the maps [14] and thus require a smoothing step in post-processing [15]. Fully convolutional networks like U-Net implicitly perform a regularization step similar to what Conditional Random Fields (CRF) do. However, a post-processing step using CRF increases complexity and is rather sensitive to hyper-parameter selection [16].

Volpi et al. [17] compare a simple fully-convolutional patch based method to a downsample-upsample network similar to U-Net but without skip connections.

In the case of operational land cover mapping of large geographical areas, compared with semantic segmentation techniques patch based approaches have an important disadvantage because of their lower computational efficiency.

Segmentation network architectures were initially applied to dense annotations, in other words, all the pixels of the training image patches need to be annotated. In the case of sparse annotations (reference labels are given as polygons which may not be adjacent to each other, may represent only parts of the objects and cover a small portion of the area of interest), for patch based methods one simply needs to exclude all patches centered on non-annotated pixels. However, segmentation networks need to be adapted in order to exploit the local class correlations available in the sparse polygons which patch-based approaches are not able to exploit.

One of the main drawbacks of segmentation CNNs, such as Segnet and U-Net, is their lack of geometric accuracy, i.e. the prediction masks are too smooth and do not show sharp straight borders between classes. This is most problematic in the case of cartographic products where classes have sharp edges (e.g. buildings, roads, field boundaries). To improve upon this situation, in [18], the authors introduced additional supervision in the form of an class edge detector into the final loss function. In [19] a modified U-Net with residual connections and a guided filter were used in post-processing to straighten building prediction mask edges.

Practical large scale reference data for land cover prediction, outside of academic datasets, is hard to obtain. Maggiori et al. [20] run an initial training pass on roughly annotated data automatically generated from OpenStreetMap and then fine-tune the network on precise manual annotations. In this fashion, they train a multi-scale patch classifier on 10 3000×3000 pixels from 0.5m resolution Pleiades images.

Methods for multi-temporal multi-source land cover classification were proposed by Kussul et al. [21], learning a classifier for 7x7 patches with 54 channels, containing 19 temporal samples sourced from Landsat 8 and Sentinel 1A data. Ji et al. [22] adapt the patch methods to better take into account the temporal dynamics by using 3D 3x3x3 convolutions (2 spatial dimensions and 1 time dimension). They show results on 4m and 15m resolution RGB+NIR tiles.

Creating classifiers that specifically take into account temporal dynamics for multi-temporal imagery is important, since these dynamics could be lost when training on large amounts of data. In [23], a deep Recurrent Neural Network (using Long Short Term Memory, LSTM, cells) is used for change detection between two satellite images.

1.5. Deep Learning on a High Performance Computing cluster

Land cover map production at a 10m resolution on a country-wide scale has large computing and storage requirements. Deep Learning training is highly computationally intensive and benefits from large amounts of data. Moreover, several dozens of passes (epochs) need to be made over the training data. In this work, a HPC cluster infrastructure was available to support this. Yet, efficiently scaling distributed stochastic gradient descent for an HPC architecture while dealing with imbalance in the reference data is a challenge.

1.6. Contributions

In our work we address the challenges described above - country-scale data with high temporal frequency, sparse annotations across varying climatic regions, imprecise segmentation produced by standard deep CNN approaches - by making these contributions:

1. a framework of patch generation, data augmentation and patch distribution in a computing cluster that alleviates the very high class imbalance in the data set;
2. a novel adaptation of the U-Net model for dealing with sparse data, while producing detailed and fine-grained class predictions;
3. we show results in an operational setting on the country scale for multi-temporal multispectral production-quality data obtained from the Sentinel-2 constellation.

2. Data

The data used in this work was selected in order to assess the performance of the proposed algorithms allowing to draw conclusions for an operational setting. Imagery volume, reference labels quality and quantity, as well as landscape and eco-climatic variability are representative of operational, country-scale land cover map production contexts.

2.1. Remote sensing imagery

Eleven Sentinel-2 tiles (100 km × 100 km) covering different areas of metropolitan France were selected. They are shown in figure 1. For each tile, all available Sentinel-2 through the Theia Land Data Centre (http://www.theia-land.fr/en/presentation/products) for the year 2016 were used. The Sentinel-2 products are orthorectified prior to their release by ESA (Level-1C) and then, Theia processing chains based on the algorithms described in [24] perform cloud (and cloud shadow) screening and atmospheric corrections to produce the Level-2A imagery used in this work. Only the bands with 10 m resolution (blue, green, red, near-infrared) and 20 m resolution (4 bands in the red-edge and 2 in the short-wave infrared) will be used. The 20 m resolution bands are resampled to 10 m using a bicubic interpolation with the appropriate spatial radius to avoid over-smoothing or aliasing.

As described in [8], linear temporal gapfilling is used to estimate image reflectance for pixels flagged as clouds, cloud shadows, saturation or non acquired and a subsequent temporal resampling is applied in order to reconstruct temporal profiles having identical time stamps. This allows to have the same image features for all pixels despite the different orbit cycles (effective acquisition dates), and despite the presence of any artifact such as clouds.

At the end of this procedure, each pixel at 10 m resolution is described by 330 features (10 spectral bands for 33 dates).

2.2. Reference data sources and nomenclature

The reference data set is composed of the same existing data sources and produced with the same procedure as described in [8]. Therefore, the nomenclature is composed of the following classes: **CUF** Continuous urban fabric; **DUF** Discontinuous urban fabric; **ICU** Industrial or commercial units; **RSF** Road surfaces; **ASC** Annual summer crops; **AWC** Annual winter crops; **IGL** Intensive grassland; **ORC** Orchards; **VIN** Vineyards; **BLF** Broad-leaved forest; **COF** Coniferous forest; **NGL** Natural grasslands; **WOM** Woody moorlands; **BDS** Beaches, dunes and sand plains; **BRO** Bare rock; **GPS** Glaciers and perpetual snow; **WAT** Water bodies.

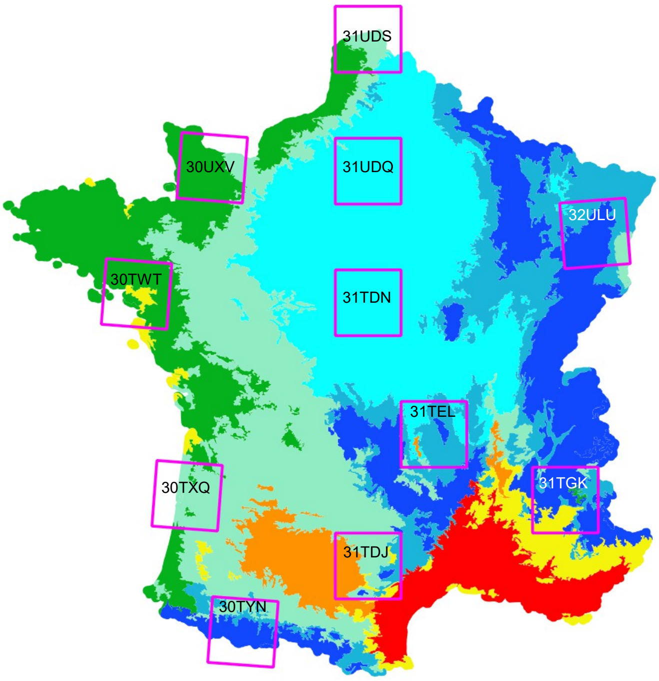


Figure 1. Imagery used in this work. Location of the 11 Sentinel-2 tiles and eco-climatic areas.

2.3. Training and test data

As usual in supervised classification, we randomly split the available reference data in 2 disjoint sets, one used for classifier training and the other one used to measure the quality of the classification using standard metrics.

Since the reference data is composed of polygons where the pixels can be highly correlated (agricultural plots, a forest stand, etc.), we ensure that training and test polygons are different, which avoids that two pixels from the same polygon, and therefore likely very similar, are used one for training and the other for validation. This avoids optimistic accuracy estimations. For this work, 67% of the data was used for training and 33% for validation. The split was done so that these ratios are enforced for each image tile and class.

Table 1 shows the surface in hectares for each class and each tile. The variability in terms of class composition between the tiles is related to eco-climatic and landscape variability. For instance, BRO is a rare class in most tiles except for T30TYN and T31TGK where it is one of the most represented classes. Similar behaviours can be observed for other classes and other tiles.

3. Method

In this work we address the problem of *dense* supervised classification of land cover with many multispectral temporal samples, learned using *sparse annotations*. In many remote sensing applications, labeled data is highly sparse: most of the pixels are not labeled, and labeled areas consist of isolated polygons which may be inside objects without fitting their boundaries. To tackle this problem, we propose to use a new variant of the fully-convolutional U-Net architecture.

The U-Net [13] network is a classical architecture to address the problem of semantic segmentation. Indeed, it is a fully-convolutional segmentation network initially applied to biomedical images. Its symmetrical architecture is able to gather context in the encoding phase and accurately localize features in the decoding phase, as illustrated figure 2.

Due to the high level of sparsity of the labeled data, this classical network suffers from some limitations in our specific application. In our data sets, annotated polygons are sparse and do not necessarily include the boundaries between classes (e.g: the edge between a cultivated field and a

	Tile Name											
	T30TWT	T30TXQ	T30TYN	T30UXV	T31TDJ	T31TDN	T31TEL	T31TGK	T31UDQ	T31UDS	T32ULU	Total
ASC	5650	3306	2415	6738	3700	4154	3849	159	3327	1951	6900	42149
AWC	5204	85	572	10235	7982	22297	5617	866	32808	9125	4511	99302
BLF	2063	2659	11636	3152	9358	27161	10855	8144	12829	4465	17184	109506
COF	2546	112152	7544	803	10961	5594	47692	57027	2442	94	52772	299627
NGL	272	0	18862	0	7735	62	1307	51040	24	201	1484	80987
WOM	3856	3952	11312	552	3788	1433	3465	18915	749	466	518	49006
CUF	368	1019	203	863	106	157	1183	20	8619	906	941	14385
DUF	5962	11108	1536	3059	1501	1949	7019	833	31143	5045	6514	75669
ICU	5331	7951	842	3116	1172	2758	4689	358	25165	6525	4647	62554
RSF	108	694	112	138	61	179	584	73	2031	779	615	5374
BRO	0	0	20990	48	27	0	3	53160	0	0	20	74248
BDS	140	4327	0	374	0	107	0	923	0	348	0	6219
WAT	237383	219169	814	125722	2204	5045	1403	2293	2567	113073	3305	712978
GPS	0	0	63	0	0	0	0	2519	0	0	0	2582
IGL	7595	648	7891	18446	5319	5493	23270	6165	4544	2479	9458	91308
ORC	48	9	2	120	107	55	5	378	144	19	37	924
VIN	139	1620	6	0	1002	174	25	3	0	0	190	3159
Total	276672	368708	84807	173373	55028	76625	110973	202885	126397	145481	109102	1730051

Table 1. Areas occupied by the different land cover types (in hectares) in the 11 tiles

forest). As confirmed by our initial evaluation of the original U-Net architecture in Section 5, this leads to blurry, soft and erroneous class boundaries when applying the network in a dense prediction setting.

In this section we introduce and discuss FG-UNET our modified U-Net architecture that is adapted to our specific problem of land cover classification.

3.1. U-Net architecture overview

The original U-Net architecture of [13] contains symmetric downsampling and an upsampling paths. On the downsampling side, the network takes as input an image patch with C channels of size $N \times N \times C$. This side contains the following operations: $\texttt{5x [2x [64*2}^{\texttt{i}} \times \texttt{conv3x3}_{\texttt{i}}\texttt{(previous) + relu activation], maxpool2x2]}$, where $\texttt{i} \in [0..4]$. On the upsampling side, skip connections allow the transfer of raw information from layers on the downsampling side. This operation is performed through concatenation of the output of convolution layers on the downsampling side to features produced on upsampled input: $\texttt{5x [concat(upsample2x2(previous), output(conv3x3}_{\texttt{i}}\texttt{)), 2x [64*2}^{\texttt{4-i}} \times \texttt{conv3x3 + relu activation]]}$. The receptive field of the original U-Net architecture is 95 pixels.

The original U-net architecture, with the 5 steps of 2 convolutions on the downscale path can be seen in Figure 2. For regularization, `Dropout` layers are placed between the two 3 × 3 convolutions in each block. The output layer has size $N \times N \times K$ where K is the number of classes.

3.2. Fine grained U-Net architecture (FG-UNET)

The proposed network, called FG-UNET, is represented in figure 3. It is inspired from the classical U-Net architecture with some adaptations.

3.2.1. Receptive field and network size adaptation

On 10m resolution images, features such as trees, roads and buildings are only a few pixels wide. Moreover, at this resolution, texture details for classes such as vineyards, forests and industrial areas is fine-grained, with sub-pixel frequencies up to several pixels. Thus, the original receptive field is not adapted, and having a 5 step deep network increases the number of parameters that need to be learned. The depth of the original U-Net network was thus unnecessary in this application. A more shallow

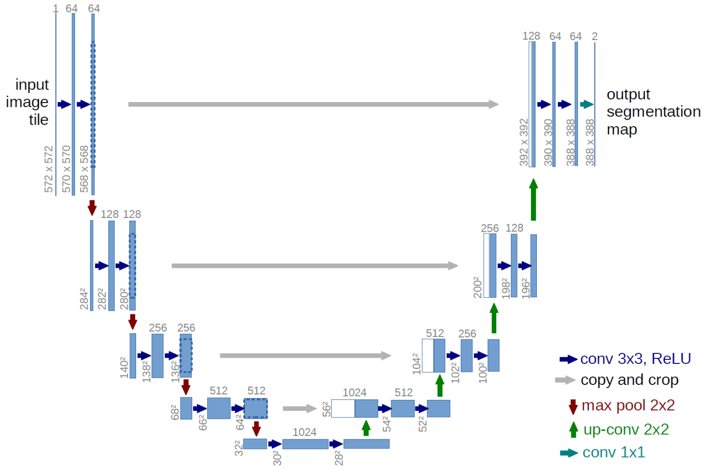


Figure 2. The original U-Net architecture [13]

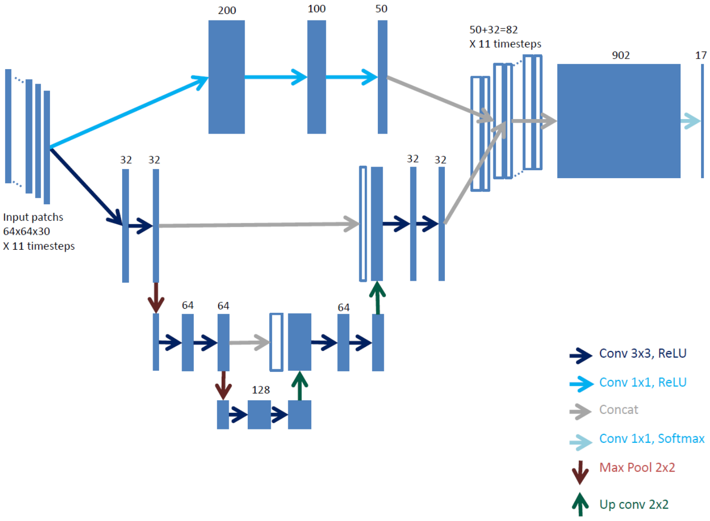


Figure 3. The proposed FG-UNET architecture

U-Net, with 3 steps, was preferred in order to limit overfitting and computation time. The resulting receptive field is 23 pixels. Moreover, we reduced the number of filters, using 32 filters on the first and last layers of the network and adapting the number of filters on the other layers in consequence. Also, instead of retaining only the valid part of the image at the output of a convolution, the whole image is kept and the borders are trimmed at the output of the network. This simplifies the workflow in the prediction step.

3.2.2. Pixel-wise 1x1 path

As the state of the art shows, some classes can be very well classified by pixel-wise classifiers. Moreover, during training, the receptive field of the network will cover both non annotated and annotated pixels. When only one or few non-central pixels in the receptive field are annotated, the weights of the network will be modified only based on the gradient produced on those pixels. We anticipate that this will induce smooth and blurry boundaries between classes during inference. To address this issue, and to add a low-parameter count pixel-wise classifier in our network, we propose to add a convolutional 1x1 pixel-wise classifier in parallel with the convolution based one. Thus, the network is augmented with a cascade of three layers of 200, 100 and 50 1x1 convolutional filters whose output is fused with the output of the 3x3 filters.

3.2.3. Time series handling

The data used for this study are 33 Sentinel-2 images with 10 spectral channels for the entire year 2016. Two approaches of network design to handle this kind of data are possible: early fusion and late fusion. Early fusion supposes that the first layer takes as input $N \times N$ patches with $C \times T$ channels, where $C = 10$ spectral channels per patch and $T = 33$ temporal samples. Thus, a fusion of all these acquisitions at the beginning of the network would drastically increase the number of parameters in the first layers, where back-propagated gradients have low values and high levels of noise.

The method chosen is a late fusion. The input is broken up into G temporal groups containing T/G temporal samples of C channels. The network takes as input one temporal group and outputs an intermediate representation. This network is duplicated G times, with parameter sharing. These intermediate representations are then concatenated, and input to a linear classification layer to obtain the final prediction.

In our case, $G = 11$ and $T = 33$ so each group has 30 channels. This value for G is a satisfactory trade-off between accuracy and learning time. Figure 3 shows the input layer of the FG-UNET architecture with the 11 temporal groups of 30 channel patches that are formed.

The late fusion approach also enables time series processing with LSTM layers. We aim to evaluate the performance gain of using a convolutional implementation of LSTM layers, ConvLSTM2D [25], with respect to simple concatenation of the intermediate representations.

3.2.4. Sparse and imbalanced annotations handling

In order to reduce the influence of class imbalance, the loss function chosen is weighted categorical cross entropy [26]. The formula for this function, for one patch, is given in eq. 1.

$$\mathcal{L} = \sum_{pixels} \sum_{k=1}^{class\_nb} weight_k \cdot y_{true_k} \cdot \log\left(y_{pred_k}\right) \tag{1}$$

We encode the ground truth using either one-hot vectors of size K for annotated pixels, or vectors of zeros of size K for unlabeled ones. These vectors are packed into patches of size $N \times N \times K$. This representation is well suited to sparse labels, because unlabeled pixels will have zero loss and are therefore not taken into account.

3.2.5. Spectral indices

Classical spectral indices can be added as extra channels in the training patches. Some examples are:

- Normalized Difference Vegetation Index (NDVI), computed using Sentinel-2 spectral bands B4 and B8;
- Normalized Difference Water Index (NDWI), computed using Sentinel-2 spectral bands B3 and B8;
- Brightness, as the mean of the ten Sentinel-2 available spectral bands.

We expect that our method and, in general, non-linear classification methods do not benefit in any significant manner from using these indices, since the relations they express can be learned from the data. We perform experiments with and without these indices to verify this hypothesis.

3.3. Training method

To efficiently learn the proposed FG-UNET model, a training method was set up in order to take into account the specificities of satellite images, of the reference ground-truth data and of our HPC infrastructure. Indeed each Sentinel-2 tile used in this study has a size around 80GB (16 bit encoding, 33 dates, 10 spectral bands, at 10 m resolution over 110 km × 110 km) which is much bigger than current images used in deep learning applications, labels used are very sparse and class labels are strongly imbalanced. Our HPC cluster provides a large number of compute nodes with many CPU cores and 120GB of memory per node.

3.3.1. Preprocessing

For faster convergence, CNN models need input data that has normalized values, stored as floating point values. Some works normalize by subtracting the mean and dividing by the standard deviation of each channel. Another approach, used here in order to normalize patches used during learning phase, is to compute the extreme values for each band. We use 3rd and 97th percentiles computed across the whole dataset instead of minimum and maximum values, due to outliers or missing pixels (no data due to orthorectification) potentially present in images. In this way, for each pixel $(i, j) \in N \times N$ of each channel $c \in [0..329]$, each channel c of input patches p is normalized in the interval [0,1]:

$$p'_c(i,j) = clip\left(\frac{p_c(i,j) - 3^{rd}pct(c)}{97^{th}pct(c) - 3^{rd}pct(c)}, 0, 1\right) \tag{2}$$

Reference data is available as polygons. They are split in train and test sets at the polygon level in order to avoid that two pixels from the same polygon are used one for training and the other for validation. These polygons are projected onto a raster to create class index images of the same size as the satellite image tiles but with only one integer channel containing the class (land cover type) index. A zero index signifies a non annotated pixel.

3.3.2. Distributed patch generation

Deep learning methods take advantage of large amounts of data better than classical machine learning. Thus, in our case, we aim to learn land cover classification models from multiple tiles. On one hand, extracting and storing all patches from all tiles before the learning phase is not conceivable due to I/O limitations, thus patch generation must be performed on the fly. Yet, due to tile size (80GB) and available memory on each node (120GB), one node cannot load more than one tile. Consequently, tile loading is distributed among several nodes.

As described in 2.1, input images are composed of 10 spectral bands and 33 time steps. We take $N = 64$ thus a patch size of 64x64 pixels is provided to the network, with $C = 330$ bands.

Furthermore, as annotations are sparse, we want to eliminate patches that do not contain sufficient annotated pixels. We set the threshold of annotated pixels at 10%. We precompute the valid patch positions for each tile to speed up patch choice.

At each training iteration we generate a batch of patches. However, some tiles do not contain some of the classes: mountain tiles do not contain beaches and vineyards while tiles covering plains do not contain glaciers. Moreover, coastal tiles contain large amounts of only one class: water. To keep a balance between different classes, patches must be sampled carefully across different tiles.

Our solution is to have a centralized patch sampling mechanism, and for each node, to extract the patches that correspond to the tile it has loaded and then to communicate these patches to other nodes so that all the nodes have equal workloads. The full algorithm is given in section 3.3.6.

3.3.3. Hyperparameters for distributed training

Deep Learning frameworks are efficient in performing computations on large batches of images. Usually the size of the batch is set to the maximum for which the training network graph fits in memory. In our case this is $b = 8$ patches. Distributed training must thus use these batch sizes across all nodes, giving an effective batch size which for our network is $B = \#nodes \times b$.

As remarked in other works [27], training with large batches requires using a learning rate warm up period. We use the Adam optimizer [28] with a base learning rate $lr = 0.001$. For the first $w = 5$ epochs the learning rate is modulated by a warm up factor:

$$lr'_i = \frac{lr}{\#nodes}\left(\frac{\#nodes - 1}{w} i + 1\right) \tag{3}$$

where $i \in [0..w]$.

3.3.4. Class imbalance handling

The frequency of the different classes in the Sentinel 2 dataset is heavily imbalanced, showing a ratio of 1 to 100 between the least and most frequent classes, as illustrated in table 1. We compute class weights, used in equation 2 as follows:

$$weight_k = \frac{n_{tot}}{n_k} \tag{4}$$

Furthermore, during training, instead of choosing a random patch, we first randomly choose a class before choosing a random patch for this class. This increases the number of patches containing minority classes that are used for training.

3.3.5. Data augmentation

We use data augmentation to introduce learned invariances in the model. We apply rotation (90°, 180°or 270°) and mirroring (horizontal and vertical). This is particularly useful for classes with very few annotations to avoid always giving the same patches to the network.

3.3.6. Complete batch creation procedure

Initially all nodes load one satellite tile. To produce one batch during training the patch generation procedure, executed by all compute nodes, is the following:

1. $B = \#nodes \times b$ examples of a randomly chosen class are sampled
2. B patches containing at least 10% of pixels of the respective class are randomly chosen
3. the worker $n_i = 0$ (first node):
    - determine which nodes have loaded the tile of each of the B patches
    - sets up the patch transfer scheme between nodes: t_{ij}. Say a node n_i must fill c_i patches in the batch. If $c_i > b$, n_i must send $c_i - b$ patches to another node. If $c_i <= b$, n_i must receive

$b - c_i$ patches for another node. We set t_{ij} as the number of patches node n_i must send or receive to nodes $j \in [0..\#nodes]$ so that in the end it has b patches

4. each node n_i:
   - receives the transfer scheme t_{ij}
   - $\forall j \in [0..\#nodes]$, if $t_{ij} < 0$ node n_i sends $-t_{ij}$ patches to node j. Else, it receives t_{ij} patches from node j
5. if spectral indices are enabled, they are computed and additional channels are added to the patches
6. data augmentation is applied to the patches.

3.4. Classification method

The proposed FG-UNET model is fully convolutionnal. Thus, the classification can be performed on images of any size, as long as they can fit in memory. In this work, we chose to split input Sentinel-2 tiles into patches of size 512 × 512 pixels.

However, U-Net classification can present poor performances near border of images. This is due to the fact that padded convolution can introduce a bias in the border of the output image. To overcome this issue, patches are extracted from input tiles with an overlap of 16 pixels.

Once each of these patches is classified, the output classification map of the tile is built.

4. Experimental setup

4.1. Experimental configuration

To test our method, the hardware environment is an HPC infrastructure: a cluster on a low latency network (Infiniband) with shared GPFS storage. Each node has 24 CPU cores and 120GB of memory. In total, there are 252 compute nodes available for a theoretical maximum of more than 6000 CPU cores. Our software environment is composed of CentOS 7 operating system, Python 3.5 as programming language, the high-level neural networks API Keras with TensorFlow back end, and Horovod for a distributed training. Our code is publicly available online[1].

4.2. Baselines

As mentioned previously, an operational processing chain based on the approach described in [8] is currently used to produce land cover maps. It uses Random Forests (RF), a state-of-the-art remote sensing image classifier. We use the same RF implementation to assess convolutional neural networks (CNN) method. It is provided by OpenCV through the supervised classification framework of Orfeo Toolbox 6.2.

Parameters used are 100 trees in the forest, a maximum depth of the tree of 20 and at least 10 samples in each node. For noise reduction, a regularization of generated map is performed using a structuring element with a radius of one pixel. This RF method is applied to the same dataset as the CNN approach.

We introduce several other baselines:

1. **MLP 3 layers**: the 1x1 3 layer convolutional path of the FG-UNET model by itself. Time series handling is however done in an early fusion manner, the input layer thus having 330 channels.
2. **LSTM 3 layers**: the 1x1 3 layer convolutional path of FG-UNET, using input channel grouping (11 groups of 3 channels) and late fusion through an LSTM layer instead of concatenation followed by linear classification.

[1] DOI 10.5281/zenodo.1474224
https://github.com/vpoughon/RT_DL_OSO_public

Method	Parameters	Learning time	Learning time on 1 CPU
RF	-	25h	25h
MLP 3 layers	40 717	7h	1 760h
LSTM 3 layers	324 024	140h	36 960h
U-Net 3 steps	485 297	7h	1 760h
FG-UNET	525 997	13h	3 300h
TSFG-UNET	4 137 272	237h	62 700h

Table 2. Parameter count and training time comparison

3. **U-Net 3 steps**: the U-Net part of FG-UNET by itself, with late fusion through concatenation and linear classification.
4. **TSFG-UNET**: We replace convolutions by convolutional LSTM layers.

In table 2 we compare several useful training quantities:

1. *number of parameters* for each convolutional neural network tested. CNN parameter numbers can not be compared with RF, whose parameters are chosen in a discrete set of values.
2. *learning times* for a learning performed on 11 nodes containing 24 CPUs each
3. *learning time without parallel computing*. For CNN methods it corresponds to the previous column multiplied by 11 nodes and 24 CPUs. It can be compared to RF learning which is directly performed on one CPU, because the implementation used does not allow parallelism.

CNN methods are trained for 50 epochs, a value chosen experimentally, beyond which the loss did not decrease further.

The ability to work on a CPU HPC cluster is important for the application. However, a training was performed on a GPU for comparison. For the FG-UNET method, the learning time was 8h on a GPU.

4.3. Metrics and their limitations

The statistical comparison presented here uses standard metrics: Cohen's Kappa coefficient as an overall measure and the F-Score coefficient (the harmonic mean between precision and recall). Although the use of Cohen's Kappa can be argued when comparing different data sets [29] we use it here to compare different methods on the same data sets.

However, in the operational context, these metrics have some limitations. In the validation data set used in these experiments (similar to [8]), pixels located in the center of spatially coherent areas such as agricultural plots or city blocks are far more common than pixels on the edges or in the corners. This implies that global scores like Overall Accuracy, Kappa, F-scores do not fully reflect the quality of the classification. Indeed, the degradation of minority elements like edges and corners can be overshadowed by an improvement in the central areas. However, these negative effects are very visible in the output maps. For example, in figure 7a, the fine details of the harbour area are smoothed out. The aim of this section is to show how this degradation can be quantified through a robust and fast measure that evaluates whether or not a contextual classification method respects the high spatial frequency areas in the image.

If a dense testing data set is available, validation samples can be split into separate categories that focus on the corners, edges, or fine elements, as is done in [30]. However, in this study, the validation data set is sparse, meaning that such an approach can not be used. Furthermore, the reference polygons are eroded to avoid errors due to misregistrations between satellite images and annotations. The idea is therefore to use a pixel-based classification as a validation map, under the assumption that pixel-based methods capture the geometry of high spatial frequency areas of the image. The pixel-based classification map has other issues, such as noise and the incorrect characterization of

doi:10.20944/preprints201906.0270.v1

context dependent classes, but it should better preserve the corners and fine elements in the image. Therefore, it will be used as a base to validate the quality of the geometry of a classification map generated by a contextual method.

The most visible deformation source is the rounding of sharp corners in the image, as shown in figure 7, so the geometrical quality measurement will be based on the preservation of corners in a contextual classification, with respect to a pixel-based classification. This can be achieved by extracting the corners respectively in the pixel-based and contextual classification results, and by seeing how many of these corners match up within a certain spatial radius. The process for extracting the corners in a classification map is as follows. First, the classification output is split into binary maps for each class. Then, a Line Segment Detector, based on the work in [31], generates a set of segments that follow portions of straight lines in the binary maps. In order to find corners between areas of various classes, all of the class segments are then merged. A corner is found if the extremity of one segment is close enough to another segment, and if their angle is smaller than a given threshold.

Once the corners are extracted, the geometrical resemblance of two classification maps can be quantified as the ratio between the number of corners that appear in both images to the number of corners in the reference image. To obtain a robust measurement, ten reference classifications were used, generated by using different subsets of the training data (cross validation). This was done for a pixel-based classification using Random Forest, and for the CNN result.

Results of this geometric precision analysis are presented in section 5.4.

5. Results

In order to assess the proposed FG-UNET network, various experiments are carried out. First, it is compared to the Random Forest baseline. Second, the importance of individual items of the proposed architecture is evaluated. These experiments contain:

- statistical comparison on tiles and on individual classes;
- visual comparison on level of detail of the produced maps.

5.1. FG-UNET vs RF

Statistical results represented by Cohen's Kappa coefficient are represented for each tile on Figure 5. It shows that FG-UNET method has better performances than RF for all tiles except T30TYN. Detailed results for each class and for tile T30TWT are represented on Figure 6. We observe that :

- FG-UNET is similar or better than RF for all classes, except NGL.
- FG-UNET is significantly better for all urban classes (CUF, DUF, ICU, RSF). FG-UNET takes advantage of the use of the texture. Contrary to pixel based methods like RF, convolutionnal networks can take into account the context of the pixel to classify, which is crucial in artificial areas.
- Error bars show that the variability of results is higher for FG-UNET than for RF. FG-UNET and RF learning and classification were performed 5 times, in order to analyze variability of models. The Kappa coefficient remains the same, but for some classes FG-UNET results may vary.

It is interesting to notice in Figure 5 that the more a tile contains artificial areas (see table 1), the better are FG-UNET results. Indeed, tiles T30TYN and T31TGK are located on mountain areas, and T31UDQ (bigger gap between FG-UNET and RF) is located over Paris.

Figure 7 allows a visual comparison on extracts of produced land cover maps (for tile T30TWT). These extracts have a size of 400 × 500 pixels (4km × 5km). The legend is presented on Figure 4. It highlights that:

- Road recall is better with FG-UNET, but with a lower precision
- Level of detail is similar with FG-UNET and RF in rural regions

🟧	Annual Summer Crops (ASC)
🟨	Annual Winter Crops (AWC)
🟩	Broad-leaved Forests (BLF)
⬛	Coniferous Forests (COF)
🟩	Natual Grasslands (NGL)
🟩	Woody Moorlands (WOM)
🟪	Continuous Urban Fabric (CUF)
🟪	Discontinuous Urban Fabric (DUF)
🟪	Industrial and Commercial Units (ICU)
🟦	Road Surfaces (RDF)
🟥	Bare Rock (BRO)
🟧	Beaches, dunes and sand (BDS)
🟦	Water Bodies (WAT)
⬜	Glacier and perpetual snow (GPS)
🟨	Intensive Grasslands (IGL)
🟪	Orchards (ORC)
🟫	Vineyards (VIN)

Figure 4. Colours of the 17 classes

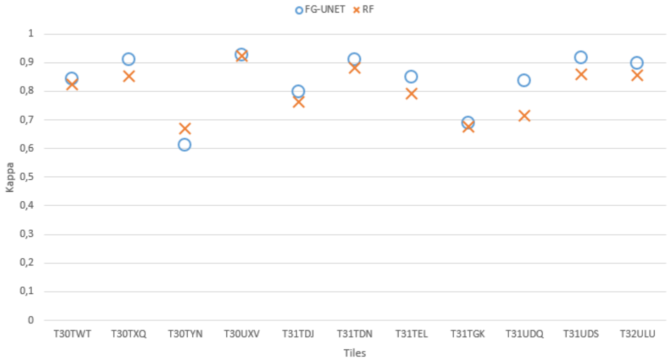


Figure 5. Cohen’s kappa comparison for all tiles between FG-UNET and RF

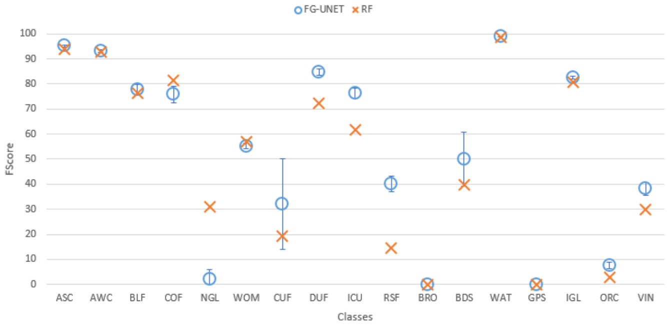
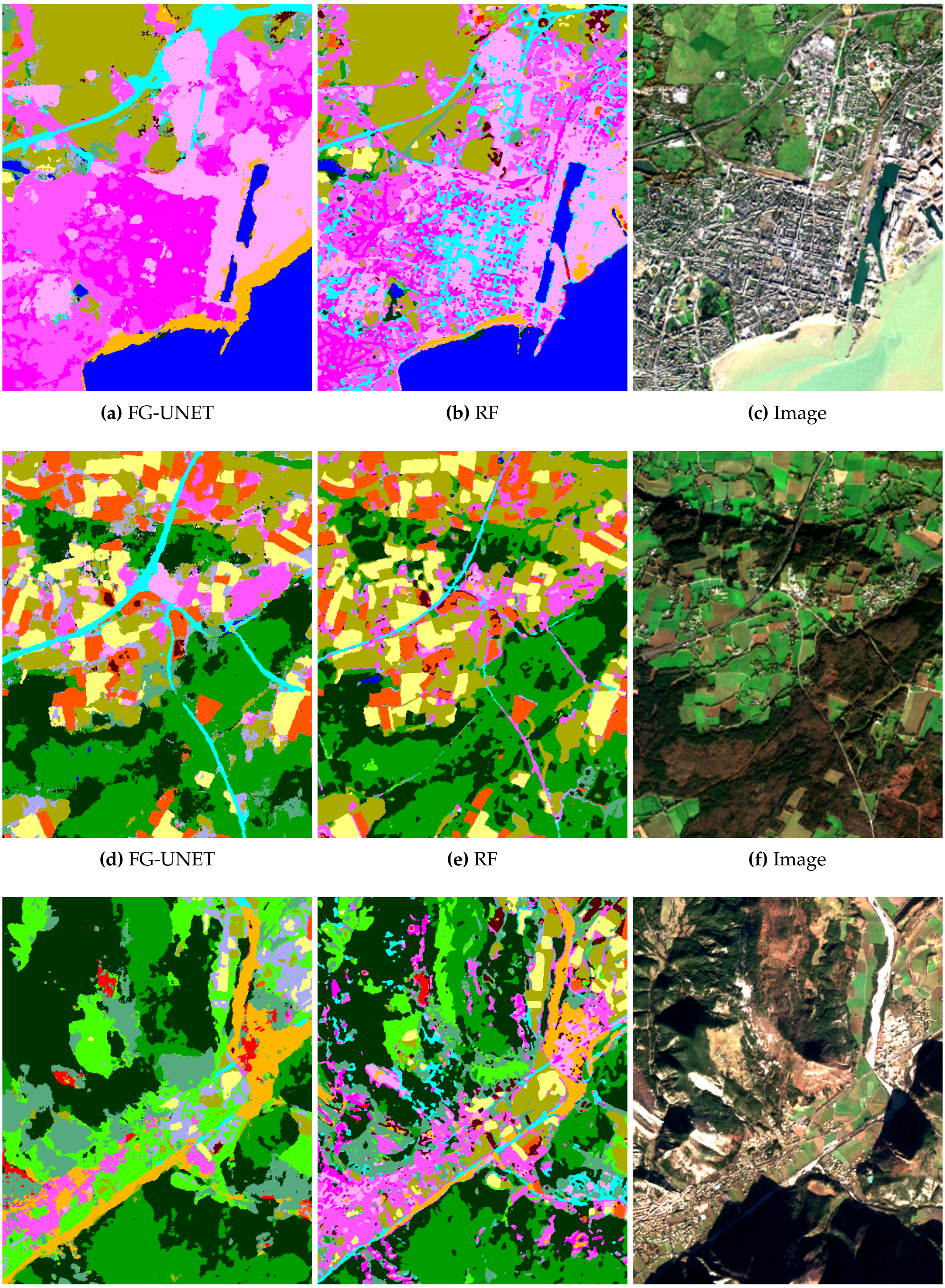


Figure 6. FScore comparison between FG-UNET and RF for tile T30TWT

- Comparison of Figures 7a and 7b highlights that city streets are not classified as roads with FG-UNET, contrary to RF. These streets are not part of the training set, so the expected class is not well defined. However, if some street training samples were part of classes CUF or DUF, FG-UNET with the use of context could be able to classify them correctly, whereas with a pixel based method like RF it would be more challenging.

5.2. Importance of individual FG-UNET features

In order to prove that the proposed FG-UNET is appropriate for this study, tests of alternative networks close to FG-UNET are performed.



(a) FG-UNET **(b)** RF **(c)** Image

(d) FG-UNET **(e)** RF **(f)** Image

(g) FG-UNET **(h)** RF **(i)** Image

Figure 7. Extracts of produced land cover maps with FG-UNET and RF

1. FG-UNET vs UNET. This test was carried out in order to prove the importance of the pixel-wise 1x1 path added to UNET to obtain FG-UNET. Figure 8 shows that FG-UNET gives results slightly

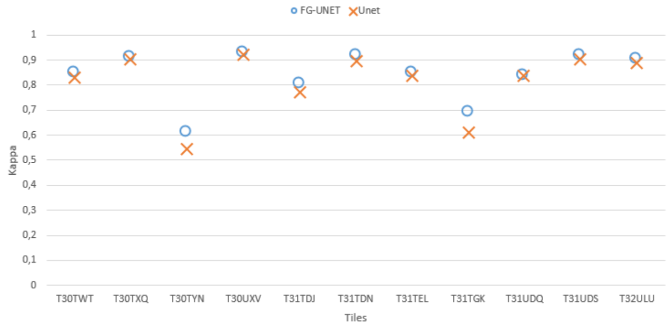


Figure 8. Cohen's kappa comparison for all tiles between FG-UNET and UNET

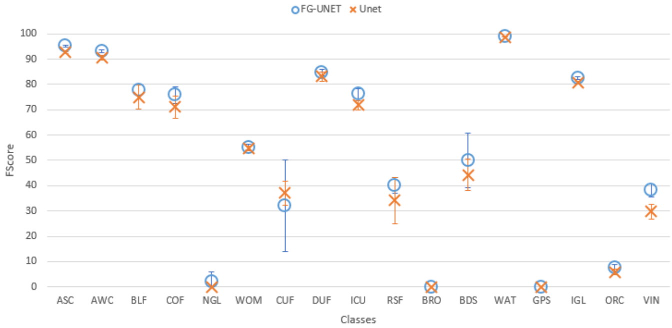


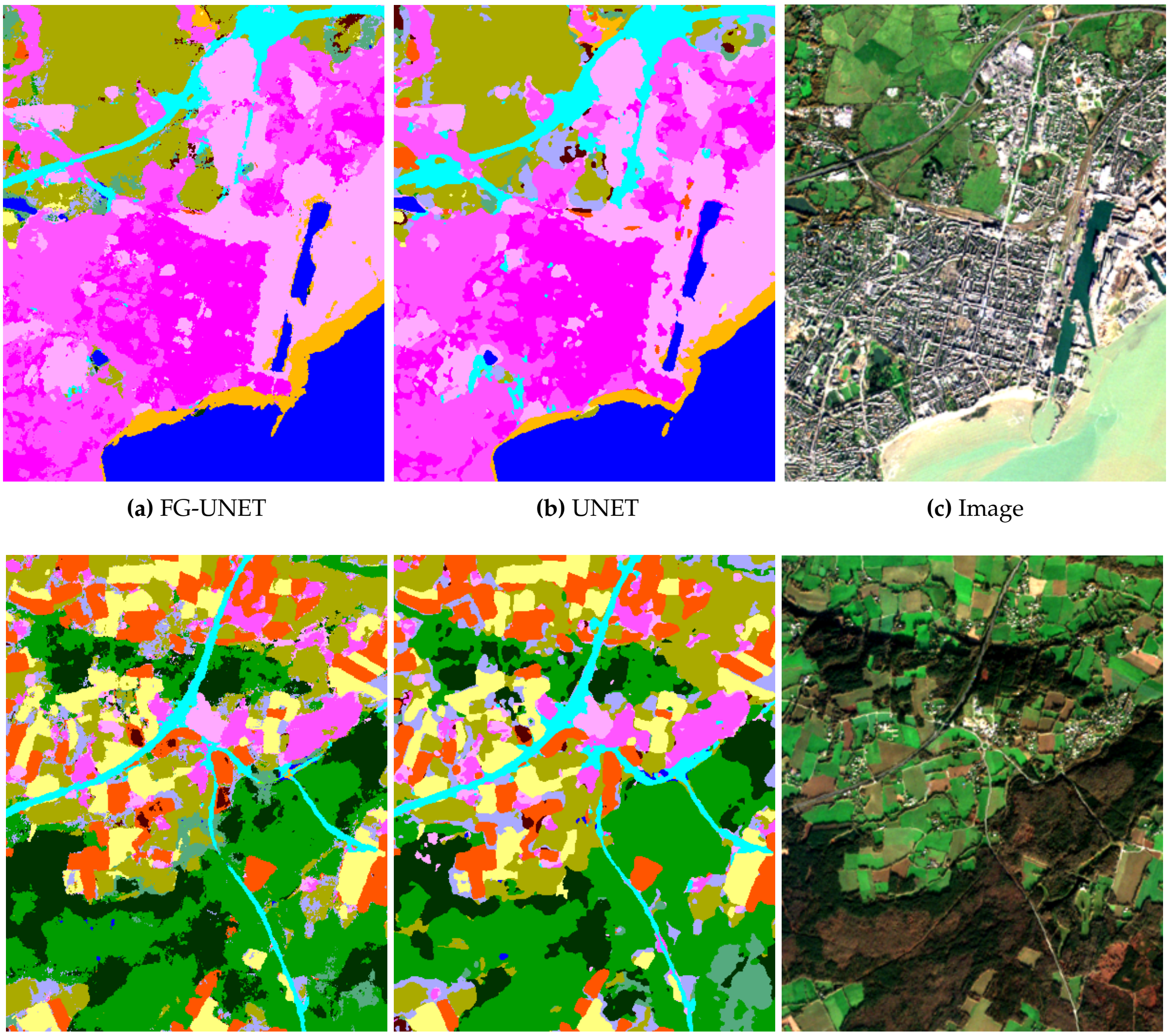
Figure 9. FScore comparison between FG-UNET and UNET for tile T30TWT

above UNET for all tiles. Results are even significantly better for mountain tiles (T30TYN and T31TGK). The detail by class for T30TWT on Figure 9 shows that FG-UNET is better than UNET (except for CUF). Thus, these statistical results prove the importance of pixel-wise 1x1 path contribution. Moreover, visual analysis of produced land cover maps (Figure 10) highlights the increasing in level of detail with FG-UNET. For instance, the jetty of Image 10c is better rendered in 10a than in 10b.

2. FG-UNET vs MLP 3 layers. In this test we compare FG-UNET with a network composed of only the pixel-wise 1x1 path (MLP). It is a pixel based method, without use of context, like RF. Results of MLP are below FG-UNET for all tiles, and all classes on T30TWT as represented on Figure 11 and Figure 12. We can also observe that RF provides better results than MLP with 3 layers.
3. FG-UNET vs LSTM 3 layers. If the temporal aspect is taken into account with a LSTM network, results are better than with a MLP 3 layers. It is still a pixel based approach, and results are very closed to RF baseline, and consequently below FG-UNET, as shown in Figure 13 and Figure 14.
4. FG-UNET vs TSFG-UNET. A test was performed replacing all convolutions of FG-UNET by ConvLSTM2D [25], which is a convolutional implementation of LSTM. Figure 15 and Figure 16 show that results of TSFG-UNET are very close to FG-UNET. Yet, as represented in Table 2 the training time is 19 times higher with ConvLSTM2D. Consequently FG-UNET remains more interesting for this application.
5. Spectral indices usefulness. To assess the usefulness of spectral indices NDVI, NDWI and brightness defined in 3.2.5, a test was performed on the tile T30TWT. We can see on Figure 17 that results for all classes are very close with and without the use of spectral indices. It confirms that information brought by these features can be learned from the data.

5.3. Cross domain transferability

It is interesting to compare the ability of RF and FG-UNET to classify a tile whose training dataset is not used for learning. To do so, for both RF and FG-UNET methods, learning is performed on training datasets of 10 tiles, and the classification is evaluated on the testing dataset of the 11th tile.



(a) FG-UNET **(b)** UNET **(c)** Image

(d) FG-UNET **(e)** UNET **(f)** Image

Figure 10. Extracts of produced land cover maps with FG-UNET and UNET

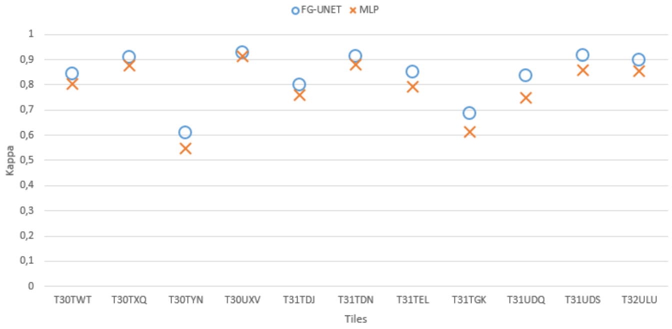


Figure 11. Cohen's kappa comparison for all tiles between FG-UNET and MLP

Results are presented on Figure 18 and compared to the classical results (model learned on training dataset of 11 tiles). We observe on Figure 18:

- a drop in performance when a tile is not part of learning. Yet, it highly depends on the tile content. This drop is particularly high for mountain tiles (T30TYN and T31TGK) whose content is different from other tiles, and for T30TXQ which also has very few AWC labels. On the other hand, their is almost no drop for tile T30UXV.

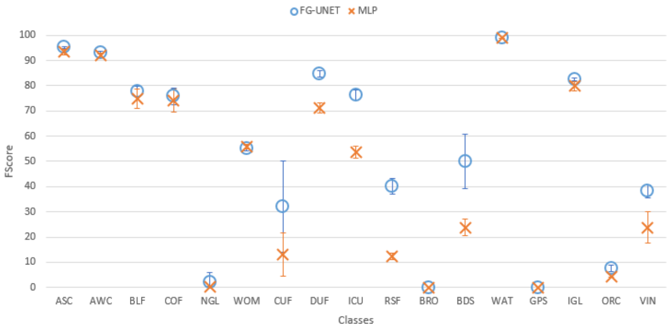


Figure 12. FScore comparison between FG-UNET and MLP for tile T30TWT

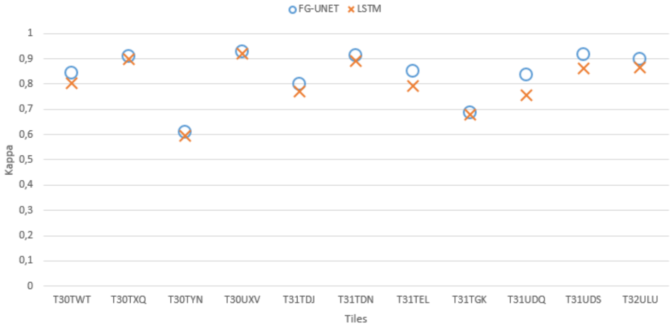


Figure 13. Cohen's kappa comparison for all tiles between FG-UNET and LSTM

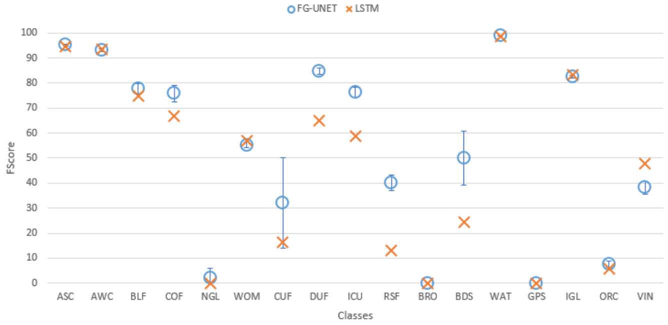


Figure 14. FScore comparison between FG-UNET and LSTM for tile T30TWT

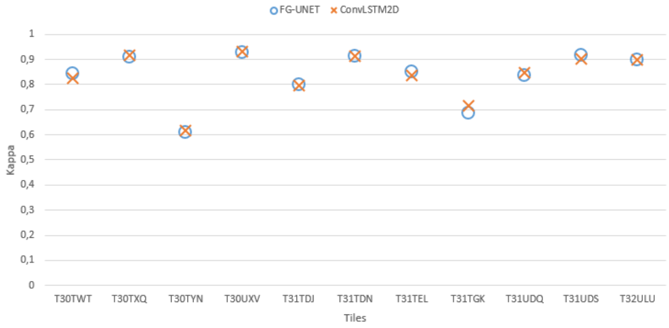


Figure 15. Cohen's kappa comparison for all tiles between FG-UNET and ConvLSTM2D

- FG-UNET suffers from a higher decrease than RF. It tends to show that our implementation of FG-UNET cannot generalize as well as RF.

5.4. Geometric precision

Figure 19 shows the results of the geometric precision ratio as defined in section 4.3, averaged over the ten classifications, along with the standard deviation of the measurement. On all of the different

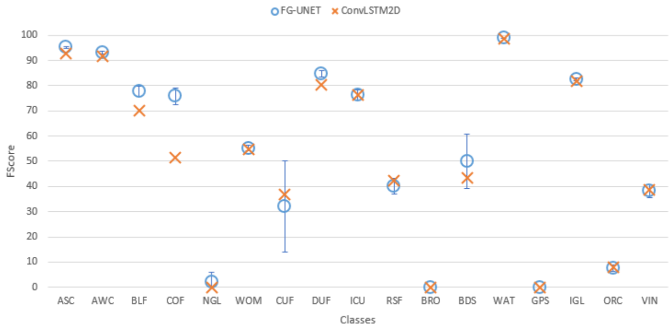


Figure 16. FScore comparison between FG-UNET and ConvLSTM2D for tile T30TWT

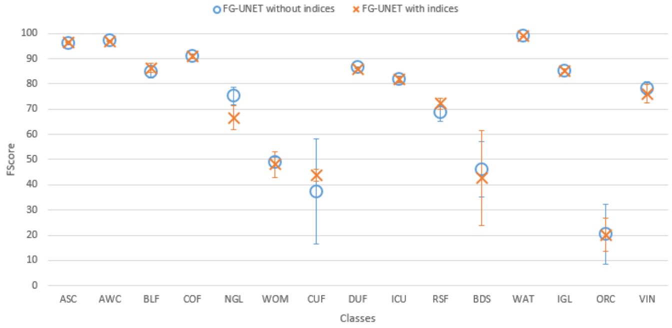


Figure 17. FScore comparison on tile T30TWT for FG-UNET with and without indices NDVI, NDWI and brightness

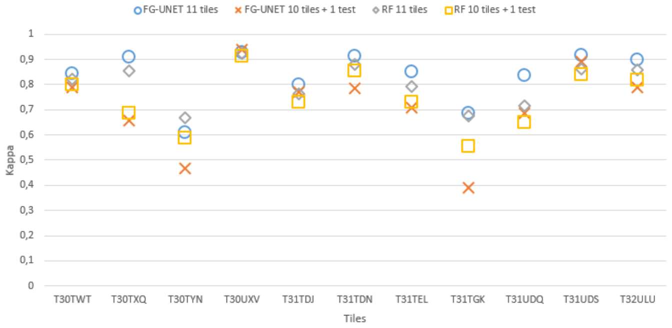


Figure 18. Cohen's kappa comparison for all tiles between test of a model learned on 11 tiles and test on a 11th tiles of a model learned on 10 tiles

tiles, the geometrical accuracy is lower with the FG-Unet than with a pixel-based classifier, Random Forest. This means that some of the corners are not restored properly by the FG-Unet, as was observed in the result maps. Furthermore, this is a recurrent issue, as it appears in different geographical areas, and for different reference images.

6. Conclusion

In this work we studied how to use a fully convolutional deep learning architecture to classify Sentinel-2 image time series at the country scale. Our model, FG-Unet, when trained on all tiles, obtains equivalent or better accuracy than methods that are currently used in production, especially over highly textured area such as urban classes. However this improvement comes with a large computational cost. Without significant engineering effort to reduce the time required to train the model, it is still unclear whether the trade-off is justifiable in a operational context.

This work also confirms the major importance of the quality of the training data. It has a large and measurable effect on the quality of the final land cover map produced.

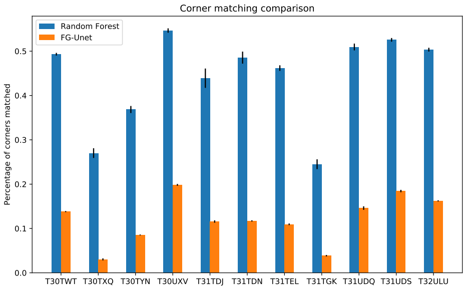


Figure 19. Comparison between geometrical accuracy of a pixel level classification map and a map generated by a FG-Unet. Each one was compared to ten different pixel level classifications, in terms of how well the sharp corners matched. The bars show the average and standard deviation of these measurements. The geometrical accuracy for our FG-Unet result is clearly lower than a pixel-based classifier. This means that there is some deformation in the sharp corners.

Nevertheless, none of those challenges are impossible to solve. Future work following this study could focus, firstly, on reducing over-fitting and increasing generalization across different eco-climatic regions. One approach could be to optimise the architecture of the network through automatic methods such as Differentiable Architecture Search [32]. Secondly, to alleviate the problems of sparse annotations, semi-supervised learning methods such as [33] could be employed.

Furthermore, a more fair comparison with Random Forest should be performed. CNNs naturally capture a pixel spatial context due to their wide receptive field. However random forest is only ever given a single pixel time series as input. This makes the theoretical comparison of their classification performance slightly biased, because the two classifiers are not given the same information per data point. Nonetheless, the comparison presented in this work remains valid as a empirical investigation of the capacities of the two systems in an operational context.

Finally, our code should be benchmarked on a GPU cluster, which would improve training times, although at a high acquisition cost. Our code is available under an open-source license online[2].

Author Contributions: Conceptualization, Andrei Stoian, Vincent Poulain, Jordi Inglada and Victor Poughon; Data curation, Jordi Inglada; Formal analysis, Andrei Stoian and Vincent Poulain; Methodology, Andrei Stoian, Jordi Inglada, Victor Poughon and Dawa Derksen; Project administration, Victor Poughon; Software, Andrei Stoian and Vincent Poulain; Validation, Vincent Poulain and Dawa Derksen; Writing – review editing, Andrei Stoian, Vincent Poulain, Jordi Inglada, Victor Poughon and Dawa Derksen.

Funding: This research received no external funding

Conflicts of Interest: The authors declare no conflict of interest.

1. Heymann, Y. *CORINE land cover: Technical guide*; Office for Official Publ. of the Europ. Communities, 1994.
2. Arino, O.; Gross, D.; Ranera, F.; Leroy, M.; Bicheron, P.; Brockman, C.; Defourny, P.; Vancutsem, C.; Achard, F.; Durieux, L.; Bourg, L.; Latham, J.; Gregorio, A.D.; Witt, R.; Herold, M.; Sambale, J.; Plummer, S.; Weber, J.L. GlobCover: ESA service for global land cover from MERIS. 2007 IEEE International Geoscience and Remote Sensing Symposium, 2007. doi:10.1109/igarss.2007.4423328.
3. Drusch, M.; Bello, U.D.; Carlier, S.; Colin, O.; Fernandez, V.; Gascon, F.; Hoersch, B.; Isola, C.; Laberinti, P.; Martimort, P.; Meygret, A.; Spoto, F.; Sy, O.; Marchese, F.; Bargellini, P. Sentinel-2: ESA's Optical High-Resolution Mission for GMES Operational Services. *Remote Sensing of Environment* **2012**, *120*, 25–36. doi:10.1016/j.rse.2011.11.026.

[2] DOI 10.5281/zenodo.1474224
https://github.com/vpoughon/RT_DL_OSO_public

4. Whitcraft, A.; Becker-Reshef, I.; Killough, B.; Justice, C. Meeting Earth Observation Requirements for Global Agricultural Monitoring: An Evaluation of the Revisit Capabilities of Current and Planned Moderate Resolution Optical Earth Observing Missions. *Remote Sensing* **2015**, *7*, 1482–1503. doi:10.3390/rs70201482.
5. Chen, J.; Chen, J.; Liao, A.; Cao, X.; Chen, L.; Chen, X.; He, C.; Han, G.; Peng, S.; Lu, M.; Zhang, W.; Tong, X.; Mills, J. Global Land Cover Mapping At 30m Resolution: A Pok-Based Operational Approach. *ISPRS Journal of Photogrammetry and Remote Sensing* **2015**, *103*, 7–27. doi:10.1016/j.isprsjprs.2014.09.002.
6. Inglada, J.; Vincent, A.; Thierion, V. Theia Oso Land Cover Map 2106, 2017. doi:10.5281/zenodo.1048161.
7. Inglada, J.; Vincent, A.; Thierion, V. 2017 Metropolitan France Land Cover Map - CESBIO - OSO, 2018. doi:10.5281/zenodo.1993595.
8. Inglada, J.; Vincent, A.; Arias, M.; Tardy, B.; Morin, D.; Rodes, I. Operational High Resolution Land Cover Map Production At the Country Scale Using Satellite Image Time Series. *Remote Sensing* **2017**, *9*, 95. doi:10.3390/rs9010095.
9. Defries, R.S.; Townshend, J.R.G. Global Land Cover Characterization From Satellite Data: From Research To Operational Implementation. GCTE/LUCC Research Review. *Global Ecology and Biogeography* **1999**, *8*, 367–379. doi:10.1046/j.1365-2699.1999.00139.x.
10. Hansen, M.C.; Loveland, T.R. A Review of Large Area Monitoring of Land Cover Change Using Landsat Data. *Remote Sensing of Environment* **2012**, *122*, 66–74. doi:10.1016/j.rse.2011.08.024.
11. Zhu, X.X.; Tuia, D.; Mou, L.; Xia, G.S.; Zhang, L.; Xu, F.; Fraundorfer, F. Deep learning in remote sensing: a comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine* **2017**, *5*, 8–36.
12. Badrinarayanan, V.; Kendall, A.; Cipolla, R. Segnet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *CoRR* **2015**, [arXiv:cs.CV/1511.00561].
13. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. International Conference on Medical image computing and computer-assisted intervention. Springer, 2015, pp. 234–241.
14. Postadjian, T.; Le Bris, A.; Sahbi, H.; Mallet, C. INVESTIGATING THE POTENTIAL OF DEEP NEURAL NETWORKS FOR LARGE-SCALE CLASSIFICATION OF VERY HIGH RESOLUTION SATELLITE IMAGES. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences* **2017**, *IV-1/W1*, 183–190. doi:10.5194/isprs-annals-IV-1-W1-183-2017.
15. Maggiolo, L.; Marcos, D.; Moser, G.; Tuia, D. Improving Maps From Cnns Trained With Sparse, Scribbled Ground Truths Using Fully Connected Crfs. *IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium* **2018**. doi:10.1109/igarss.2018.8517947.
16. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic Image Segmentation With Deep Convolutional Nets, Atrous Convolution, and Fully Connected Crfs. *CoRR* **2016**, [arXiv:cs.CV/1606.00915].
17. Volpi, M.; Tuia, D. Dense Semantic Labeling of Subdecimeter Resolution Images With Convolutional Neural Networks. *IEEE Transactions on Geoscience and Remote Sensing* **2017**, *55*, 881–893.
18. Marmanis, D.; Schindler, K.; Wegner, J.; Galliani, S.; Datcu, M.; Stilla, U. Classification With an Edge: Improving Semantic Image Segmentation With Boundary Detection. *ISPRS Journal of Photogrammetry and Remote Sensing* **2018**, *135*, 158–172. doi:10.1016/j.isprsjprs.2017.11.009.
19. Xu, Y.; Wu, L.; Xie, Z.; Chen, Z. Building Extraction in Very High Resolution Remote Sensing Imagery Using Deep Learning and Guided Filters. *Remote Sensing* **2018**, *10*, 144. doi:10.3390/rs10010144.
20. Maggiori, E.; Tarabalka, Y.; Charpiat, G.; Alliez, P. Convolutional neural networks for large-scale remote-sensing image classification. *IEEE Transactions on Geoscience and Remote Sensing* **2017**, *55*, 645–657.
21. Kussul, N.; Lavreniuk, M.; Skakun, S.; Shelestov, A. Deep learning classification of land cover and crop types using remote sensing data. *IEEE Geoscience and Remote Sensing Letters* **2017**, *14*, 778–782.
22. Ji, S.; Zhang, C.; Xu, A.; Shi, Y.; Duan, Y. 3D Convolutional Neural Networks for Crop Classification with Multi-Temporal Remote Sensing Images. *Remote Sensing* **2018**, *10*, 75.
23. Mou, L.; Bruzzone, L.; Zhu, X.X. Learning spectral-spatial-temporal features via a recurrent convolutional neural network for change detection in multispectral imagery. *arXiv preprint arXiv:1803.02642* **2018**.
24. Hagolle, O.; Sylvander, S.; Huc, M.; Claverie, M.; Clesse, D.; Dechoz, C.; Lonjou, V.; Poulain, V. SPOT4 (Take5): Simulation of Sentinel-2 Time Series on 45 Large sites. *Remote Sensing* **2015**, *7*, 12242–12264. doi:10.3390/rs70912242.

25. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.; Wong, W.; Woo, W. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. *CoRR* **2015**, *abs/1506.04214*, [1506.04214].
26. Sudre, C.H.; Li, W.; Vercauteren, T.; Ourselin, S.; Jorge Cardoso, M. Generalised Dice Overlap As a Deep Learning Loss Function for Highly Unbalanced Segmentations. *Lecture Notes in Computer Science* **2017**, pp. 240–248. doi:10.1007/978-3-319-67558-9_28.
27. Goyal, P.; Dollár, P.; Girshick, R.; Noordhuis, P.; Wesolowski, L.; Kyrola, A.; Tulloch, A.; Jia, Y.; He, K. Accurate, large minibatch SGD: training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677* **2017**.
28. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *CoRR* **2014**, *abs/1412.6980*, [1412.6980].
29. Pontius, R.G.; Millones, M. Death To Kappa: Birth of Quantity Disagreement and Allocation Disagreement for Accuracy Assessment. *International Journal of Remote Sensing* **2011**, *32*, 4407–4429. doi:10.1080/01431161.2011.552923.
30. Bruzzone, L.; Carlin, L. A multilevel context-based system for classification of very high spatial resolution images. *IEEE transactions on Geoscience and Remote Sensing* **2006**, *44*, 2587–2600.
31. Von Gioi, R.G.; Jakubowicz, J.; Morel, J.M.; Randall, G. LSD: a line segment detector. *Image Processing On Line* **2012**, *2*, 35–55.
32. Liu, H.; Simonyan, K.; Yang, Y. Darts: Differentiable architecture search. *arXiv preprint arXiv:1806.09055* **2018**.
33. Tarvainen, A.; Valpola, H. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in Neural Information Processing Systems 30*; Guyon, I.; Luxburg, U.V.; Bengio, S.; Wallach, H.; Fergus, R.; Vishwanathan, S.; Garnett, R., Eds.; Curran Associates, Inc., 2017; pp. 1195–1204.