Graph Neural Networks Extract High-Resolution Cultivated Land Maps from Sentinel-2 Image Series

Lukasz Tulczyjew, Michal Kawulok, Member, IEEE, Nicolas Longépé, Bertrand Le Saux, Senior Member, IEEE, and Jakub Nalepa, Member, IEEE

Abstract—Maintaining farm sustainability through optimizing the agricultural management practices helps build more planetfriendly environment. The emerging satellite missions can acquire multi- and hyperspectral imagery which captures more detailed spectral information concerning the scanned area, hence allows us to benefit from subtle spectral features during the analysis process in agricultural applications. We introduce an approach for extracting 2.5 m cultivated land maps from 10 m Sentinel-2 multispectral image series which benefits from a compact graph convolutional neural network. The experiments indicate that our models not only outperform classical and deep machine learning techniques through delivering higher-quality segmentation maps, but also dramatically reduce the memory footprint when compared to U-Nets (almost 8k trainable parameters of our models, with up to 31M parameters of U-Nets). Such memory frugality is pivotal in the missions which allow us to uplink a model to the AI-powered satellite once it is in orbit, as sending large nets is impossible due to the time constraints.

Index Terms—Sentinel-2 images, temporal analysis, segmentation, land mapping, graph convolutional neural networks.

I. INTRODUCTION

Maintaining farm sustainability by improving the agricultural management practices has become an important issue nowadays to ensure sustainable food security [1], as appropriate management of the cultivated land is paramount for sustaining economic growth [2], and can help us deal with the climate change. As a consequence of the rapid growth of the Earth observation (EO) satellites [3], we are currently able to acquire image data capturing detailed spectral information over large areas, e.g., through the Copernicus Programme, where the Sentinel-2 (S-2) Multi-Spectral Images (MSIs) can be freely accessed. Therefore, the recent advances in EO and artificial intelligence (AI) can play a pivotal role in scalable monitoring of large areas which would not be feasible through the time-consuming and costly in-situ measurements. The S-2 MSIs embrace 13 bands of varying spatial resolution (60 m, 20 m, and 10 m GSD)-extracting information from such imagery has been important in an array of tasks, including environmental monitoring [4], change and object detection, estimating crop production or vegetation analysis thanks to the available spectral information [5]. Such data is, however, highly-dimensional and may be redundant, and extracting the relevant information from it remains an open issue. It is



Fig. 1: Graphical representation of the proposed GCNN for extracting high-resolution cultivated land maps from S-2 image series—as the output, we return the segmented cultivated land map.

especially important for on-board AI in EO, because it can allow us to prune unnecessary data before its transfer. On the other hand, the highest available 10 m GSD bands may still be a limiting factor in some use cases, such as detecting or measuring small land areas. To tackle this issue, we may benefit from single- and multi-image super-resolution (SR) reconstruction algorithms which enhance the spatial image resolution [6]. However, this process should not impact the spectral information [7]. The SR methods range from classical interpolations [8] to data-driven models which learn the correspondence between the low- and high-resolution data [9]. Unfortunately, existing SR solutions are application-agnostic and they are seldom validated for a specific EO task. Finally, MSIs can be acquired for the same area in multiple time points-this temporal aspect opens doors for other use cases, such as monitoring of vegetation or natural disasters [10].

There is a need for bringing AI into space to automate image analysis [11]. This allows us to not only "keep the brain next to the eyes", but also to minimize the amount of data to transfer and accelerate the response time to the monitored events.

LT, MK and JN are with Silesian University of Technology, Gliwice, Poland (e-mail: jnalepa@ieee.org) and with KP Labs, Gliwice, Poland. NL and BLS are with Φ -lab, European Space Agency, Frascati, Italy.

This work was funded by the European Space Agency (the GENESIS project), and supported by the ESA Φ -lab (https://philab.phi.esa.int/). LT, MK and JN were supported by National Science Centre, Poland (2019/35/B/ST6/03006).

While various architectures were exploited for processing such imagery [12], convolutional neural nets (CNNs) are dominant, as they can capture spectral and spatial features. Although CNNs can be optimized for the on-board use [13], they can still be too large to be efficiently transferred to a reconfigurable satellite. Recently, graph convolutional neural nets (GCNNs) that can be resource-frugal and are applicable to irregular data [14] started gaining attention [15]. We follow this research avenue to build lighter yet performant segmentation models.

We introduce an approach for extracting cultivated land maps from S-2 MSIs (Fig. 1). We address the challenges of (i) building the algorithms for accurate segmentation of multitemporal MSIs that can generalize well over the unseen data, and (ii) developing compact deep learning models (Section II). The experiments performed over the real-life data released within the framework of the Enhanced S-2 Agriculture Challenge¹ revealed that our approach outperforms classical and deep learning methods and delivers higher-quality cultivated land maps (Section II-B). Also, we dramatically reduced the memory requirements of our models (quantified as the number of trainable parameters which was reduced up to $3900 \times$ when compared to U-Nets) while obtaining better segmentation, and showed that the GCNN offers fast inference. We made our framework publicly available at https://gitlab.com/jnalepa/ gcnn4sentinel2 to maintain full reproducibility.

II. MATERIALS AND METHODS

In this section, we discuss the dataset, together with the training/test split exploited in the experimental study (Section II-A). Our GCNNs for the cultivated land segmentation from S-2 image series are presented in Section II-B.

A. Dataset Description

The S-2 MSI time-series data was acquired for the growing season (March-September 2019) in the Republic of Slovenia and its neighboring countries within the Enhanced S-2 Agriculture Challenge. The original area is divided into 125 scenes, 100 of which are labeled, whereas for the remaining 25 scenes the labels have not been disclosed. Each scene covers an area of 5×5 km, forming a 500×500 pixels region of interest (at the 10 m GSD) that is paired with the 2000×2000 ground-truth (GT) cultivated land map of a higher resolution with the 2.5 m GSD. Here, the challenge organizers have upscaled all lowerresolution bands (of 20 m and 60 m GSD) to 10 m in the input image stacks. Therefore, the task is to not only segment the cultivated land map from the 12-band S-2 time series (ranging from 19 to 48 geospatially co-registered MSIs; band B10 was removed by the organizers), but also to upsample it to 2.5 m GSD at the same time. The 2.5 m GSD of super-resolved S-2 images was shown to be sufficient to allow for accurate geometrical analysis of small objects and finer descriptions and change detections in many areas, including agriculture [16].

To quantitatively assess the segmentation performance, we focus on the 100 scenes for which the GT is known. We

randomly sampled 20 scenes and included them in the test set Ψ , while the remaining 80 scenes form the training set T. It is worth mentioning that a similar split was used by our team internally during the challenge to locally assess the quality of our techniques, and the relations across the investigated techniques were fairly consistent with those for the 25 scenes with undisclosed GT that were assessed on the server.

B. Proposed Method

Our proposed approach is depicted in Fig. 1, and it can be decomposed into several steps that are performed sequentially. First, the MSIs from an input time series are stacked along the spectral axis, resulting in a variable-length collection of MSIs. Subsequently, the concatenated input is forwarded into an adaptive max pooling layer, which extracts a constant number of temporal and spectral features for each pixel in the scene. This operation allows us to maintain a constant number of fused bands, hence the size of feature vectors for each node in the GCNN is constant as well. Afterwards, the resulting tensor is upsampled utilizing the bicubic interpolation to match the target size (here, the 2.5 m cultivated land map). The GCNN incorporates four hidden layers, consisting of 64, 32, 16 and 8 activations per node, respectively, and the output layer with a single unit (in Fig. 1, we denote this hyperparameter as f). The activation function employed in all hidden layers is the rectified linear unit (ReLU), whereas sigmoid is used in the output layer. Finally, the probability map is thresholded (we use the threshold of 0.5) to obtain the cultivated land maps.

Each node in the graph that becomes the input to GCNN represents a different pixel in the scene (we utilize the 8-connectivity). Our GCNN is a single-parameter model, with only one weight matrix in each layer. Such strategy allows for minimizing the effect of overfitting for graphs with a limited number of labeled nodes. In the cultivated land segmentation task, each node of the input image is labeled (cultivated land vs. background). However, due to the high memory requirements concerned with processing the entire scene at a time, we split it into non-overlapping patches. Therefore, utilizing a single-parameter model should address the problem of overfitting to such spatially-reduced samples.

Each (l + 1)-th layer in GCNN can be given as [17]:

$$H^{l+1} = \operatorname{ReLU}(AH^{l}W^{l}), \tag{1}$$

where H^l and W^l represent the activation and learnable weight matrices of layer l, respectively. In the first layer of the model (l = 1), H^l constitutes the input patch in matrix-based format, where the number of rows is equal to the number of pixels, and columns define the feature vectors of each node. Furthermore, \hat{A} is the normalized adjacency matrix:

$$\hat{A} = D^{-\frac{1}{2}} A D^{-\frac{1}{2}},\tag{2}$$

where D is the diagonal degree matrix calculated by summing over all columns of A in each consecutive row i, and emplacing the value in $d_{i,i}$. We incorporate the self-connections within the graph, hence each node can utilize its own features during the aggregation step. It is achieved by adding the

¹See details at the permanently-open challenge page: https://platform.ai4eo. eu/enhanced-sentinel2-agriculture-permanent (accessed on Mar. 24, 2022).

identity matrix to the adjacency $\widetilde{A} = A + I$, and recomputing the degree matrix from \widetilde{A} . Consequently, we have:

$$\hat{A} = \widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}}.$$
(3)

To control the magnitudes of the activations aggregated for each node, we perform the adjacency matrix normalization. This procedure becomes extremely useful when a vertex shares a lot of connections with other nodes. Such phenomenon would induce large values in their representations, while maintaining lower activations for the "border" vertices, which consequently may lead to the exploding and vanishing gradient problems [18]. When the adjacency matrix is normalized, the representation of each node is calculated as a weighted mean of its neighbors (including itself), which additionally helps improve stability of the training phase. Finally, it is worth mentioning that in the (l+1)-th layer we maintain the (l+1)order of relationship [19]. It means that e.g., in the second layer, the aggregation of features for a node covers not only its neighbors but also the vertices connected to them. It allows us to enhance the contextual information captured by GCNN, hence to detect more abstract spatial and temporal features.

III. EXPERIMENTAL RESULTS

The main objective of our study is to investigate the segmentation capabilities of the proposed GCNN, and to confront it with both the deep learning and classical machine learning techniques for the task of segmenting cultivated land from S-2 image series. For comparison, we took a U-Net model which established the state of the art in a wide range of image segmentation tasks [20], together with a recent long short-term memory (LSTM) network for segmenting S-2 image stacks treated as time series [21], and a random forest (RF) classifier which utilizes both spectral and spatial features, averaged across the temporal dimension (the MSIs are co-registered). The U-Net follows the topology introduced in [20], and we apply batch normalization after each 3×3 convolutional layer in the contracting and expansive paths. Furthermore, we evaluate two versions of this architecture: in U-Net-B, we exploit the bicubic interpolation to spatially upsample the feature maps in the contracting pass. In U-Net-TC, the transposed convolutional layers are used to tackle the intrinsic super-resolution task. For RF, we clean the data by applying the cloud masks for each image. Afterwards, for each pixel, we calculate its spatial and spectral statistical features by investigating its 5×5 neighboring patch (the pixel of interest is the central one). Here, the extraction process is performed within the image stack, and we obtain the minimum, maximum, mean, median, standard deviation, and the 1^{st} , and 3^{rd} quartile of the pixel values, together with the span of values within the patch. The features are extracted for each band separately, therefore the size of the feature vector for each patch is $96 = 8 \cdot 12$, since there are 8 statistical properties and 12 S-2 bands. This approach allowed us to take the 6th place (out of 17 teams) in the Enhanced S-2 Agriculture Challenge.

The investigated approaches were implemented in Python 3.8 with Pytorch 1.10.0 and PyTorch Geometric.

The experiments were run using an NVIDIA Tesla T4 GPU (16 GB VRAM), and all deep models were trained with the same hyperparameters, where the number of input features resulted from the adaptive pooling is equal to 80 (in Fig. 1, we denote it as k). The patch size (PS) is kept constant, and was experimentally set to 100×100 due to the available VRAM (similarly, the RF was trained sequentially for each batch, as the entire training set could not be loaded into the available memory). During the training process, we utilize early stopping, with the maximum number of epochs without improving the loss value (binary cross-entropy) over the validation set V equal to 12. The validation set contains 8 random training scenes (10% of T). The maximum number of epochs was 100. We utilized the Adam optimizer, and the batch size and learning rate were set to 100 and 0.001, respectively.

To evaluate the investigated models, we employ the Fscore, overall accuracy (Acc), and the Matthews correlation coefficient (MCC), with the latter metric being a reliable quality metric for imbalanced classification (it was also the metric used to rank the participants in the challenge). All metrics should be maximized, with one indicating the perfect segmentation. As the input data includes a supplementary mask that indicates which pixels should be excluded from evaluation, we calculate two versions of each metric—over the entire resulting segmentation mask (Full), and with pruning such pixels (Mask). Note that the GT is available for the full scenes, and the latter metric was used during the challenge.

TABLE I: The results obtained over our test S-2 image series (we report standard deviation in the subscripts). The best result is boldfaced, and the second best is underlined.

	MCC		F-Score		Accuracy	
Model↓	Full	Mask	Full	Mask	Full	Mask
RF	$.606_{.222}$	$.621_{.231}$	$.649_{.255}$	$.663_{.263}$.873.071	.873.076
U-Net-B	$.532_{.100}$	$.541_{.102}$	$.627_{.103}$	$.638_{.103}$	$.837_{.060}$	$.834_{.062}$
U-Net-TC	$.634_{.109}$	$.646_{.112}$	$.721_{.096}$	$.734_{.097}$	$.870_{.056}$	$.870_{.056}$
LSTM	$.551_{.180}$	$.559_{.185}$	$.628_{.200}$	$.639_{.204}$	$.851_{.053}$	$.848_{.055}$
GCNN	$.684_{.097}$	$.696_{.098}$	$.760_{.082}$	$.772_{.081}$	$.889_{.051}$	$.889_{.052}$

In Table I, we gather the results obtained for both scoring schemes (with and without masking). We can appreciate that GCNNs significantly outperformed RF, both versions of the U-Nets, and LSTMs, and delivered the highest-quality and the most stable (across the entire test set, as quantified by standard deviation) cultivated areas. The high-quality land segmentation obtained using GCNN is also manifested in Fig. 2, where we render an example test scene segmented using all techniques (we excluded U-Net-B, as it was significantly worse than U-Net-TC), together with the GT land cultivation map (for more examples, see the supplementary material). The GCNN is able to capture subtle characteristics of the region of interest, hence precisely delineate tiny parcels (this may be pivotal in quantifying the area of such fields). GCNNs do not perform any reduction of the spatial features within the network (in contrast to U-Nets), and the edges of their binary maps appear sharper. Furthermore, the U-Net segmentation maps lack finegrained details in those regions. It is visible in Fig. 2(e), and this issue is resolved in GCNN, as shown in the magnified areas in Fig. 2(f). Additionally, this visual example shows the difficulty of the task-the S-2 images within one series can



Fig. 2: Visualization of a frame (a) from an example test S-2 series (ID: #344), the (b) GT cultivated land segmentation with a 2.5 m spatial resolution, and the predicted maps obtained using (c) RF, (d) U-Net-TC, (e) LSTM, and (f) the GCNN (red denotes the false positives, yellow—false negatives). We zoom a part of the image showing fine-grained details of the parcel's boundary that were correctly captured by GCNN. The best quality metrics are boldfaced.

.816

.825

77.8

GCNN

.749

.758

vary due to changing acquisition conditions and cloud covers. Our GCNNs offer prediction faster than other methods (it is confirmed in Table II; the time in Fig. 2 was measured on an Intel i7-8565U CPU, and in Table II, on NVIDIA Tesla T4).

To quantify the memory requirements of the deep models in the context of deploying them in hardware-constrained execution environments, we present the number of their trainable parameters in Table II. We can observe that our GCNN is a compact model, with more than $2180 \times$ and $3916 \times$ less parameters than U-Net-B and U-Net-TC, respectively (note that the model with the transposed convolutional layers instead of a bicubic interpolation incorporates almost two times larger number of weights). This massive difference is reflected in the size of the serialized model (33.88 kB vs. 118.65 MB, 66.09 MB and 225.46 MB for GCNN, U-Net-TC, U-Net-B and RF). It will directly impact the uplink time of such trained models to deploy them to the satellite once it is in orbit. This model update strategy² will be exploited in our Intuition-1 hyperspectral mission. Finally, we can appreciate that our GCNNs are not only compact, but also offer fast inference.

TABLE II: The number of trainable parameters in all deep models, together with the execution time averaged for the test scenes.

Model	Trainable params	Inference time [s]
U-Net-TC	31,081,985	14.8
U-Net-B	17,307,329	14.5
LSTM	331,393	107.9
GCNN	7,937	14.1



Fig. 3: The MCC and binary cross-entropy (BCE) scores obtained over the validation set V by the GCNNs trained with different patch sizes (PS), and by both U-Nets. The varying length of the training curves indicates that the early stopping condition has been met.

To verify the impact of the patch size on the GCNN training, we show MCC and binary cross-entropy obtained for V for all epochs in Fig. 3. PS affects the number of input nodes of GCNN, and—when coupled with the constant batch size directly influences training. We can appreciate that the final MCC and cross-entropy are high for small PS values too, thus our GCNN is robust against a range of values of this hyperparameter and always delivers accurate segmentation. On the other hand, MCC and cross-entropy for both U-Nets indicate that these networks converge faster to the worse models. In the supplement, we can see that changing the bicubic interpolation to other techniques (such as nearestneighbor or Lanczos) has the minimal impact on the training.

Finally, we utilized all models to segment the test data delivered by the challenge organizers—the weighted MCC

²This strategy can allow us to not only replace the model with a new one, perhaps tackling a different EO task, but also to uplink an updated model for the current task (e.g., fine-tuned over real imagery). Thus, we aim at turning our mission into the *flying laboratory*, where we can change its AI operations.

scores³ calculated by the validation server amounted to 0.597, 0.475, 0.544, 0.543, and 0.635 for RF, U-Net-B, U-Net-TC, LSTM and GCNN, respectively (for GCNN with the nearest-neighbor, Lanchos and inter-area interpolations we got 0.623, 0.610, and 0.611). To further improve these results, it would be pivotal to increase the patch size fed to the models (due to the VRAM limitations, we trained and infer over 100×100 patches, as mentioned earlier). For the U-Nets (with 722,615 trainable parameters, roughly $91 \times$ more than GCNN) trained over the full scenes (500×500 pixels) using the GPU server provided by the organizers, we obtained MCC of approximately 0.663. Furthermore, we were able to train a more complex version of U-Net (30,601,221 parameters, $3856 \times$ more than GCNN), incorporating an already upscaled image of size 2000×2000 (we used bicubic interpolation), for which MCC amounted to 0.773. These experiments show that increasing the spatial context could improve the segmentation scores-it constitutes our current efforts for GCNNs [22].

IV. CONCLUSIONS AND FUTURE WORK

We proposed an end-to-end processing pipeline built upon a graph convolutional network to extract precise high-resolution cultivated land segmentation maps from S-2 image series of lower spatial resolution. The experimental results, obtained over a range of scenes containing a varying number of MSIs captured in different time points revealed that our technique significantly outperforms both classical and deep learning models through delivering higher-quality segmentation maps. Additionally, we massively reduced the number of trainable parameters of the deep learning models when compared to the U-Nets (up to more than $3900 \times$). This, in turn, allowed us to dramatically decrease the memory footprint of the segmentation model. Elaborating such resource-frugal deep networks is pivotal in the context of satellite missions which can benefit from uplinking a trained (or fine-tuned) model to the satellite while it is in in-orbit operation, as it directly influences the upload time. Finally, we showed that our GCNN is robust against the image patch size, and lead to precise segmentation for a range of vastly different patch sizes, and it offers fast inference. We believe that the research reported in this letter can become an important step toward deploying compact yet efficient and robust deep models in EO satellites.

We are currently working on using the weighted edges (and heterogeneous graphs) to further exploit the temporal information, and on porting the models to Intuition-1. We focus on not only benchmarking such algorithms [23], but also on verifying their robustness against on-board conditions [24]. Finally, we work on utilizing other interpolation techniques and our multi-image SR algorithms, and to make them taskdriven for precise segmentation of satellite images.

REFERENCES

 K. Spangler, E. Burchfield, and B. Schumacher, "Past and current dynamics of U.S. agricultural land use and policy," *Frontiers in Sust. Food Systems*, vol. 4, 2020.

³The organizers of the challenge report the modified MCC metric which assigns a higher weight to small parcels.

- [2] D. Kpienbaareh, X. Sun, J. Wang, I. Luginaah, R. Bezner Kerr, E. Lupafya, and L. Dakishoni, "Crop type and land cover mapping in northern Malawi using the integration of sentinel-1, sentinel-2, and planetscope satellite data," *Remote Sensing*, vol. 13, no. 4, p. 700, 2021.
- [3] J. Castillo-Navarro, B. Le Saux, A. Boulch, and S. Lefèvre, "Energybased models in earth observation: From generation to semisupervised learning," *IEEE TGRS*, vol. 60, pp. 1–11, 2022.
- [4] E. Grabska, P. Hostert, D. Pflugmacher, and K. Ostapowicz, "Forest stand species mapping using the sentinel-2 time series," *Remote Sensing*, vol. 11, no. 10, p. 1197, 2019.
- [5] M. Karlson, M. Ostwald, J. Bayala, H. R. Bazié, A. S. Ouedraogo, B. Soro, J. Sanou, and H. Reese, "The Potential of Sentinel-2 for Crop Production Estimation in a Smallholder Agroforestry Landscape, Burkina Faso," *Frontiers in Env. Science*, vol. 8, 2020.
- [6] D. Valsesia and E. Magli, "Permutation invariance and uncertainty in multitemporal image super-resolution," *IEEE TGRS*, vol. 60, pp. 1–12, 2022.
- [7] C. Lanaras, J. Bioucas-Dias, S. Galliani, E. Baltsavias, and K. Schindler, "Super-resolution of Sentinel-2 images: Learning a globally applicable deep neural network," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 146, pp. 305–319, 2018.
- [8] S. M. A. Bashir, Y. Wang, M. Khan, and Y. Niu, "A comprehensive review of deep learning-based single image super-resolution," *PeerJ Computer Science*, vol. 7, p. e621, 2021.
- [9] M. Kawulok, P. Benecki, S. Piechaczek, K. Hrynczenko, D. Kostrzewa, and J. Nalepa, "Deep learning for multiple-image super-resolution," *IEEE GRSL*, vol. 17, no. 6, pp. 1062–1066, 2020.
- [10] F. Bioresita, A. Puissant, A. Stumpf, and J.-P. Malet, "Fusion of sentinel-1 and sentinel-2 image time series for permanent and temporary surface water mapping," *IJRS*, vol. 40, no. 23, pp. 9026–9049, 2019.
- [11] G. Giuffrida, L. Fanucci *et al.*, "The *φ*-sat-1 mission: The first on-board deep neural network demonstrator for satellite earth observation," *IEEE TGRS*, vol. 60, pp. 1–14, 2022.
- [12] N. Audebert, B. Le Saux, and S. Lefèvre, "Deep learning for classification of hyperspectral data: A comparative review," *IEEE GRSM*, vol. 7, no. 2, pp. 159–173, June 2019.
- [13] J. Nalepa, M. Antoniak, M. Myller, P. Ribalta Lorenzo, and M. Marcinkiewicz, "Towards resource-frugal deep convolutional neural networks for hyperspectral image segmentation," *Microprocessors and Microsystems*, vol. 73, p. 102994, 2020.
- [14] J. Zhou, G. Cui, S. Hu, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *AI Open*, vol. 1, pp. 57–81, 2020.
- [15] T. Tarasiewicz, J. Nalepa, and M. Kawulok, "A graph neural network for multiple-image super-resolution," in *Proc. IEEE ICIP*, 2021, pp. 1824– 1828.
- [16] N. Latte and P. Lejeune, "PlanetScope Radiometric Normalization and Sentinel-2 Super-Resolution (2.5 m): A Straightforward Spectral-Spatial Fusion of Multi-Satellite Multi-Sensor Images Using Residual Convolutional Neural Networks," *Remote Sensing*, vol. 12, no. 15, 2020.
- [17] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," arXiv preprint arXiv:1609.02907, 2016.
- [18] B. Hanin, "Which neural net architectures give rise to exploding and vanishing gradients?" *Proc. NIPS*, vol. 31, 2018.
- [19] H. Le, T. Tran, and S. Venkatesh, "Self-attentive associative memory," in *Proc. ICML*, vol. 119, 2020, pp. 5682–5691.
- [20] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Proc. MICCAI*. Springer, 2015, pp. 234–241.
- [21] H. Zhao, S. Duan, J. Liu, L. Sun, and L. Reymondin, "Evaluation of Five Deep Learning Models for Crop Type Mapping Using Sentinel-2 Time Series Images with Missing Information," *Remote Sensing*, vol. 13, no. 14, 2021.
- [22] A. Zhou, J. Yang, Y. Gao, T. Qiao, Y. Qi, X. Wang, Y. Chen, P. Dai, W. Zhao, and C. Hu, "Brief industry paper: optimizing memory efficiency of graph neural networks on edge computing platforms," in *Proc. IEEE RTAS*, 2021, pp. 445–448.
- [23] M. Ziaja, P. Bosowski, M. Myller, G. Gajoch, M. Gumiela, J. Protich, K. Borda, D. Jayaraman, R. Dividino, and J. Nalepa, "Benchmarking deep learning for on-board space applications," *Remote Sensing*, vol. 13, no. 19, 2021.
- [24] J. Nalepa, M. Myller, M. Cwiek, L. Zak, T. Lakota, L. Tulczyjew, and M. Kawulok, "Towards on-board hyperspectral satellite image segmentation: Understanding robustness of deep learning through simulating acquisition conditions," *Remote Sensing*, vol. 13, no. 8, 2021.

Graph Neural Networks Extract High-Resolution Cultivated Land Maps from Sentinel-2 Image Series (Supplementary Material)

Lukasz Tulczyjew, Michal Kawulok, Nicolas Longépé, Bertrand Le Saux, Jakub Nalepa jnalepa@ieee.org

This supplementary material collects the training curves obtained for the proposed graph neural network models coupled with a range of interpolation techniques, together with the example Sentinel-2 scenes segmented using the proposed technique (Section 1).

1 Detailed Experimental Results

We present the training curves elaborated for our graph neural networks coupled with various interpolation techniques used to obtain the 2.5 m Sentinel-2 images (Figure 1), to-gether with the example Sentinel-2 image stacks segmented using our technique (Figure 2).



Figure 1: The MCC and binary cross-entropy (BCE) scores obtained over the validation set V by the GCNNs trained with patch size equal to 100, and different types of image interpolation. The varying length of the training curves indicates that the early stopping condition has been met.



Figure 2: Example nine-image Sentinel-2 time-series stacks segmented using the proposed graph neural networks. For each scene, we present an example image from the stack (a, c, e, g), together with the corresponding cultivated land map (some parts of the maps have been zoomed for clarity).