Land cover mapping from medium-resolution SAR and multi-spectral remote sensing images

Ivan Dubrovin [1] **Shakir Sofi** [1] **Veronika Shirokova** [1] **Ilya Barskiy** [1] **Evgeny Avdotin** [1] **Kundyz Onlabek** [1] **Arina Ivanova** [1] **Aleksandr Gamayunov** [1]

Abstract

The project is dedicated to land cover mapping, using SAR and multi-spectral remote sensing images from the SEN12MS dataset. Several models, including ResNet, DenseNet, U-Net and DeepLabV3+, were explored and implemented for land cover segmentation task. Different approaches, involving denoising of sensing images, utilizing of different bands, application of topological loss, were elaborated to improve the segmentation quality.

Github repo: link to GitHub repository
Video presentation: link to video

1. Introduction

Computer vision usually deals with conventional photographs of everyday objects, whereas remote sensing data is more versatile and much more difficult to interpret. Yet, land cover maps represent spatial information on different types (classes) of physical coverage of the Earth's surface, e.g. forests, grasslands, croplands, lakes, wetlands (CGL). Dynamic land cover maps include transitions of land cover classes over time and hence captures land cover changes. Land use maps contain spatial information on the arrangements, activities and inputs people undertake in a certain land cover type to produce, change or maintain it.

Sentinel missions data used in the research provide free data with global coverage of less than 12 days in most places on Earth, which can be used to monitor a large spectum of different events in near real time. Some examples of applications: assessing damage during disasters like floods/forest fires/earthquakes; monitoring forest degradation/recovery; mapping urban footprint growth; monitoring crop life-cycles etc.

Image segmentation in general is defined as the process of partitioning an image into multiple segments. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Efficient image segmentation is one of the most critical tasks in computer vision and has been interpreted differently for different applications. In remote sensing, it is often viewed as an aid to landscape change detection and land use/cover classification, which state that image segmentation is present in every kind of image analysis. This constitutes the motivation to our work.

2. Related works

With the paper (Schmitt et al., 2019), the authors publish the SEN12MS dataset, which contains 180,662 triplets of Sentinel-1 dual-polarimetric SAR data, Sentinel-2 multispectral images, and MODIS-derived land cover maps. In order to provide an example for the usefulness of the dataset with regard to the development of land cover classification solutions, they trained two deep convolutional neural network architectures for predicting LCCS land use classes: ResNet-110 (He et al., 2016) and DenseNet (Jégou et al., 2017).

The ResNet architecture was designed to mitigate the problem of vanishing gradients, which tended to appear for very deep CNNs before. This is realized by the introduction of so-called shortcut connections, i.e. instead of learning a direct mapping from input to output layer, the shortcut connection skips one or more layers by passing the original input through the network without modifications. Then, the network learns residual mappings with respect to this input.

The DenseNet architecture is based on the finding that CNNs can be deeper, more accurate and efficient to train if they contain shorter connections between layers close to the input and layers close to the output. Thus, DenseNets directly connect each layer to every other layer in order to ensure maximum information flow between layers in the network. Different to ResNets, the features are not combined through summation; instead, they are concatenated.

[*] Equal contribution [1] Skolkovo Institute of Science and Technology, Moscow, Russia. Correspondence to: Ivan Dubrovin <Ivan.Dubrovin@skoltech.ru>.

Final Projects of the Deep Learning 2021 Course, Skoltech, Moscow, Russian Federation, 2021.

However, the conclusion about the comparison of the two models performance cannot be done as the authors used different images for tests.

In paper (Yu et al., 2020), authors proposed segmentation model based on DeepLabV3+. The model involved preprocessing of Sentinel-1 images, augmentation of Sentinel-2 images, label refinement and DeepLabV3+ implementation. The first step implied speckle noise removal with Enhanced Lee Filter, which managed noisy edge boundaries by edge directed windows. The main augmentation techniques included random flip, rotation, warp and contrast stretch. Each one was applied with probability of 0.5. Label refinement process dealt with the fact of incorrect labels in Moderate Resolution Imaging Spectroradiometer (MODIS) data. The authors inserted module, which updates the label every 5 epochs. As the backbone in DeepLabV3+, ResNet-101 (pretrained on ImageNet) was used. SEN12MS dataset was utilized as training set. Validation and testing sets were performed by DFC2020 dataset. The main merit of the proposed segmentation model was significant improvement in average class (from 41.12% to 50.18%), pixel-wise (from 49.93% to 62.57%) and most of in-class (by 15% max) accuracies. The limitations included tendency to misclassify cropland, wetland and grassland. Also, authors mentioned that the model wasn't trained for the sufficient number of epochs due to computational capacity.

In the paper (Schmitt et al., 2020), the authors investigated several models for the task of segmentation on the SEN12MS dataset. They applied U-Net, DeepLabV3+ as segmentation architectures and k-means and Random Forest as classification models with different input sets (only Sentinel-2 and Sentinel-1 + Sentinel-2). Authors utilized the SEN12MS as training set and DFC2020 dataset as validation and test sets. The best results in terms of average accuracy (average accuracy among in-class accuracies) were obtained for the U-Net (48.1%) and DeepLabV3+ (44.2%) models for both input sets. This work can be considered as the state-of-art one.

2.1. Denoising

Speckle, the grainy salt and pepper like noise, is inherent to all SAR images. Authors of the dataset have not done any filtering to address it, so Sentinel-1 images are noisy. It was decided to look for additional experiments dedicated to removing noise from images. Hence, the method of denoising the convolutional neural networks that was described in (Zhang et al., 2020) was used. The DnCNN experiment results in (Adrian et al., 2021) confirmed that the denoising architecture produces favorable image denoising and optimized run times.

In our work, getting rid of noise is only the intermediate stage of the experiments, therefore, it was decided to use a pre-trained neural network for this step, the details of which can be found at (Deep Plug-and-Play Image Restoration).

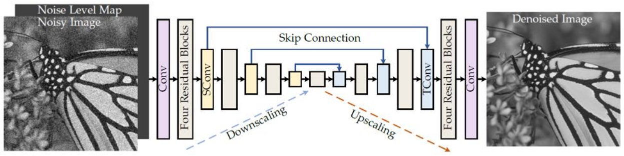

Figure 1. The architecture of the DRUNet denoiser network. DRUNet takes an additional noise level map as input and combines U-Net and ResNet. "SConv" and "TConv" represent strided convolution and transposed convolution, respectively.

The architecture of DRUNet is shown on figure 1. This solution uses the advantage of the half-quadratic splitting algorithm, the iterative optimization of three different image restoration tasks, including deblurring, super-resolution, and color image demosaicing. It consists of alternately solving a data subproblem which has a closed-form solution and a prior subproblem which can be replaced by a deep denoiser. The results in (Zhang et al., 2020) revealed that plug-and-play image restoration with powerful deep denoiser prior have several pluses. Firstly, it boosts the effectiveness of the model-based methods due to the implicit but powerful prior modeling of deep denoiser. Secondly, without task-specific training, it is more flexible than learning based methods while having comparable performance. Overall, the proposed plug-and-play image restoration with deep denoiser prior not only significantly outperforms other state-of-the-art model-based methods but also achieves competitive or even superior performance against state-of-the-art learning-based methods.

2.2. Topological image segmentation

Segmentation algorithms are prone to topological errors on fine-scale structures, e.g., broken connections. In (Vandaele & Gevaert, 2020) the authors propose a novel method based on Topological Data Analysis (TDA) consisting of Topological Image Modification (TIM) and Topological Image Processing (TIP). Through this newly introduced method, they artificially destructed irrelevant objects, and constructed new objects with known topological properties in irrelevant regions of an image. This ensures that they are able to identify the important objects in relevant regions of the image. It is done by the means of persistent homology (Hu et al., 2019), which allows to simultaneously select appropriate thresholds, as well as the objects corresponding to these thresholds, and separate them from the noisy background of an image. This leads to a new image, processed in a completely unsupervised manner, from which one may more efficiently extract important objects.

3. Object and Methodology

The object of this study is the SEN12MS dataset. It is one of the largest datasets, containing full multi-spectral information in geocoded imagery. It was created for the needs of the remote sensing community. Moreover, it covers all regions of the Earth over all meteorological seasons, while most of the other datasets are restricted to fairly small areas.

3.1. Dataset description

The SEN12MS dataset contains 180662 patch triplets (Sentinel-1 dual-pol SAR, Sentinel-2 multi-spectral, MODIS land cover) in the form of multi-channel GeoTiff images (Schmitt et al., 2019). It is available under the open access license CC-BY at the library of the Technical University of Munich (Schmitt & Hughes, 2019). The dataset is based on randomly sampled regions of interest, resulting from four different seed values (1158, 1868, 1970 and 2017), which correspond to the four meteorological seasons defined for the northern hemisphere (spring, summer, fall and autumn, respectively). The Sentinel-1 data can be recognized by the abbreviation `s1`, the Sentinel-2 data by `s2`, and the MODIS land cover data by `lc`. The file naming follows the scheme: `ROIsSSSS_SEASON_DD_pXXX.tif`, where `SSSS` denotes the seed value, `SEASON` denotes the meteorological season as defined for the northern hemisphere, `DD` denotes the data identifier, and `XXX` denotes the patch identifier (Schmitt et al., 2019). The image size is 256×256 pixels.

MODIS data is used as labels for training. It contains four bands - classes according to the International Geosphere-Biosphere Programme (IGBP) classification scheme, the land cover classification system (LCCS) land cover (LC) layer, the LCCS land use (LU) layer, and the LCCS surface hydrology (SH) layer. In (Schmitt et al., 2019) authors used LCCS LU values for classification and segmentation models training. In other works of (Schmitt et al., 2020; Schmitt & Wu, 2021; Yu et al., 2020) IGBM values were used as ground-truth labels. The example of the dataset triplet (Sentinel-1, Sentinel-2 and four MODIS bands) is presented on Figure 2.

In this project we use simplified IGBP values as labels. The classes are presented in Table 3.2.

3.2. Data exploration

The original SEN12MS dataset consists four different schemes of MODIS derived land cover labels. From those schemes, the International Geosphere-Biosphere Program (IGBP) scheme was chosen as background for the conversion into a classification dataset. This was done because the IGBP scheme features rather generic classes, including both natural and urban environments with a moderate level of semantic granularity. The Land Cover Classification System (LCCS) in contrast is less generic and over-focus on different topics of interest. As already proposed by (Robinson, 2021), the 17 original IGBP classes were simplified to the scheme described in Table 1 to mitigate the class imbalance of SEN12MS to some extent.

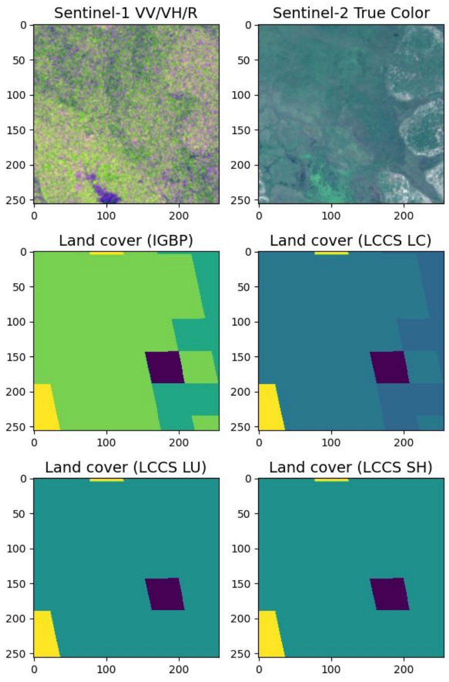

Figure 2. Random patch triplet from the dataset, related to SPRING season

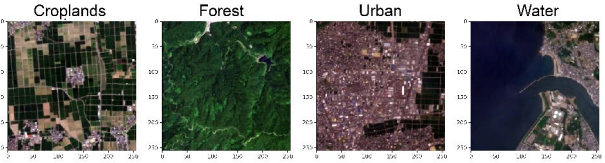

Figure 3. Sentinel-2 (RGB) samples from the SEN12MS dataset, including simplified IGBP land cover annotations.

3.3. Denoising

We decided to carry out additional experiments dedicated to removing noise from images. For this task, we used the existing method of denoising convolutional neural networks described above.

We have worked out two options for noise removal. In the first case, denoising occurred separately for each channel by applying grayscale version of DRUNet, that you can see

IGBP Class Number	IGBP Class Name	Simplified Class Number	Simplified Class Name	Description	Color
1	Evergreen Needleleaf Forest	1	Forest	Lands covered by woody vegetation at >60% and height exceeding 2 m	#009900
2	Evergreen Broadleaf Forest				
3	Deciduous Needleleaf Forest				
4	Deciduous Broadleaf Forest				
5	Mixed Forest				
6	Closed Shrublands	2	Shrublands	Lands with shrub canopy cover >10% and <2 m tall	#c6b044
7	Open Shrublands				
8	Woody Savannas	3	Savanna	Lands with understory systems, and with forest cover between 10% and 60% and >2m tall	#fbfd7e
9	Savanna				
10	Grasslands	4	Grassland	Herbaceous lands with <10% trees/shrubs	#b6ff05
11	Permanent Wetlands	5	Wetlands	Lands with a permanent mixture of water and herbaceous or woody vegetation	#27ff87
12	Croplands	6	Croplands	Lands covered with temporary crops followed by harvest and a bare soil period	#c24f44
14	Cropland/Natural Vegetation Mosaics				
13	Urban and Built-up Lands	7	Urban/Built-up	Lands covered by buildings and other man-made strutures	#a5a5a5
15	Permanent Snow and Ice	8	Snow/Ice	Lands under snow/ice cover throughout the year	#69fff8
16	Barren	9	Barren	Lands with exposed soil, sand, rocks	#f2f3b1
17	Water Bodies	10	Water	Oceans, seas, lakes, reservoirs and rivers	#1c0dff

Table 1. The simplified IGBP land cover classification scheme.

on the center images from Figure 4. In the second case, it occurred for two channels in RGB format simultaneously by applying color version of DRUNet - right images on Figure 4. Both cases looks better than the original one, but in the second case there are cleaner water areas. Finally, we enrich the original dataset with two additional layers in sentinel-1 for making experiments with it.

3.4. Models

Dense Convolutional Network (DenseNet), which connects each layer to every other layer in a feed-forward way. In contrast to traditional convolutional networks with L layers have L connections - one between each layer and its subsequent layer this network has L(L+1)/2 direct connections. For each layer, the feature-maps of all preceding layers are used as inputs, and its own feature-maps are used as inputs into all subsequent layers.

The U-Net network is a classical architecture for a semantic segmentation problem. It has a symmetric fully convolutional architecture. The U-Net was applied to land cover mapping in works (Stoian et al., 2019; Schmitt et al., 2020).

DeepLabV3+ architecture shown on Figure 6 was used. That network allow to use pre-trained backbone part to better understand image fiches and relation between them by applying aspp part. We used architecture realised by Lukas and Jonathan in DeepLab repo with small modification from original paper (Chen et al., 2018). We took that realisation and configure they for our case. Pre-trained on ImageNet ResNet101 used as backbone with modified first convolution layer according to different size of input channels.

4. Experiments

4.1. ResNet and DenseNet: paper replication

The authors of the dataset used in our research (Schmitt et al., 2019) proposed two main models for land cover classification solutions. ResNet-110 and DenseNet.

In table 2, we can see characteristics of nets that was used. However, after preliminary exploration of the models we found out that ResNet is much more inefficient in computational point than other models we described above in Section 3.4.

For the presented baseline experiment with DenseNet, we cut images of 256 × 256 pixels from the Sentinel-2 samples from the summer subset, and used the following ten bands as channel information: B2 (Blue), B3 (Green), B4 (Red), B8 (Near-infrared), B5 (Red Edge 1), B6 (Red Edge 2), B7 (Red Edge 3), B8a (Red Edge 4), B11 (Short-wavelength infrared 1), and B12 (Short-wavelength infrared 2). We used the fully convolutional DenseNet for semantic segmentation,

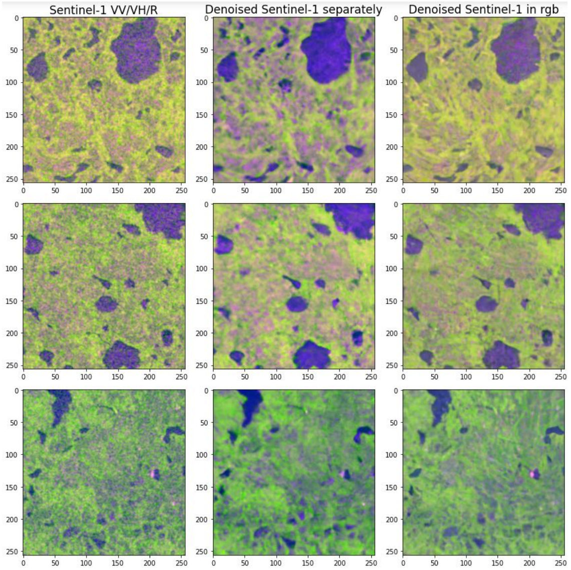

Figure 4. Comparison of two denoising settings with original images from sentinel-1

aiming at assigning a class label to every pixel of the input image. The result is also shown in figure and the overall accuracy metrics. The evaluation is based on manually labeled reference points.

As we can see on 7 and 8 predictions have sharp boundaries. However, for "Water" class we got much more accurate boundary detection.

	ResNet-110	DenseNet
Upsampling type	n/a	Deconvolution
Input size	64 x 64 x 10	256 x 256 x 10
Batch size	16	4
Loss	Categorical cross-entropy	
Optimizer	Nesterov Adam	
Initial LR	0.0005	0.0001
LR schedule	ReduceOnPlateau	

Table 2. Description of models used in paper related to the original dataset

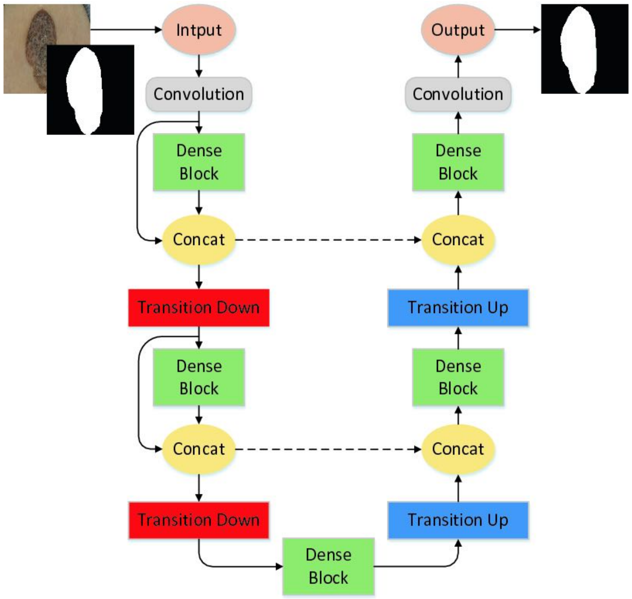

Figure 5. DenseNet architecture

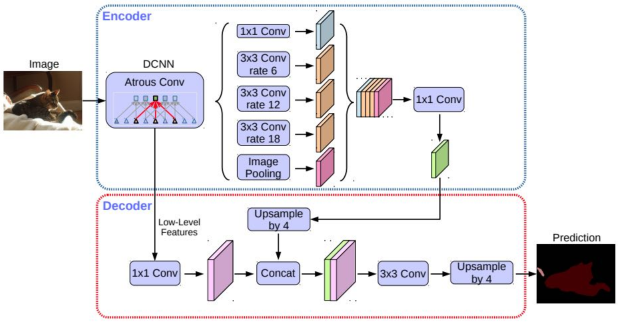

Figure 6. DeepLabv3+ architecture

4.2. U-Net

In this project, it was decided to investigate U-Net on the dataset sample (4 seasons, 4 scenes, 3157 patches) for several cases of input images: 1) original Sentinel-1 + original Sentinel-2 with 3 bands; 2) original Sentinel-1 + Sentinel-2 with 3 bands and non-destructive transformations; 3) filtered Sentinel-1 + Sentinel-2 with 3 bands and non-destructive transformations; 4) original Sentinel-1 + original Sentinel-2 with 10 bands; 5) original Sentinel-1 + Sentinel-2 with 10 bands and non-destructive transformations; 6) filtered Sentinel-1 + Sentinel-2 with 10 bands and non-destructive transformations.

The dataset sample was divided into training (75%), validation (15%) and test (10%) sets. The training and validation procedures were run for 20 epochs with a batch size of 2.

For the cases 1-3, we used RGB bands (B04, B03, B02) of Sentinel-2. For the cases 4-6, we used 10 bands (B02, B03, B04, B05, B06, B07, B08, B08a, B11, B12) of Sentinel-2. The idea was to figure out the optimal number of input channels.

As non-destructive transformations we applied horizontal flip, vertical flip and rotation by 90 with probability of 0.5 to Sentinel-2 and MODIS during training. Such an option was proposed in the work (Yu et al., 2020). We applied Lee

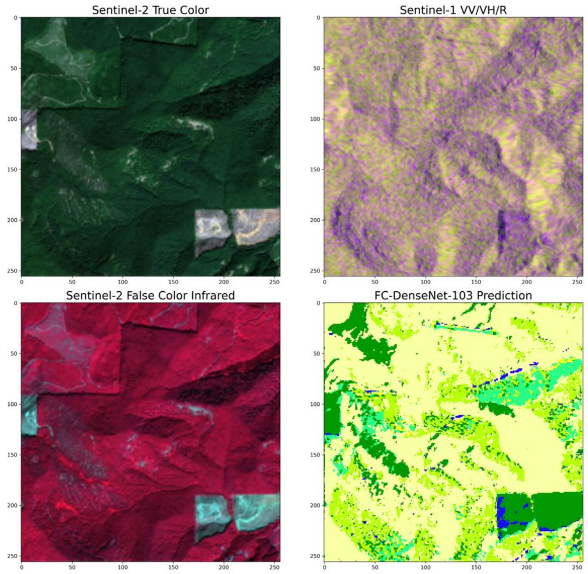

Figure 7. Comapison of the Sentinel-1 and Sentinel-2 data with the DenseNet results

filter to handle speckles of Sentinel-1, as proposed in the work (Yu et al., 2020).

As a loss function, categorical cross-entropy was used. As optimizer we utilized Adam optimizer with initial learning rate of 0.0001 and weight decay of 0.0001. To reduce learning rate dynamically, we applied ReduceLROnPlateau scheduler with a factor of 0.5 and patience of 4.

4.3. DeepLabV3+

We use DeepLabV3+ architecture mentioned before. For comparison with baseline results, the data preparation and experiment was constructed equal to described above in section 4.1. 10 input layers are used for only sentinel-2 images of 256×256 pixels and 12 input layers are used for second experiment with added denoised images from sentinel-1.

4.4. Topological image segmentation

Firstly, we tried to use the same dataset mentioned in (Masoud et al., 2020), but we were not successful in dealing with the huge amount of data due to the small computational resources. Hence, the dataset from (Hu et al., 2019) was used with the same idea of segmenting the boundaries in cells. We used popular UNet model for boundary segmentation.

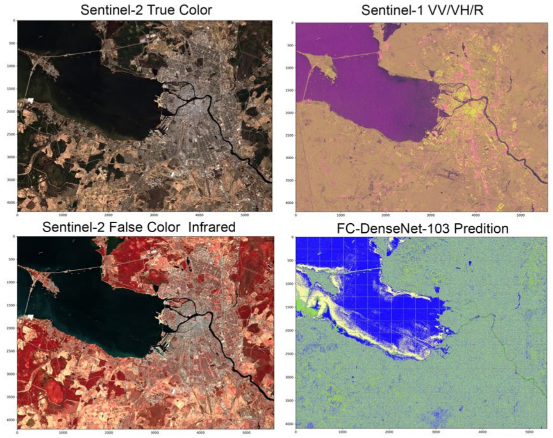

Figure 8. Comapison of the Sentinel-1 and Sentinel-2 data with Water label with the DenseNet results

5. Results

5.1. U-Net experiments

To estimate the performance of U-Net model for different cases it was decided to calculate overall accuracy (number of correctly predicted labels per total number of predicted labels). The detailed description of the cases one can observe in section 4.2. The OA values are presented in Table 3. They correspond to the best model parameters, related to the minimal loss on validation set. One can mention, that the best test accuracies are obtained for the 4th and 6th cases. In these experiments we used 10 bands of Sentinel-2 and 2 bands of Sentinel-1. However, one can also mention, that the minimal val-test gap corresponds to the 6th case. In such a way, it can be preliminary said, that the best input combination for U-Net model is filtered Sentinel-1 + 10 bands of Sentinel-2 with transformations.

Table 3. Overall accuracies (OA) for U-Net experiments

Case	Train set, %	Validation set, %	Test set, %
1	23.41	15.25	18.1
2	22.99	15.27	13.7
3	22.24	21.24	14.2
4	22.39	12.84	21.6
5	23.32	22.11	16.8
6	23.47	20.98	21.7

Now let’s try to estimate the predictive quality of the U-Net model for the 6th case. In Figure 9 some examples from the test set are presented. It can be mentioned that the results are rather controversial. On the one hand, comparing with the true labels, the model predicts poorly. On the other hand, if one compare Sentinel-1 and prediction, it can be noticed, that the model tends to comply with it. So, here we face the problem of MODIS IGBP values confidence.

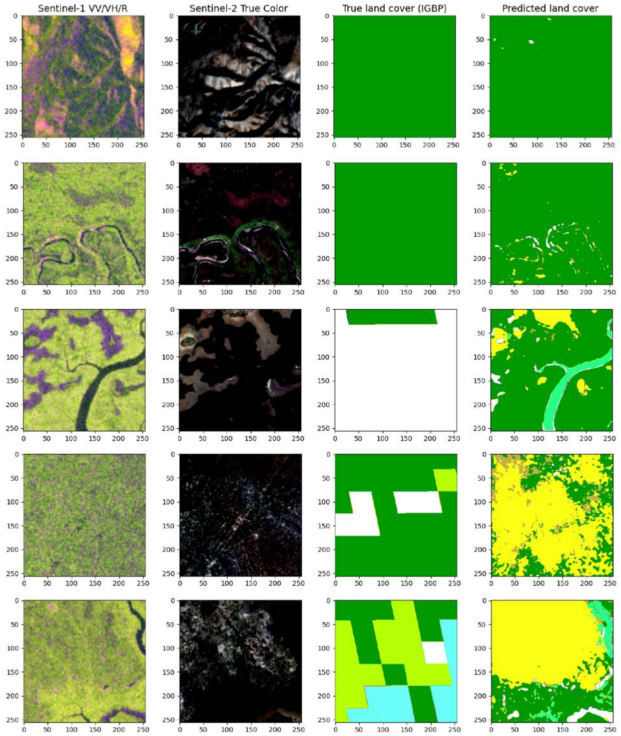

Figure 9. Visual analysis of segmentation results of the U-Net model for the 6th case

5.2. DenseNet experiments

DenseNet was the most promising experiment of all that we tried. However, the results are quite bad in general, the structure of the image is intact, including even small details considering that the labels are very noisy because of the 500 meter resolution. We assume that if we had more time and computing power available, we could get better results. Also we can mention that the denoised versions should become noticeably better than plain ones (table 4). This model in our case was more memory efficient and fast to apply than all others.

Experiment	OA St. Petersburg	OA Rome	OA Munich
FC-DenseNet-103	26.13 %	0%	0.44%
FC-DenseNet-103 + de-noising	0.65%	0.13%	0.29%
DeepLab	0.12%	0%	0.038%
DeepLab + de-noising	33.27 %	0.06%	0.84%

Table 4. Test accuracy for the main experiments

The results of St. Petersburg is much better compared to images of Rome and Munich. The reason is that a huge part of the image is covered by water, which is the easiest class to distinguish from all others. Even the lower accuracies reported in the St. Petersburg column of table 4 are only lower because the class for the huge water body is incorrect. The shape of the body and even the details (bridge, ships) are noticeable in all images. We believe that if we had time to train all our models longer, the level of details and the overall resolution would stay the same or only increase, while the classes would shift to being closer to ground truth. The images of Rome and Munich do not have such great standing out features (even visually, they are hard to interpret). The prediction for them looks like a prediction for the land part of the St. Petersburg image, so the accuracy is close to 0.

5.3. Topological image segmentation

In this experiment, instead of feeding the images directly to the segmentation model, we have processed (smoothening, Border modification with Persistent Homology) the input images topologically, so that modified image contains only relevant topological information. The results can be observed on Figures 10 and 11 from which we can conclude the results with topological prediction are comparable better. Accuracy is increased from 0.900 to 0.908, Betti number error is also reduced from 0.72 to 0.20

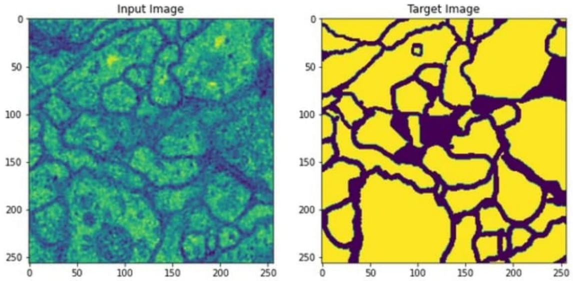

Figure 10. Input image and the ground truth segmentation

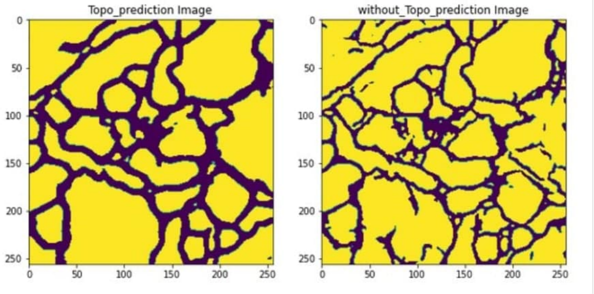

Figure 11. Comparison of the results with and without topological models

6. Conclusions

We conducted 6 experiments with the U-Net model. According to the values of computed overall accuracies, the best input set was defined as filtered Sentinel-1 + 10 bands of Sentinel-2 with non-destructive transformations. However, it should be underlined, that such an input consumed significant amount of memory. According to the visual analysis of the segmentation results for the selected input set, it should be mentioned, that the reliability of the MODIS IGBP values is ambiguous. The model tended to correspond to Sentinel images. So, the main recommendation for further investigations is to utilize more accurate labels (maybe even manually assigned).

Overall, it is obvious that our results are not even close to any existing state-of-the-art and baseline results in the field of semantic segmentation of remote sensing images (one example baseline is IEEE GRSS Data Fusion Contest 2020). That is mainly because we severely underestimated the time it takes to train a collection of semantic segmentation models from zero. We still, however, believe that we have arrived at some important insights into the topic during the time we spent preparing and running our experiments and analyzing the results. The first self-evident result is that even though the labels are of significantly lower resolution compared to the images, the nature of convolutional neural networks makes the resulting segmentation maps match the input in resolution. We have come to an understanding that fully convolutional networks adapted for semantic segmentation are the weapon to choose when tackling the problem of semantic segmentation of remote sensing images. The part of our team that is involved in the space segment will continue building on these insights and results in future research.

References

Copernicus global land service: Land cover.

Adrian, J., Sagan, V., and Maimaitijiang, M. Sentinel sar-optical fusion for crop type mapping using deep learning and google earth engine. *ISPRS Journal of Photogrammetry and Remote Sensing*, 175:215–235, 2021.

Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., and Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. pp. 801–818, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. pp. 770–778, 2016.

Hu, X., Li, F., Dimitris, S., and Chao, C. Topology-preserving deep image segmentation. *arXiv preprint arXiv:1906.05404*, 2019.

Jégou, S., Drozdzal, M., Vazquez, D., Romero, A., and Bengio, Y. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. pp. 11–19, 2017.

Masoud, K. M., Persello, C., and Tolpekin, V. A. Delineation of agricultural field boundaries from sentinel-2 images using a novel super-resolution contour detector based on fully convolutional networks. *Remote sensing*, 2020.

Robinson, Malkin, J. C. Q. S. G. H. Y. Global land-cover mapping with weak supervision: Outcome of the 2020 ieee grss data fusion contest. *IEEE JOURNAL OF SELECTED TOPICS IN APPLIED EARTH OBSERVATIONS AND REMOTE SENSING, VOL. 14, 2021*, 2021.

Schmitt, M. and Hughes, L. Sen12ms, 2019.

Schmitt, M. and Wu, Y.-L. Remote sensing image classification with the sen12ms dataset. *arXiv preprint arXiv:2104.00704*, 2021.

Schmitt, M., Hughes, L. H., Qiu, C., and Zhu, X. X. Sen12ms–a curated dataset of georeferenced multi-spectral sentinel-1/2 imagery for deep learning and data fusion. *arXiv preprint arXiv:1906.07789*, 2019.

Schmitt, M., Prexl, J., Ebel, P., Liebel, L., and Zhu, X. X. Weakly supervised semantic segmentation of satellite images for land cover mapping - challenges and opportunities. *CoRR*, abs/2002.08254, 2020.

Stoian, A., Poulain, V., Inglada, J., Poughon, V., and Derksen, D. ”land cover maps production with high resolution satellite image time series and convolutional neural networks: Adaptations and limits for operational systems. *Remote Sensing*, 11(17):1–26, 2019. doi: https://doi.org/10.3390/rs11171986.

Vandaele, Robin, G. A. N. and Gevaert, O. Topological image modification for object detection and topological image processing of skin lesions. *Scientific Reports*, 2020.

Yu, Q., Liu, W., and Li, J. Spatial resolution enhancement of land cover mapping using deep convolutional nets. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B1-2020, 2020. doi: https://doi.org/10.5194/isprs-archives-XLIII-B1-2020-85-2020.

Zhang, K., Li, Y., Zuo, W., Zhang, L., Van Gool, L., and Timofte, R. Plug-and-play image restoration with deep denoiser prior. *arXiv preprint arXiv:2008.13751*, 2020.

A. Team member's contributions

Ivan Dubrovin

- Literature review
- Preparation of the GitHub Repo
- Preparation of the dataset and PyTorch wrappers for the data loader
- Preparation of test images and manually labeled grids for testing
- Coding the DenseNet implementation and experiments
- Coding the final image generation and utility code for dataset visualization
- Running the long-term experiments on a workstation with persistent GPUs
- Preparing a part of the Conclusions section of this report

Veronika Shirokova

- Literature review
- Coding and running U-Net experiments for different input sets
- Preparing report sections 2, 3.1, 4.2, 5.1, 6
- Preparing presentation

Aleksandr Gamayunov

- Literature review
- Preparing the GitHub Repo
- Denoise Sentinel-1 data and enrich the original dataset
- Preparing DeepLab experements
- Preparing the report
- Preparing the presentation

Arina Ivanova

- Literature review
- Data preprocessing
- Preparing the report 2.1, 3.2, 3.3, 3.4, 4.1, 5.2 + editing
- Preparing the presentation

Shakir Sofi

- Literature review about topological segmentation
- Prepared Data processing/Metric and Visulatization of topological segmentation
- Preparing Topological segmentation experement
- Preparing the presentation

Kundyz Onlabek

- Literature review, search for better approaches than baseline
- Preparing the report: introduction, related work, methods (3.3, 3.4), experiments (4.4), results (5.2, 5.3), conclusions; editing the whole report and connecting parts together
- Preparing the presentation

Evgeny Avdotin

- Literature review
- Preparing the report: introduction, related work

Ilya Barskiy

- Literature review on transfer learning
- Attempts to apply transfer learning with DenseNet model

A. Used 3rd-party codes

* DeepLab repo

* Deep Image Restoration

* UNet repo