Deep Learning Segmentation for Improved Land Cover Maps and Estimates

**Keywords:** Land Cover, Satellite Imagery, Sentinel-2, Deep Learning, Convolutional Neural Networks, Semantic Segmentation, U-Net.

# Introduction

Timely and frequently updated Land Cover (LC) information is of paramount importance to modern National Statistical Institutes. The Italian National Institute of Statistics (Istat) is currently investigating whether Deep Learning [1] methods could be used to derive automated LC estimates of satisfactory quality from Sentinel-2 [2] satellite images. A prototype software system is being developed within the scope of this research [3][4]. Some of Istat’s investments in this project have been contributed to Work Package C ‘Earth Observation’ of Eurostat’s ESSnet ‘Big Data II’ [5], and are partly documented in WPC deliverables H1 [6] and H2 [7]. This paper focuses on *‘automated land cover maps’*, a very relevant output artefact of our automatic LC estimation system. Section 2.1 accounts for the foundations and early results of our project, Section 2.2 critically discusses the limitations of our initial attempts, whereas subsequent Sections illustrate ongoing countermeasures and advancements. The main aim of the paper is to show how we succeeded in improving the quality of our automated LC maps by integrating an advanced model for Semantic Segmentation [8] (i.e. a customized U-Net Deep Learning architecture) into our previous processing pipeline. More specifically, the U-Net [9] model helped us solve the overestimation issue that affected our previous approach for linear narrow LC classes, like rivers and highways.

# Methods

## Early design of the system: a classify‑and‑count approach to Land Cover estimation and maps

An automatic LC estimation system should be able to take as input a satellite image depicting a portion of territory, and to return as output a table of LC statistics. Although LC estimation is a quantification problem rather than a classification one, we started implementing our system according to a pure ‘classify‑and‑count’ design. The main driver of this initial design choice was to incorporate into our system a Convolutional Neural Network (CNN) [1], so as to take advantage of its tremendous performance in image classification tasks. Without going into details, our classify-and-count design can be summarized as follows:

1. Train a CNN to predict the LC class of a satellite image ‘tile’ (i.e. a small, fixed-size sub-image).
2. Divide the satellite images covering a ‘target area’ (i.e. the territory for which LC statistics have to be computed) into tiles.
3. Use the trained CNN to predict the LC class of all the tiles generated in step (2).
4. Obtain LC statistics for the target area by simply computing the absolute or relative frequencies of predicted LC classes.

We decided to adopt the EuroSAT dataset [10] as training set for our CNN. As CNN model, we selected Google’s Inception-V3 architecture [11], which we customized and trained on the EuroSAT dataset. EuroSAT contains 27,000 manually labelled image patches of size 64 x 64 pixels. These patches have been cropped from carefully selected Sentinel-2 satellite images covering 34 European countries, and manually labelled according to a non-standard LC taxonomy of only 10 classes. The dataset is roughly balanced with respect to the 10 classes, as class cardinalities range from 2,000 to 3,000 patches. As expected, our customized Inception-V3 CNN performed remarkably well on EuroSAT data, achieving impressive classification performance levels on a random 25% test set (e.g. Accuracy: 98.43%, Cohen’s Kappa: 98.25%). Once the CNN has been trained on the EuroSAT dataset, our automatic LC estimation system can be fed with a satellite image and return LC statistics for the corresponding territory. To do so, our classify-and-count approach is operationalized according to the following logical steps:

1. The input Sentinel-2 image is split into a set of (possibly overlapping) tiles of size 64 x 64 pixels. These tiles are generated by cropping the input image along a regular spatial grid, through a ‘sliding window’ algorithm.
2. The trained CNN classifies one tile at a time and logically links the predicted LC class to the corresponding area of the original image. The output of the whole process is a ‘classification matrix’: each element of this matrix corresponds to a tile of the original image and stores its predicted LC class.
3. The area share of each LC class for the whole territory depicted in the input satellite image is estimated by the relative frequency of the corresponding label within the classification matrix.
4. A land cover map of the territory depicted in the input satellite image is obtained by rendering the classification matrix as a raster image.

While the LC statistics calculated in **S3)** have to be regarded as the main output of our system, a further interesting artefact can be distilled, as a by-product, from the classification matrix. Indeed, as mentioned in **S4)**, an *automated land cover map* can be produced by simply rendering the classification matrix as a raster image. It is worth stressing that this is only possible because of the geometric structure of the systematic spatial sample of tiles generated by the sliding window algorithm in **S1)**.

## Early design of the system: the overestimation issue of Land Cover classes ‘River’ and ‘Highway’

We tested and validated the early design of our system on two large Sentinel‑2 images representing very different Italian territories: the ‘Lecce image’ and the ‘Pisa image’ (with an area of 751 km2 and 443 km2, respectively) [6]. We carried out the quality evaluation in two ways. First, we compared the automated LC estimates produced by our system with information available from [CORINE](https://land.copernicus.eu/pan-european/corine-land-cover) and [LUCAS](https://ec.europa.eu/eurostat/web/lucas) (to this end, we had of course to devise tentative and approximate mappings between different classifications). Second, we carefully examined visually the corresponding automated LC maps. This analysis showed that our automated LC estimates had a remarkably good accuracy for most LC classes, except for a *systematic upward bias* affecting narrow linear structures like rivers and highways. This overestimation issue turned out to be inextricably linked to the adoption of EuroSAT as training set and to our tile-based classify-and-count approach. In fact, EuroSAT constrains our CNN to process tiles of size 64 x 64 pixels, whose surface area is big (about 41 hectares, given the 10m per pixel resolution of Sentinel-2 RGB bands). But rivers and highways are narrow linear structures. Therefore, any highway fragment framed into a tile occupies just a rather small portion of the tile. Nonetheless, whenever the CNN correctly detects a highway in a tile, our system attributes the whole surface area of the tile to the ‘Highway’ class. This evidently leads to overestimation. The same happens, of course, when the ‘River’ class is concerned. Figure 1 provides a clear-cut demonstration of the overestimation issue affecting ‘River’ and ‘Highway’ LC classes, and allows appreciating visually the scale of the implied upward bias in LC estimation. The left panel shows a detailed view of the course of the Arno River, cropped from the north-east quadrant of the ‘Pisa image’ and overlaid with a semitransparent version of the corresponding automated LC map [6]. While the topology of the Arno and of nearby canals seems correctly captured and well described by the map (pale blue areas), the width of the detected water areas is evidently much larger than it should be.

|  |
| --- |
|  |

**Figure 1. Visual illustration of the overestimation issue affecting our early system with respect to LC classes ‘River’ (left panel: bends of the Arno River, cropped from the north-east quadrant of the ‘Pisa image’, see [6]) and ‘Highway’ (right panel: a fragment of a highway cropped from the south-east quadrant of the ‘Lecce image’, see [6]).**

The right panel shows a fragment of a highway cropped from the south-east quadrant of the ‘Lecce image’, along with a green edge-line that outlines the borders of the ‘Highway’ class in the corresponding LC map. To detect this edge-line, the Canny Edge Detector algorithm was used [6]. Unsurprisingly, the average horizontal distance between the green lines, as measured by a GIS, almost exactly matches the width of the tiles we exploit in our classify-and-count approach, i.e. 640 meters.

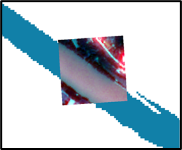
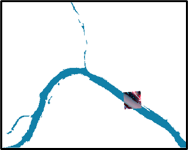
## System re-design: integrating a Deep Learning model for Semantic Segmentation

As anticipated in the Introduction, we decided to address the overestimation issue affecting the original design of our system by integrating our tile-based CNN model with a proper image-segmentation algorithm (see [7]). Semantic Segmentation [8] is the task of localizing objects of interest within a digital image (e.g. rivers in our application scenario) by identifying their shape and/or contours. The segmentation process groups different image pixels together according to their features, and binds them to a common class label (e.g. ‘River’). Among several solutions for image-segmentation proposed in the field of Deep Learning, our choice fell on the U-Net [9] model for its intuitive architecture and because it succeeded in many real-world applications. The basic idea that guided the re-design and improvement of our system is simple:

1. Train the U-Net to identify and reconstruct only rivers and highways. Use the trained U-Net to produce partial LC maps of input images (i.e. maps where only classes ‘River’ and ‘Highway’ are detected).
2. Train our existing Inception-V3 CNN on all the other LC classes of the EuroSAT dataset, and let the old-fashion classify-and-count approach produce partial LC maps of input images (i.e. maps where classes ‘River’ and ‘Highway’ are no longer detected).
3. Properly merge the partial LC maps produced in step (1) and (2).
4. For all the pixels of the merged map where the U-Net detected either class ‘River’ or class ‘Highway’, trust the U-Net and neglect the predictions of the CNN.

The final output of steps (1) – (4) is an integrated, complete (and still automated) Land Cover map, encompassing all the original EuroSAT LC classes. Automated LC estimates can eventually be obtained from this integrated map by simply computing class frequencies across its pixels.

While the logic underpinning the proposed re-design and improvement of our system is simple, it comes with non-negligible costs in terms of data preparation. Indeed, in order to train the U-Net to perform the segmentation of rivers and highways, a suitable training set has to be constructed for each of the corresponding LC classes. Here we focus on the ‘River’ LC class and only quickly hint at ongoing developments for the ‘Highway’ class. A good segmentation training set for rivers requires (i) samples of diverse Sentinel-2 images representing portions of rivers of different sizes and shapes, along with (ii) the corresponding segmentation masks. These segmentation masks can be thought as binary images (e.g. black and white images) where each pixel value encodes the information about the presence (white pixel) or absence (black pixel) of a river in the pixel location. While the EuroSAT dataset offers plenty of European rivers images (2’500), the generation of the corresponding segmentation masks is, of course, a non-trivial task. We tackled this task by leveraging the auxiliary information on European water areas provided by Copernicus [High Resolution Layers](https://land.copernicus.eu/pan-european/high-resolution-layers). This procedure entailed two phases. In the first phase, we managed to automatically build provisional segmentation masks for all the 2’500 river examples. To this end we used Python’s API for the GDAL library; see Figure 2 for an intuition.



**Figure 2. Overlaying a sample EuroSAT ‘River’ image on the Water High Resolution Layer (QGIS).**

In the second phase, we evaluated these masks by visual inspection, rejecting those whose correspondence to the input EuroSAT image was not accurate enough. This way, we ended up with 1’500 validated segmentation masks univocally associated to EuroSAT ‘River’ examples. These 1’500 image-pairs constitute our final U-Net training set for the ‘River’ segmentation task. Three image-pairs examples are shown in Figure 3 below.

|  |
| --- |
|  |

**Figure 3. Three sample EuroSAT ‘River’ images (left) with the corresponding obtained segmentation masks (right).**

# Results

Once trained, our U-Net can be exploited for the segmentation of rivers within production-scale input images. To this end, a ‘sliding-window algorithm’ must be applied, so that the U-Net can generate a partial LC map covering the whole input image. Figures 4 – 7 provide one example of a final (i.e. merged) LC map obtained for a large Sentinel-2 input image (the ‘Pisa image’ [6]). This output was generated by reconciling and integrating the partial maps produced, for the same territory, by the Inception-V3 CNN and by the U-Net. As we expected and hoped for, the ‘River’ overestimation issue no longer affects the final LC map returned by our re-designed system.

# Conclusions

The integration of an advanced model for Semantic Segmentation (i.e. a customized U‑Net Deep Learning architecture) into our processing pipeline helped us solve the overestimation issue affecting our previous classify-and-count approach for linear narrow Land Cover classes, like rivers and highways. We are currently using [OpenStreetMap](https://www.openstreetmap.org/) data as auxiliary information to create segmentation masks associated to EuroSAT ‘Highway’ images, along the lines of what we did for the ‘River’ class.

|  |  |
| --- | --- |
| **Figure 4. Inception-V3 output showing the ‘River’ overestimation issue (‘Pisa image’).** | **Figure 5. Inception-V3 classification without the ‘River’ class (‘Pisa image’).** |
| **Figure 6. U-Net classification output for LC class ‘River’ (‘Pisa image’).** | **Figure 7. Merge of U-Net and Inception-V3 output LC maps (‘Pisa image’).** |

# References

1. I. Goodfellow, Y. Bengio, A. Courville (2016). Deep learning. MIT press.
2. A. Gatti and A. Bertolini (2013). Sentinel-2 products specification document. Available online, URL:

<https://sentinel.esa.int/documents/247904/685211/Sentinel-2-Products-Specification-Document>

1. E. Bernasconi, F. Pugliese, D. Zardetto, M. Scannapieco (2019). Satellite-Net: Automatic Extraction of Land Cover Indicators from Satellite Imagery by Deep Learning. In Proceedings of NTTS 2019.
2. F. De Fausti, F. Pugliese, D. Zardetto (2020). Automated Land Cover Maps from Satellite Imagery by Deep Learning. In Book of short Papers - SIS 2020, Pearson, 2020, pp. - , ISBN: (in press).
3. ESSnet Big Data II. URL:

<https://webgate.ec.europa.eu/fpfis/mwikis/essnetbigdata/index.php/ESSnet_Big_Data>

1. WPH – ESSnet Big Data II. Deliverable H1: Interim technical report. Available online, URL:

<https://webgate.ec.europa.eu/fpfis/mwikis/essnetbigdata/images/3/31/WPH_Deliverable_H1_Interim_technical_report_2019_11_05.pdf>

1. WPH – ESSnet Big Data II. Deliverable H2: Report on results and activities executed by case studies. Available online (to appear).
2. A. Garcia-Garcia, S. Orts-Escolano, S. Oprea, V. Villena-Martinez, P. Martinez-Gonzalez, J. Garcia-Rodriguez (2018). A survey on deep learning techniques for image and video semantic segmentation. Applied Soft Computing, 70, 41-65. Elsevier.
3. O. Ronneberger, P. Fischer, T. Brox (2015). U-net: Convolutional networks for biomedical image segmentation. In International Conference on Medical image computing and computer-assisted intervention (pp. 234-241). Springer, Cham.
4. P. Helber, B. Bischke, A. Dengel, D. Borth (2019). Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing.
5. C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna (2016). Rethinking the inception architecture for computer vision. In Proceedings of the IEEE conference on computer vision and pattern recognition.