

# Reproducibility review of: Road Network Mapping from Multispectral Satellite Imagery: Leveraging Deep Learning and Spectral Bands

Eftychia Koukouraki 

2024-05-31



This report is part of the reproducibility review at the AGILE conference. For more information see <https://reproducible-agile.github.io/>. This document is published on OSF at <https://doi.org/10.17605/osf.io/txgzv>. To cite the report use

Koukouraki, E. (2024, May). Reproducibility review of: Road Network Mapping from Multispectral Satellite Imagery: Leveraging Deep Learning and Spectral Bands. <https://doi.org/10.17605/osf.io/txgzv>

## Reviewed paper

Samuel Hollendonner, Negar Alinaghi, and Ioannis Giannopoulos: Road Network Mapping from Multispectral Satellite Imagery: Leveraging Deep Learning and Spectral Bands, AGILE GIScience Ser., 5, 6, <https://doi.org/10.5194/agile-giss-5-6-2024>, 2024

## Summary

The paper uses semantic segmentation to extract the road network from RGB and multi-spectral high resolution satellite imagery and applies post-processing to improve the segmentation results. The initial dataset is called SpaceNet (challenge 3) and is openly available, but the authors provided a small pre-processed subset of it in order to verify the functionality of the code. As we were able to confirm that the used data and code are available and reusable, but unable to verify the reported results (due to the lack of available computational resources on the side of the reviewer), the reproduction of the paper is considered **partially successful**.

# Reproducibility reviewer notes

## Data and code sharing

The SpaceNet 3 dataset is available at <https://spacenet.ai/spacenet-roads-dataset/> and can be freely downloaded, but requires an Amazon Web Services account. The training dataset is 80.6 GB and the test dataset is 16.2 GB, for a total of 106.8 GB of data. The code, instructions and partly pre-processed subset of SpaceNet 3 dataset have been made publicly available at the following link <https://doi.org/10.48436/d5z5b-3vk12>. The authors have licensed all data files under CC BY 4.0 and all software under MIT License. The original SpaceNet dataset was pre-processed in order to convert the geojson files into training images, to reduce the satellite imagery to 8-bit format, and then convert it to PNG files, using a modified version of the Average Path Length Similarity (APLS) Python library (<https://github.com/CosmiQ/apls>). This modification was not part of the shared code, nor was it clearly described in the instructions. The subset that we used to verify the code (Python scripts) had already undergone this pre-processing and was readily available in the materials repository of this paper.

## Computational environment

The Python environment for the reproducibility review was created using conda, following as closely as possible the contents of the environment.yml file provided with the code. An exact recreation of the environment was not possible due to the different operating systems and hardware platforms on which the original study and the reproduction were done. The hardware on which the reproduction was run consists of 15.0 GiB of memory and an i7-1185G7 @ 3.00GHz × 8 processor. The operating system is Ubuntu 20.04.6 LTS. There is no special graphical processing unit (GPU) on the system, so all GPU-related computations had to be elaborated with the central processing unit (CPU). This contrasts with the original study, which was conducted on a Windows PC with a CUDA-enabled NVIDIA graphics card. The recreated Python environment has been exported as a .yml file and can be found under the name *reproduced/environment.yml* in the OSF repository of this report.

## Reproduction efforts

Using the materials provided by the authors, we were able to verify the workflow for the RGB imagery and functionality of the following scripts: *preprocessing.py*, *train\_model.py*, *postprocessing.py* and *evaluation.py*. We were not able to verify the functionality of *ms\_channel\_analysis.py* and *ms\_channel\_separation.py*, which are involved in the workflow for the multi-spectral imagery, due to the lack of available data. The scripts seem to be executed, but we cannot verify their output.

The script *preprocessing.py* is responsible for creating the PNG files under *code/data/tiled512/RGB/images* and *code/data/tiled512/RGB/rehashed\_ones*. In order to successfully execute the script, we had to manually create the directory *code/data/tiled512/RGB*.

The script *train\_model.py* is responsible for creating the file *model\_parameters.txt* and subdirectories of *code/segmentation\_results/RGB\_U-Net\_DenseNet201*, namely *checkpoints*, *epoch\_results*, *model*, *results*, *weights* and their contents. We run the training for 10 epochs and the log can be found under the name *reproduced/log\_train\_model.txt* in the OSF repository of this report.

The script *postprocessing.py* is responsible for creating the GeoJSON files under *geojsons/qgis\_geojsons* and *geojsons/sub\_geojsons*, the pickle files under *graphs*, and the PNG files under *skeletons* and *stitched*. All of these files and directories are created under *code/segmentation\_results/RGB\_U-Net\_DenseNet201/*. In order to successfully execute the script, we had to modify line 317 of the script as following.

```
# image_name = os.path.splitext(image.replace('_00_00_', ''))[0]
image_name = os.path.splitext(image.replace('_00_', ''))[0]
image_name = os.path.splitext(image_name.replace('_01_', ''))[0]
image_name = os.path.splitext(image_name.replace('_02_', ''))[0]
```

The script *evaluation.py* is responsible for creating the files *confusion\_matrix\_data.txt*, *F1\_scores\_stitched.txt*, *GED106\_4.json*, *GED\_scores\_per\_city.txt*, *IoU\_scores\_stitched.txt*, *relative\_GED0\_6137.json*,

*relGED\_scores\_per\_city.txt*, *statistics\_pls\_86\_31\_comb78\_79.json* and *Topology\_cities.txt* under *code/segmentation\_results/RGB\_U-Net\_DenseNet201*. Similarly to the previous script, we had to modify the script in order to execute it successfully. The modification refers to line 380 and is shown below:

```
# img = os.path.splitext(img.replace('_00_00', ''))[0]
img = os.path.splitext(img.replace('_00', ''))[0]
img = os.path.splitext(img.replace('_01', ''))[0]
img = os.path.splitext(img.replace('_02', ''))[0]
```

## Communication with the author

All necessary communication was done with the first author, who was helpful and willing to answer our questions. The first author updated the material repository in order to fix some bugs, which allowed us to verify part of the workflow as described above. The materials were shared through the data repository of the academic institution of the authors and had to be reviewed internally before they were made publicly available, something that induced delays in the process of the reproducibility review.