

Classifying Ditch and Stream Channels Mapped From High-Resolution Digital Elevation Models Using Machine Learning

Busarello, M., Ågren, A., & Westphal, F., Lidberg, W.

Jan 27th, 2026

We have trained machine learning models to classify ditches and stream channels using hydrological features. In a previously published work ([Busarello et al., 2025](#)), we used topographical indices to train a U-Net ([Ronneberger et al., 2015](#)) model to recognize channels across the landscape at the pixel level. Now, we converted these pixels into vector data, extracted hydrological features as tabular data for the channels, and trained an XGBoost ([Chen et al., 2014](#)) model to reclassify them as either ditch or stream with a higher performance.

This repository contains the U-Net model used to map the water channels, the XGBoost models to classify water channels, the DEMs to calculate the hydrological and topographical features, and the ground truth polylines. The code available was used to train the models, predict the classes in new data, and perform all the pre-processing and processing steps to reproduce our study. If needed, future updates can be found at <https://github.com/mbusarello/Classifying-Stream-and-Ditch-Channels-Mapped-from-High-Resolution-DEMs-Using-Machine-Learning/>.

The channel network was split into different segment sizes to train the models: 10 m, 25 m, 50 m, 100 m, and unsplit. All of them can be created using the code present in the repository. The model trained with the 50 m segment size (hybrid) was chosen as the most balanced. The U-Net mapped network was improved by a more accurate and robust classification of ditches and streams from the XGBoost model, also removing most of the false positives successfully.

Data

- ground_truth.zip

Ground truth channel network (polyline shapefile), containing the fields:

- ZoneName – name of the mosaiced zone
- lengthM – length of the polylines, in meters
- length_km – length of the polylines, in kilometers
- diketyf – type of ditch, in Swedish
- ditcType – type of ditch, in English
- lannamn – county name
- kommunnamn – municipality name
- created_da – date of data creation
- created_us – user who created the data
- last_edite – date of the last data edit
- last_edi_1 – user to last edit the data
- mosc_zone – number of the mosaiced zone
- id – type of channel (1 - natural stream, 2 - ditch)

- DEM.zip

Digital elevation model from the study areas.

- training_and_testing_data_50m.zip

Complete training and testing data for the 50 m segment size XGBoost model, the most balanced one.

The data for this study comes from 12 study areas spread across Sweden, with different characteristics regarding land use and forest cover, among others. The DEMs were calculated from the laser data, which comes from [Lantmäteriet](#). The data is originally organized into tiles of 2500 m x 2500 m, later being split into chips of 250 m x 250 m to train the U-Net model. The tiles were mosaiced to calculate the hydrological features needed to train the XGBoost model.

The description of scripts used to gather the data and train the models can be found below. Any code can be run on Linux by typing into the command line:

```
python <script_name> --help
```

Mapping water channels with a U-Net model

Dockerfile

The environment to use the scripts requires Docker (<https://www.docker.com/>) installed to make use of the dockerfile. Setting the environment means that there is no need to install further libraries and packages, everything will be retrieved from the dockerfile. The dockerfile can be used to set the environment by typing on the command line (Linux):

```
docker build -t <name> .
```

To run the container, the command line is:

```
docker run <name>
```

Further commands for docker run can be found [here](#) and be added according to the user's needs.

- calculating_slope.py

Calculates the Slope from the DEM.

- calculating_skyfactor.py

Calculates the topographic index Sky-View Factor from the DEM.

- prediction_unet_prob_multiclass.py

Returns the mapped water channels from the DEM, using the Slope and Sky-View Factor as the input data. It also returns the probability maps for each channel type, and background. The order of the input matters: slope needs to be listed as input before the sky-view factor.

- DitchesStreamsSlopeSVF.h5

The model used to map water channels using Slope and Sky-view Factor as input data. Used as input in the script prediction_unet_prob_multiclass.py.

- fixing_probability_threshold.py

Used to only keep the most relevant values in the probability maps, those that are above the 0.5 threshold. Values above the threshold are kept, while values below are converted to zero.

- combining_probabilities.py

Combines the probability maps from ditches and streams after updating them according to the threshold.

- subtracted.py

Subtracts the combined probability map from the original DEM, in order to highlight the channels mapped by the U-Net.

- skeletonizing.py

Converts the channels mapped by the U-Net from raster to vector polylines.

- connecting_unet_vector_lines.py

Attempts to connect the gaps between the U-Net mapped vector polylines.

- splitting_into_segments.py

Splits the U-Net channel network into the user-attribute segment length.

- calculating_length.py

Calculates the length of each channel segment.

Hydrological features

- mosaics.py

Mosaics the subtracted DEM tiles to calculate the hydrological features. Must also be used to mosaic the probability maps.

- upslope_depression_storage.py

Calculates the Upslope Depression Storage. Should be used in the unfilled DEM.

- fillmosaics.py

Fills the mosaics to remove missing data.

- burningstreamsatroads.py

Deepens the stream network on the DEM when it crosses mapped roads. The road and stream data used are from the Swedish Property Map.

- breaching_least_cost.py

Performs optimal depression breaching to remove topographic depressions with a lower impact.

- average_flowpath_slope.py

Calculates the Average Flowpath Slope with the depressionless DEM as input.

- average_upslope_fp_length.py

Calculates the Average Upslope Flowpath Length, with the depressionless DEM as input.

- max_upslope_fp_length.py

Calculates the Maximum Upslope Flowpath Length, using the depressionless DEM as input.

- flow_acc.py

Calculates the Flow Accumulation, using the depressionless DEM as input.

Extracting the features

- creating_buffer.py

Creates a 3 m buffer around the split segments to extract near-reach attributes.

- sampling_target_data_polylines_gt.py

Samples the class from the ground truth, to be used as target data to train the XGBoost model.

- zonal_sampling_unet.py

Samples the U-Net prediction raster, selecting the majority class within the sampling buffer.

- zonal_sampling_attributes.py

Samples the hydrological features within the sampling buffer, giving six statistical measurements: maximum, minimum, median, mean, and standard deviation.

- calculate_sinuosity.py

Calculates the sinuosity of each channel segment from the polyline vectors and transfer it to the sampling buffers.

XGBoost training and prediction

- train_xgboost.py

Trains an XGBoost model to classify channels. The input is the buffers that have extracted the sampled rasters with the ground truth and the hydrological data.

- xgboost_prediction.py

Loads a trained model and, given unseen data, classifies the buffered segments into channel types. The classes predicted are:

0 – background

1 – streams

2 – ditches

Pre-trained XGBoost models

- model_10m.joblib

A model trained to classify channel segments with a maximum of 10 m.

- model_25m.joblib

A model trained to classify channel segments with a maximum of 25 m.

- model_50m.joblib

A model trained to classify channel segments with a maximum of 50 m.

- model_100m.joblib

A model trained to classify channel segments with a maximum of 100 m.

- model_unsplit.joblib

A model trained to classify channel segments of any segment size.

More explanation about using any Python script can be found by typing on the command line (Linux):

```
python <code_name.py> -help
```

License: Creative Commons Public Domain Dedication (CC0 1.0)

Responsible organization: Swedish University of Agricultural Sciences

DOI: <https://doi.org/10.5878/r0x8-kx56>

References:

Busarello, M.D.S.T., Ågren, A.M., Westphal, F. & Lidberg, W. (2025). Automatic detection of ditches and natural streams from digital elevation models using deep learning. *Computers & Geosciences*, 196, 105875. <https://doi.org/10.1016/j.cageo.2025.105875>

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., Li, M., Xie, J., Lin, M., Geng, Y., Li, Y. & Yuan, J. (2014). xgboost: Extreme Gradient Boosting. <https://doi.org/10.32614/CRAN.package.xgboost>

Ronneberger, O., Fischer, P. & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv. <http://arxiv.org/abs/1505.04597> [2023-03-14]