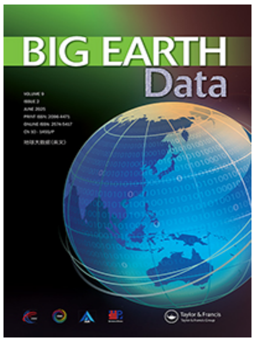


Big Earth Data

ISSN: 2096-4471 (Print) 2574-5417 (Online) Journal homepage: www.tandfonline.com/journals/tbed20

Detection of drainage ditches from LiDAR DTM using U-Net and transfer learning

Holger Virro, Alexander Kmoch, William Lidberg, Merle Muru, Wai Tik Chan, Desalew Meseret Moges & Evelyn Uuemaa

To cite this article: Holger Virro, Alexander Kmoch, William Lidberg, Merle Muru, Wai Tik Chan, Desalew Meseret Moges & Evelyn Uuemaa (2025) Detection of drainage ditches from LiDAR DTM using U-Net and transfer learning, Big Earth Data, 9:2, 243-264, DOI: 10.1080/20964471.2025.2491177

To link to this article: https://doi.org/10.1080/20964471.2025.2491177

Published online: 21 Apr 2025.

BIG EARTH DATA
2025, VOL. 9, NO. 2, 243–264
https://doi.org/10.1080/20964471.2025.2491177

RESEARCH ARTICLE

OPEN ACCESS

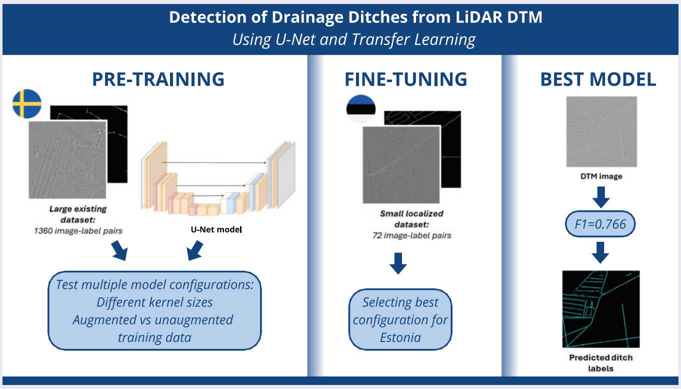
Detection of drainage ditches from LiDAR DTM using U-Net and transfer learning

Holger Virro[a], Alexander Kmoch[a], William Lidberg[b], Merle Muru[a], Wai Tik Chan[a], Desalew Meseret Moges[a] and Evelyn Uuemaa[a]

[a]Department of Geography, Institute of Ecology and Earth Sciences, University of Tartu, Tartu, Estonia; [b]Department of Forest Ecology and Management, Swedish University of Agricultural Sciences, Umeå, Sweden

ABSTRACT

Accurate mapping of ditches is essential for effective hydrological modeling and land management. Traditional methods, such as manual digitization or threshold-based extraction, utilize LiDAR-derived digital terrain model (DTM) data but are labor-intensive and impractical to apply for large-scale applications. Deep learning offers a promising alternative but requires extensive labeled data, often unavailable. To address this, we developed a transfer learning approach using a U-Net model pre-trained on a large high-quality Swedish dataset and fine-tuned on a smaller localized Estonian dataset. The model uses a single-band LiDAR DTM raster as input, minimizing preprocessing. We identified the optimal model configuration by systematically testing kernel sizes and data augmentation. The best fine-tuned model achieved an overall F1 score of 0.766, demonstrating its effectiveness in detecting drainage ditches in training data-scarce regions. Performance varied by land use, with higher accuracy in peatlands (F1 = 0.822) than in forests (F1 = 0.752) and arable land (F1 = 0.779). These findings underscore the model's suitability for large-scale ditch mapping and its adaptability to different landscapes.


Detection of Drainage Ditches from LiDAR DTM
Using U-Net and Transfer Learning

PRE-TRAINING

Large existing dataset: *1360 image-label pairs*

U-Net model

Test multiple model configurations:
Different kernel sizes
Augmented vs unaugmented training data

FINE-TUNING

Small localized dataset: *72 image-label pairs*

Selecting best configuration for Estonia

BEST MODEL

DTM image

F1=0.766

Predicted ditch labels

ARTICLE HISTORY
Received 8 December 2024
Accepted 20 March 2025

KEYWORDS
Drainage network; LiDAR; U-Net; semantic segmentation; transfer learning

CONTACT Holger Virro holger.virro@ut.ee Department of Geography, Institute of Ecology and Earth Sciences, University of Tartu, Vanemuise 46, Tartu 51003, Estonia

This article has been republished with minor changes. These changes do not impact the academic content of the article.

Supplemental data for this article can be accessed online at https://doi.org/10.1080/20964471.2025.2491177

1. Introduction

Drainage ditches are common in temperate lowland regions, where they are used to manage excess water in agricultural and forested landscapes (Du et al., 2024; Levavasseur et al., 2015; Roelens et al., 2018). Ditches have been introduced in the Baltic States, Finland and Sweden to drain wetlands for agriculture or afforested peatlands (Flyckt et al., 2022; Lidberg et al., 2023; Lõhmus et al., 2015).

They regulate groundwater levels, promote soil aeration, and support crop and tree growth, thus mitigating flood risks (Laurén et al., 2021; Melniks et al., 2022). In addition, ditches contribute to greenhouse gas emissions, particularly methane (CH_4), carbon dioxide (CO_2), and in agricultural areas, N_2O, making them significant for climate studies (Peacock et al., 2017, 2021; Wu et al., 2023). Therefore, accurate ditch mapping is crucial to facilitate sustainable agriculture and forest management and informed policymaking, yet the lack of precise ditch data remains a challenge (Du et al., 2024; Koski et al., 2023; Robb et al., 2023; Roelens et al., 2018).

Conventionally, ditch mapping mainly relies on field surveys and/or stereo mapping from aerial images (Koski et al., 2023). These methods are labor-intensive and time-consuming, particularly in forested areas where the canopy obscures the ditches. Since ditches are often narrow, shallow, heavily vegetated, and temporarily inundated, conventional mapping methods face significant challenges in accurately detecting their full extent (Du et al., 2024). In recent years, airborne LiDAR (Light Detection and Ranging) has enabled the creation of precise 3D point clouds of landscapes by scanning large areas from planes using laser pulses (Du et al., 2024; Flyckt et al., 2022; Roelens et al., 2018).

Airborne LiDAR data can be utilized for ditch detection by processing the LiDAR-derived Digital Terrain Model (DTM) or directly using the point cloud data. Previous studies focusing on the raw LiDAR point clouds for ditch detection have predominantly concentrated on smaller, open agricultural landscapes where ditches are typically easily accessible, well-maintained, and less complex (Bailly et al., 2008; Flyckt et al., 2022; Roelens et al., 2018). Moreover, areas with a high potential for soil hydrological saturation due to topography should be identified before extracting ditches from the point cloud (Rapinel et al., 2015). Therefore, a LiDAR-derived DTM is preferred in larger and forested areas because it effectively filters out vegetation and other non-ground features, enhancing accuracy (Du et al., 2024). DTMs derived from LiDAR data are typically constructed at specific spatial resolutions aligned with the precision of the LiDAR data used (Rapinel et al., 2015).

Standard methods for extracting ditches from DTMs include hydrological modeling (topographical modeling), threshold-based extraction, and manual digitization. Hydrological modeling, which involves techniques like flow routing, pit filling, flow direction computation, and defining minimum contributing areas, is particularly prevalent for large-scale drainage network extraction due to its efficiency and straightforwardness (Stanislawski et al., 2018; Zhang et al., 2021). Threshold-based techniques for drainage extraction depend on geometric measurements and similarity thresholds. The accuracy of these methods is influenced by the choice of thresholds and window sizes for curvature calculations (Kiss et al., 2015; Pirotti & Tarolli, 2010). Effective application of this approach requires substantial prior knowledge to set appropriate weights and thresholds, as incorrect configurations can significantly affect the results (Huang et al., 2023). Manual digitization, on the other hand, involves experts visually interpreting and delineating ditches from DTMs or related products (Wang & Fareed, 2021). Each method has limitations in handling complex ditch detection tasks due to their reliance on predefined parameters and expert judgment. Additionally, while supervised machine learning methods address some complexities, they often rely on hard-coded feature sets for the shape and color of ditches, which may not provide a comprehensive object representation (Robb et al., 2023).

Semantic image segmentation using deep learning (DL) algorithms has revolutionized image analysis and computer vision (Kattenborn et al., 2019). Convolutional neural networks (CNNs) are at the forefront of this revolution, significantly advancing object detection and pattern recognition (LeCun et al., 2015). These networks utilize convolution operations, i.e. applying a filter (or kernel) over the input data, to exploit the images' hierarchical (local to global) features (Waldner & Diakogiannis, 2020). Unlike traditional pixel-based methods, CNNs efficiently analyze image textures by capturing the contextual signals of multiple neighboring pixels (Kattenborn et al., 2019). This allows them to effectively capture the complexity of heterogeneous arrangements from remote sensing data (Koski et al., 2023; Robb et al., 2023).

Several deep semantic network architectures, e.g. DeepLab, SegNET, and U-Net, have been developed and applied in pixel-classification (Du et al., 2020). The U-Net architecture, initially developed for biomedical image segmentation (Ronneberger et al., 2015), is one of the most widely used CNN architectures in image segmentation with high accuracy (Falk et al., 2019). U-Net is effective in geospatial mapping of small and narrow elements because it detects shapes and textures within small regions at various scales, ultimately producing a per-pixel classification of the input imagery (Flood et al., 2019; Malik et al., 2021). The U-Net backbone architecture, also known as encoder-decoder, helps in image segmentation by compressing high-dimensional image information with the encoder and then upscaling it back to the original resolution with the decoder (Waldner & Diakogiannis, 2020). Thus, it allows precise localization and classification of specific image regions.

Although DL approaches combined with semantic network architectures have demonstrated promising results in ditch detection, they require large volumes of labeled training data, which is often impractical, especially in regions where data collection is challenging and resources are limited. This remains one of the bottlenecks for applying and exploring U-Net and other fully convolutional network architectures. One solution to overcome the lack of training labels is transfer learning, which involves leveraging existing training data from data-rich areas to pre-train an initial DL model and fine-tuning it with a small amount of locally collected training data (van Opbroek et al., 2015). The core idea is to pre-train the model to learn similar features from a general dataset and fine-tune the model to specific local characteristics. Although transfer learning has been successfully applied to remote sensing data for image segmentation (Wurm et al., 2019; Zhu et al., 2017), it has not been used to detect drainage ditches.

In Estonia, approximately 1 million hectares or 22% of land area are drained, with 584,400 ha for agricultural purposes and 338,400 ha for forestry, impacting 70% of the country's original wetlands (Ratt, 1985). Many drainage ditches, especially in forests, are not mapped, posing challenges for greenhouse gas (GHG) emission estimation in the land use, land use change and forestry (LULUCF) sector. Therefore, mapping the drainage ditches would significantly improve the current GHG emission estimation. To address this, we developed a novel transfer learning-based method for ditch detection, leveraging high-quality labeled training data from Sweden and showing that a small Estonian fine-tuning dataset is sufficient for effectively adapting the model to local conditions. Our approach minimizes preprocessing by using a single-band LiDAR DTM raster as input and systematically optimizes model performance through kernel size and data augmentation experiments. The resulting model achieved high accuracy across diverse land use classes, making it applicable for large-scale ditch mapping in Estonia and other countries with similar hydrological and topographic conditions.

2. Data

We used data from two different countries in our transfer learning workflow. We implemented data from a previous study conducted in Sweden (Lidberg et al., 2023) for pre-training and generated fine-tuning data by digitizing two 9 km^2 areas in Estonia.

2.1. Swedish pre-training data

The pre-training dataset originated from Lidberg et al. (2023) and consisted of 1360 image-label pairs. Each pair included a 0.5 m DTM raster, which Lidberg et al. (2023) processed with a high-pass median filter (HPMF) to emphasize minor differences in elevation. They produced the corresponding labels by manually digitizing ditches based on the HPMF raster. They also normalized the HPMF values to be between 0 and 1 for modeling. The dimensions of the HPMF-label pairs were 500×500 px.

2.2. Estonian fine-tuning data

We chose Estonia as the target region for transfer learning as its climate, topography, and land use patterns are comparable to Sweden’s. Both countries also have extensive drainage networks in forests and agricultural areas. Thus, pre-training the model on Swedish data should help it to generalize reasonably well to Estonian data.

We selected two 3×3 km areas in Estonia to generate the fine-tuning dataset (Figure 1a), which covered 18 km^2 in total. Area A is located in central Estonia and is dominated by forest, meaning many ditches are not visible on the orthophoto and become visible only in the HPMF raster. Area B is located in southeastern Estonia and has a mix of agricultural land and forest, with the addition of a peat extraction area in the site’s southeastern quarter. Ditches within the peat extraction area have regular spacing and are of similar length, unlike those in forests and fields. We selected these sites for fine-tuning, trying to capture differences in ditch characteristics in forests, arable lands and peatlands, the land use classes most commonly affected by drainage in Estonia, which should improve the generalizability of the fine-tuned models.

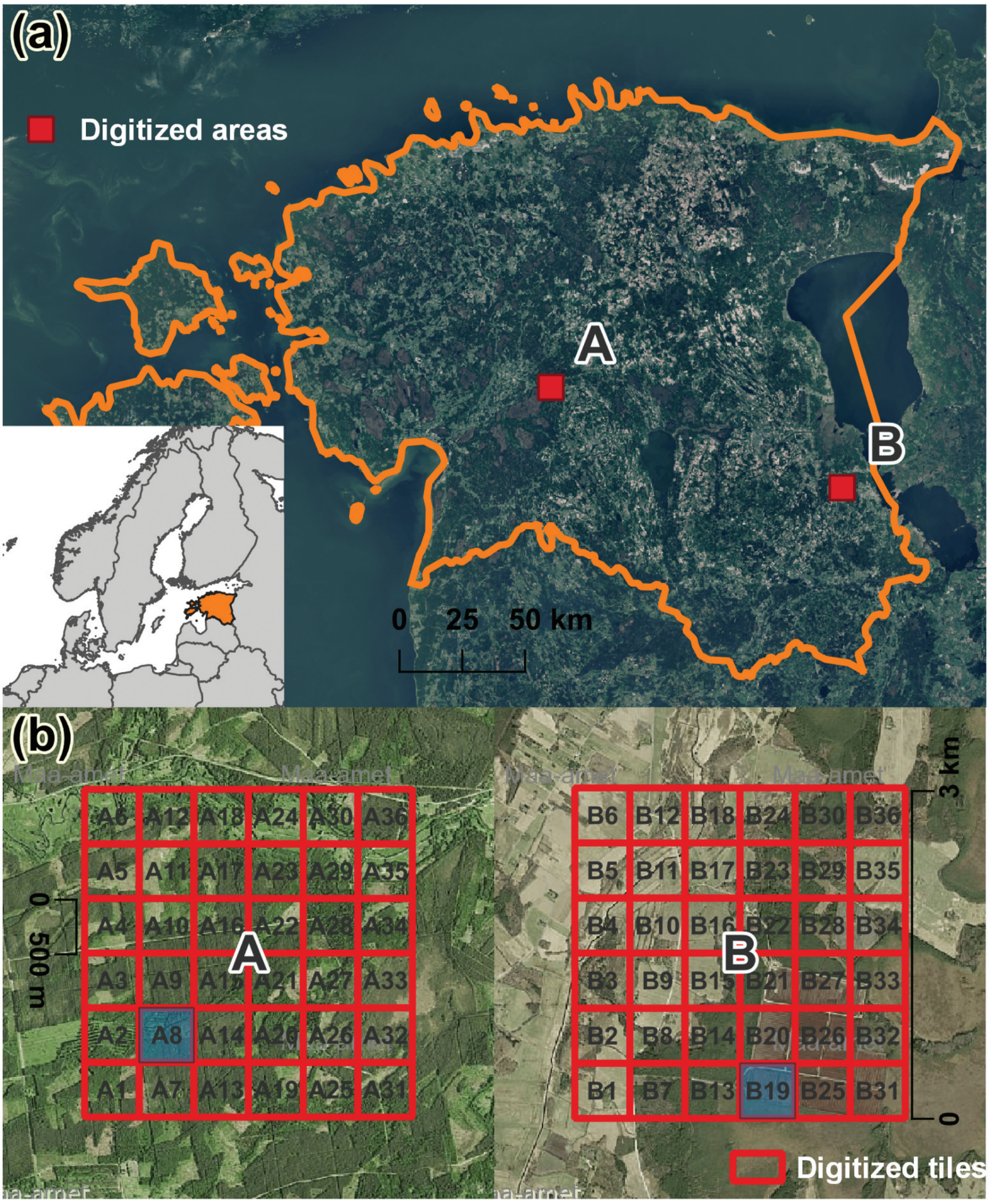


Figure 1. We selected two 3×3 km areas in Estonia to generate the fine-tuning dataset. (a) Location of areas A and B in Estonia. (b) Spatial distribution of 500×500 m tiles within the 3×3 km digitized areas. Forests dominate area A, while B includes fields, forests, and a peat extraction area. Tiles A8 and B19 are highlighted in blue and shown in detail in Figure 2.

We followed the workflow outlined in Lidberg et al. (2023) for generating labels in both areas for fine-tuning. First, we applied the HighPassMedianFilter tool from Whitebox (Lindsay, 2014) to 1 m DTM raster images provided by the Estonian Land Board, which have been derived from LiDAR data with an estimated density of 2.1 p/m^2 (Estonian Land Board, 2020). Then, we digitized and rasterized all ditches visible in the HPMF images in

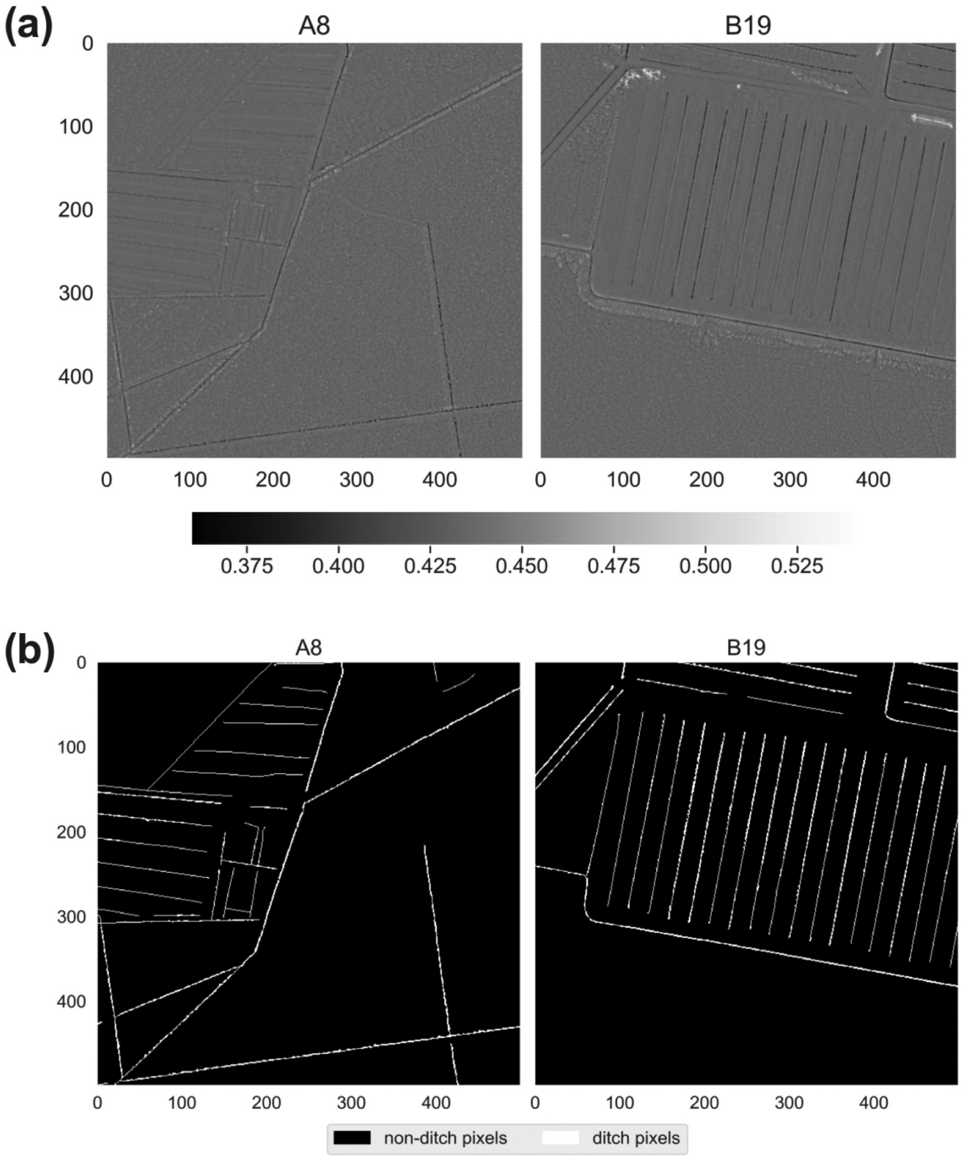


Figure 2. Examples of normalized HPMF and label pairs in tiles highlighted in blue in Figure 1b. (a) Normalized HPMF images for tiles A8 and B19. (b) Digitized labels for tiles A8 and B19.

QGIS 3.28 (QGIS Development Team, 2024). We generated 3 m buffers for the rasterized ditches, after which we labeled all pixels within buffers with HPMF values below 0.075 as ditch pixels, using the threshold determined by Lidberg et al. (2023). Finally, we divided each 3×3 km area into 500×500 m tiles, resulting in 72 hPMF-label pairs (Figure 1b).

Examples of HPMF-label pairs from areas A and B are given in Figure 2. We normalized the HPMF images like the Swedish pre-training data (Figure 2a). Tile A8 illustrates typical forest ditches found in area A. Tile B19 originates from a peat extraction area and is characterized by regularly spaced ditches (Figure 2b).

3. Methodology

3.1. U-Net model architecture

The U-Net model is a CNN that follows the autoencoder architecture, consisting of an encoder and a decoder (Figure 3). The left-hand side of Figure 3 represents the encoder part, which reduces the spatial dimensions of the input image while increasing the number of feature channels (Falk et al., 2019). This is called downsampling, which allows the model to extract and encode essential features into a compact latent representation (Kattenborn et al., 2019). The decoder, shown on the right side, then continuously expands the dimension from the latent space back to the original dimension, reconstructing the original input through a process called upsampling (Ronneberger et al., 2015). In contrast to a traditional autoencoder, which reconstructs the input image itself, U-Net transforms the input into a segmented output, where each pixel corresponds to a probability of belonging to a particular class. In addition, U-Net uses skip connections to link corresponding layers in the encoder and decoder, retaining high-resolution features from the encoder and combining them with the upsampled features in the decoder. This allows the model to perform precise localization, which is needed for accurate image segmentation (Längkvist et al., 2016; Long et al., 2015).

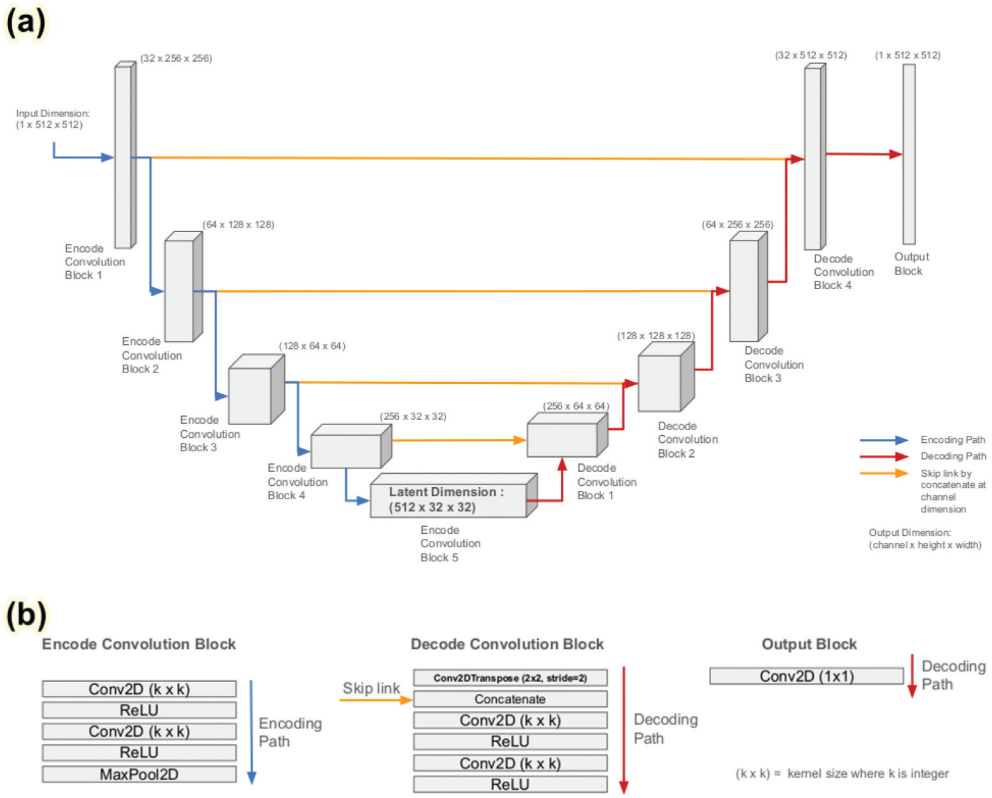

(a)
Input Dimension: (1 x 512 x 512)
(32 x 256 x 256)
Encode Convolution Block 1
(64 x 128 x 128)
Encode Convolution Block 2
(128 x 64 x 64)
Encode Convolution Block 3
(256 x 32 x 32)
Encode Convolution Block 4
Latent Dimension : (512 x 32 x 32)
Encode Convolution Block 5
(256 x 64 x 64)
Decode Convolution Block 1
(128 x 128 x 128)
Decode Convolution Block 2
(64 x 256 x 256)
Decode Convolution Block 3
(32 x 512 x 512)
Decode Convolution Block 4
(1 x 512 x 512)
Output Block
Encoding Path
Decoding Path
Skip link by concatenate at channel dimension
Output Dimension: (channel x height x width)
(b)
Encode Convolution Block
Conv2D (k x k)
ReLU
Conv2D (k x k)
ReLU
MaxPool2D
Encoding Path
Decode Convolution Block
Skip link
Conv2DTranspose (2x2, stride=2)
Concatenate
Conv2D (k x k)
ReLU
Conv2D (k x k)
ReLU
Decoding Path
Output Block
Conv2D (1x1)
Decoding Path
(k x k) = kernel size where k is integer

Figure 3. A visualization of the U-Net model implemented in our transfer learning workflow. (a) The general architecture of the U-Net and its convolutional layers. (b) The details of the encoding, decoding, and output block implementation.

3.2. U-Net based transfer learning for ditch detection

We implemented the U-Net architecture in our two-part transfer learning workflow (Figure 4). The details of pre-training and fine-tuning are explained in sections 3.2.1 and 3.2.2, respectively. Both parts consisted of five general steps:

(1) Data splitting. First, we split the data into training (80%) and test (20%) sets.
(2) Optional augmentation. Data augmentation increases variance in training data, meaning that the model generalizes (Waldner & Di-akogiannis, 2020). We investigated the effects of augmentation by augmenting the training set depending on the model configuration. We used the torchvision.transforms (Marcel & Rodriguez, 2010) Python module to apply basic geometric augmentation, which included three rotations (90, 180 and 270 degrees) and a horizontal flip.
(3) Image resampling. We used the skimage.transform Python module (van der Walt et al., 2014) to resample the images from 500×500 to 512×512 pixels to match U-Net input size (Figure 3a).
(4) Training the U-Net. We used the U-Net architecture shown in Figure 3a in step 4 of our modeling workflow. The model employs the rectified linear unit (ReLU) as the activation function in both the encoding and decoding paths (Figure 3b), with max-pooling layers used in the encoding path to reduce the spatial dimensions. We utilized the Adam optimizer with a learning rate of 0.0001 and a batch size of 4. We built the models using PyTorch 2.2 (Paszke et al., 2019) and trained them on the University of Tartu's Rocket HPC cluster (University of Tartu, 2018) with Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20 GHz (48 cores total) and NVIDIA Tesla V100 GPU.

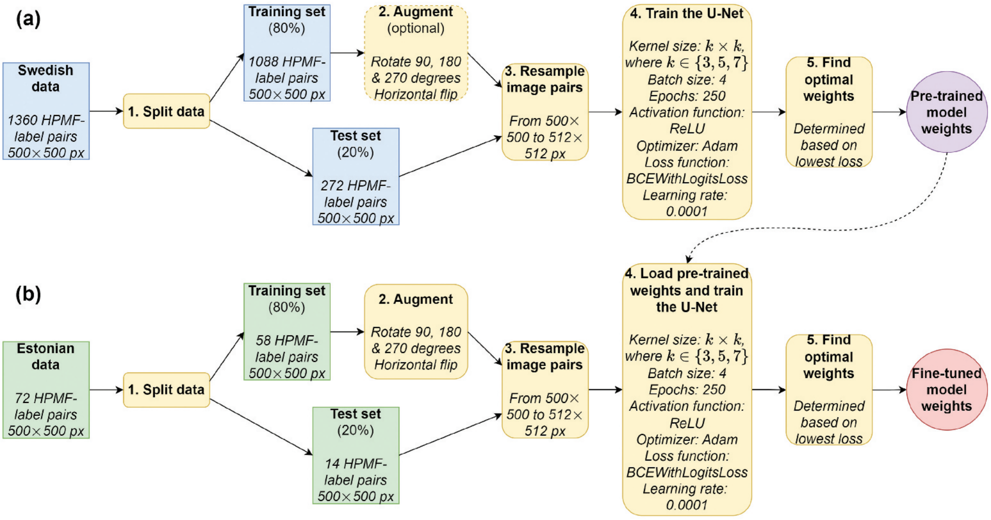

(a)
Swedish data
1360 HPMF-label pairs 500×500 px
1. Split data
Training set (80%)
1088 HPMF-label pairs 500×500 px
Test set (20%)
272 HPMF-label pairs 500×500 px
2. Augment (optional)
Rotate 90, 180 & 270 degrees Horizontal flip
3. Resample image pairs
From 500×500 to 512×512 px
4. Train the U-Net
Kernel size: $k \times k$, where $k \in \{3,5,7\}$
Batch size: 4
Epochs: 250
Activation function: ReLU
Optimizer: Adam
Loss function: BCEWithLogitsLoss
Learning rate: 0.0001
5. Find optimal weights
Determined based on lowest loss
Pre-trained model weights
(b)
Estonian data
72 HPMF-label pairs 500×500 px
1. Split data
Training set (80%)
58 HPMF-label pairs 500×500 px
Test set (20%)
14 HPMF-label pairs 500×500 px
2. Augment
Rotate 90, 180 & 270 degrees Horizontal flip
3. Resample image pairs
From 500×500 to 512×512 px
4. Load pre-trained weights and train the U-Net
Kernel size: $k \times k$, where $k \in \{3,5,7\}$
Batch size: 4
Epochs: 250
Activation function: ReLU
Optimizer: Adam
Loss function: BCEWithLogitsLoss
Learning rate: 0.0001
5. Find optimal weights
Determined based on lowest loss
Fine-tuned model weights

Figure 4. The two-part transfer learning workflow. (a) Pre-training consisted of five steps: (1) training data splitting, (2) optional data augmentation, (3) resizing the image pairs to 512×512 px, (4) training the U-Net, (5) extracting the best weights based on loss. (b) Fine-tuning mostly followed the pre-training workflow, except for step 4, where corresponding pre-trained weights were loaded to train the U-Net further.

Each model was trained for 250 epochs. In general, larger kernels help the model to generalize better by increasing the spatial context, which is why we experimented with three kernel sizes (3×3, 5×5 and 7×7) to explore the impact of spatial context on model performance. We used the BCEWithLogitsLoss function from PyTorch as the loss function, combining binary cross-entropy loss with a sigmoid activation layer to compute the loss at each epoch. The loss values are unitless because they are a relative measure of the underlying distribution from training data rather than an absolute metric with physical units. Additionally, we calculated the F1 score using the equation $F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$ where $Precision = \frac{TP}{TP+FP}$ and $Recall = \frac{TP}{TP+FN}$ to estimate binary classification accuracy after each epoch.

(5) Finding optimal weights. Finally, we identified the epoch where the loss calculated on the test set was lowest and extracted the corresponding weights.

3.2.1. Pre-training

The pre-training workflow is shown in Figure 4a. We pre-trained the models on the Swedish dataset described in Section 2.1. First, we split the data into training (80%) and test (20%) sets. The training set was then augmented depending on the configuration (Table 1), meaning that models PT3U, PT5U and PT7U were trained on unaugmented data, while training data for PT3A, PT5A and PT7A was augmented. The model names were based on the configuration and are explained in Table A1. The test set was left unaugmented in all cases. After training for 250 epochs, the weights from the epoch with the lowest loss were deemed the best weights for that particular model and used as input for the corresponding fine-tuned model (Table 2).

3.2.2. Fine-tuning

The fine-tuning workflow resembled pre-training, except we augmented training data in all configurations (Figure 4b). The main difference in the fine-tuning workflow was in step 4, where the best weights from the corresponding pre-trained model were loaded and trained further on the Estonian dataset. We fine-tuned all six pre-trained models on Estonian data, resulting in six fine-tuned model configurations (Table 2). Each fine-tuned model's kernel size matched the pre-trained model from which we loaded the weights.

Similar to pre-trained models, each fine-tuned model was trained for 250 epochs, after which the best weights were determined based on the lowest loss value, resulting in six

Table 1. This is an overview of the model configurations we tested for pre-training. The acronyms of model names are explained in Table A1. We augmented training data for models PT3A, PT5A and PT7A, which increased the number of HPMF-label pairs to 5440 from the initial 1088.

		Training set			Test set		
Pre-trained model	Kernel size (m)	Number of images	Augmented	Country	Number of images	Augmented	Country
PT3U	3×3	1088	no	Sweden	272	no	Sweden
PT5U	5×5	1088	no	Sweden	272	no	Sweden
PT7U	7×7	1088	no	Sweden	272	no	Sweden
PT3A	3×3	5440	yes	Sweden	272	no	Sweden
PT5A	5×5	5440	yes	Sweden	272	no	Sweden
PT7A	7×7	5440	yes	Sweden	272	no	Sweden

Table 2. This is an overview of the model configurations we tested for fine-tuning. The acronyms of model names are explained in Table A1. We augmented Estonian training data in all cases, which increased the number of HPMF-label pairs to 290 from the initial 58 (Figure 4b). The kernel sizes of fine-tuned models matched those of pre-trained ones.

			Training set			Test set		
Fine-tuned model	Input pre-trained model	Kernel size (m)	Number of images	Augmented	Country	Number of images	Augmented	Country
FT3U	PT3U	3×3	290	yes	Estonia	14	no	Estonia
FT5U	PT5U	5×5	290	yes	Estonia	14	no	Estonia
FT7U	PT7U	7×7	290	yes	Estonia	14	no	Estonia
FT3A	PT3A	3×3	290	yes	Estonia	14	no	Estonia
FT5A	PT5A	5×5	290	yes	Estonia	14	no	Estonia
FT7A	PT7A	7×7	290	yes	Estonia	14	no	Estonia

sets of fine-tuned weights. We deemed the model with the lowest loss value and the highest F1 score the best overall model for ditch detection in Estonia. If multiple models had the same metrics, we preferred the model with the smaller kernel size since larger kernels require more computational resources. Finally, we used the best fine-tuned model to predict ditch labels across all 72 Estonian HPMF tiles in study areas A and B (Figure 1b) and compared the output with digitized true labels.

4. Results

4.1. Models pre-trained on Swedish data

The change in loss and F1 scores during pre-training for models outlined in Table 1 is shown in Figure 5. We evaluated the models' performance based on the test set's loss values and F1 scores. Except for PT5U, all models reached their lowest test loss within the first 100 epochs, after which the loss increased rapidly (Figure 5a), while the F1 score plateaued (Figure 5b). The lowest test loss values were similar regardless of model configuration, remaining between 0.014 and 0.016. The corresponding F1 scores indicated that models trained on augmented data performed better than unaugmented versions – F1 scores ranged from 0.786–0.788 for PT3A, PT5A and PT7A (Figure 5b).

Kernel size and data augmentation affected the total model training time. Training time increased with kernel size, e.g. training PT7U took over five times longer than PT3U (see Table A2 in the Appendix). Additionally, models trained on augmented data required significantly longer to finish compared to their unaugmented counterparts.

4.2. Models fine-tuned on Estonian data

Although augmented models performed better than unaugmented ones, we chose to fine-tune all six models to determine whether they behave similarly when trained on Estonian data. The change in test loss and F1 scores during fine-tuning for models outlined in Table 2 is shown in Figure 6. The lowest test loss values were higher in all cases than in pre-trained models, i.e. between 0.020 and 0.023 (Figure 6a). The lowest loss was achieved within the first 50 epochs, after which loss rapidly increased and F1 score

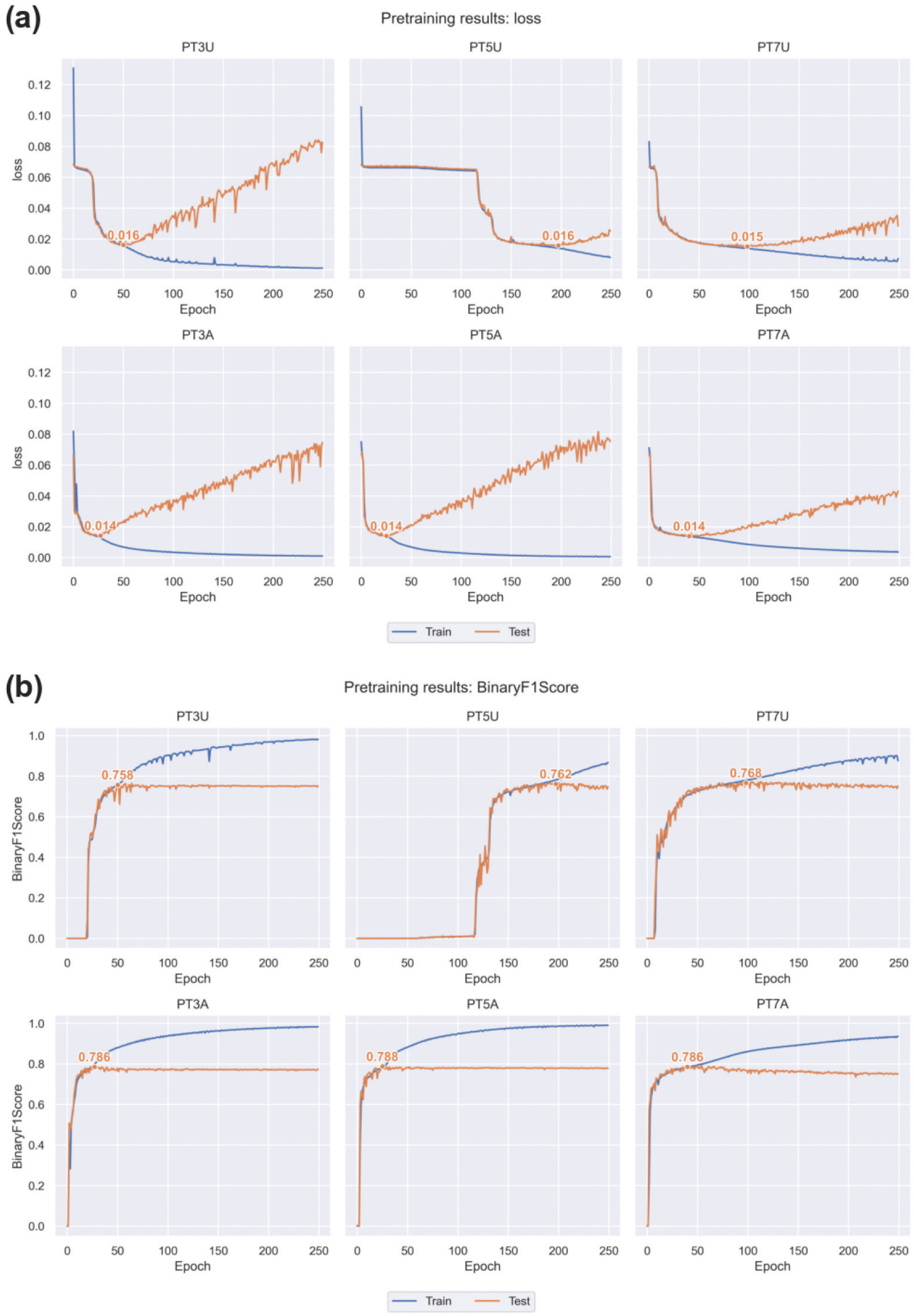


Figure 5. Results of pre-training for the model configurations given in Table 1. Change of loss (a) and F1 score (b) over 250 epochs. The lowest test loss value and corresponding F1 score have been labeled for each model.

plateaued (Figure 6b). F1 scores corresponding to the lowest test loss values ranged from 0.741 to 0.766, with model FT3A performing the best.

We labeled the F1 scores calculated on the training set for the first epoch in Figure 6b as baselines, which show how well the models would perform on Estonian data without fine-tuning. These baseline scores ranged from 0.570 to 0.677, indicating a reduced accuracy compared to Swedish data. While we did not detect any correlation between baseline score and kernel size, the augmented models FT3A and FT5A showed better performance than others. Comparing the training set baselines to the test set F1 scores shows how much the models gained from fine-tuning. Model FT7A had the most significant gain, with F1 increasing from 0.570 to 0.757, i.e. a 33% increase, while FT3A had the lowest increase (13%).

4.3. Selecting and assessing the best fine-tuned model

We selected the best-performing fine-tuned model to investigate its ditch detection capabilities in Estonia. We chose FT3A as it had the highest F1 score (0.766) while having the smallest kernel size (3 × 3). We used FT3A to predict probabilities of ditch pixels in all tiles in study areas A and B, shown in Figure 1 and compared the output to digitized ditches, i.e. true labels. We labeled pixels above the 0.5 (50%) probability threshold as ditches in the prediction output and used them as a benchmark for assessing model performance. Furthermore, we computed accuracy metrics per land use class in sites A and B to quantify potential differences in model performance.

Examples of predictions made with FT3A are shown in Figure 7. Here, tile A8 represents a forested area, while tile B19 is the edge of a peat extraction site. When using the 0.5 probability threshold in tile B19, predicted labels mostly align with the digitized peatland drainage ditches (Figure 7b). However, the individual ditch segments are incomplete and discontinuous in some areas. This aligns with the quantitative results in Table 3, where peatland achieved the highest overall F1 score (0.822). In tile A8, which is more forested, only the larger ditches could be extracted based on the 0.5 threshold, while the denser ditch network in the top left corner of the tile remained undetected (Figure 7a). This is consistent with lower F1 scores retrieved for forest (0.734 in site A and 0.752 in B). In addition, the model also performed worse in arable land (F1 = 0.719 in A and F1 = 0.779 in B) compared to peatland.

We also experimented with lowering the probability threshold to improve the completeness of the predicted labels. Here, we highlight ditch pixels extracted based on the 0.1 (10%) probability threshold as an example. In general, lowering the threshold improved continuity in both tiles. However, in tile B19, false positive artefacts were detected in the lower right corner of the area. In tile A8, on the other hand, lowering the threshold to 0.1 significantly improves the completeness of the output labels.

Finally, we further investigated areas where the model had predicted labels that were not present in training data to assess whether these were truly false positives or instead missed during digitization (Figure 8). We refer to these as possible false positive (PFP) cases. The majority of PFPs were located in dense forests or near

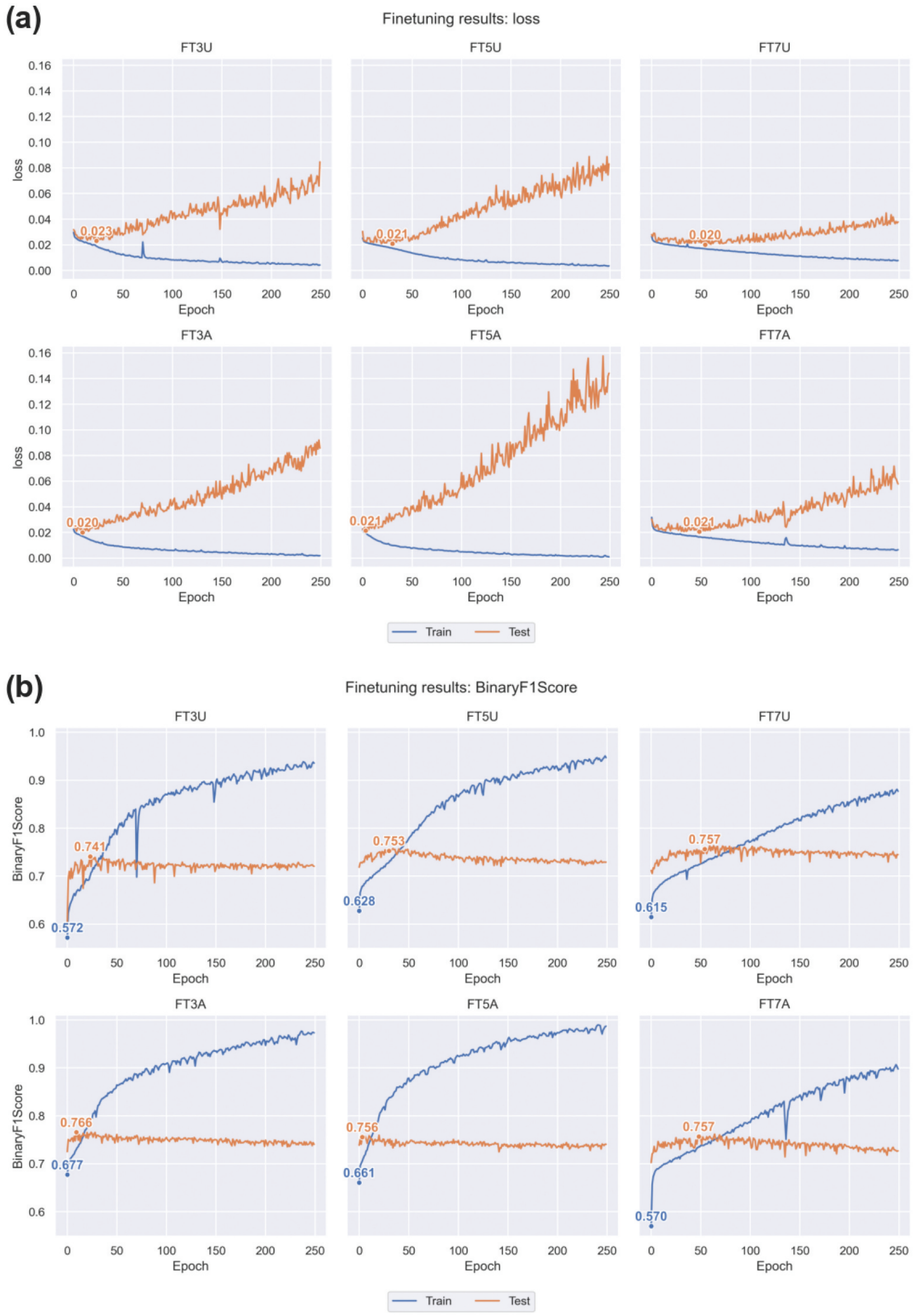


Figure 6. Results of fine-tuning for the model configurations given in Table 2 over 250 epochs. Change of loss (a) and F1 score (b) over 250 epochs. The lowest test loss value and corresponding F1 score have been labeled for each model. In addition, the F1 score calculated on the training set is labeled in blue for the first epoch.

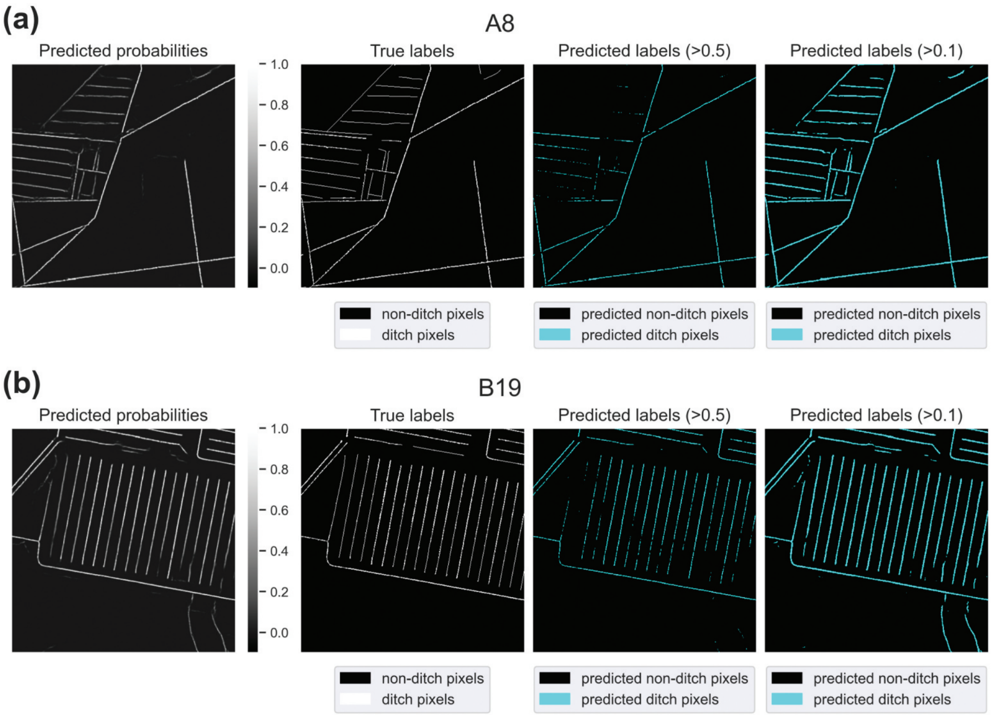


Figure 7. We used model FT3A (F1 = 0.766) to predict probabilities and extract labels based on the 0.5 and 0.1 probability thresholds in tiles highlighted in blue in Figure 1b. (a) Predicted probabilities, true labels and predicted labels for tile A8. (b) Predicted probabilities, true labels and predicted labels for tile B19.

Table 3. Precision, recall, and F1 score for different land use classes in test sites A and B computed based on predictions made with FT3A using the 0.5 probability threshold. A dash indicates that the land use class was absent in the corresponding site.

Site	Land use	Precision	Recall	F1 score
A	Arable	0.875	0.610	0.719
	Forest	0.852	0.645	0.734
	Peatland	—	—	—
B	Arable	0.840	0.727	0.779
	Forest	0.848	0.675	0.752
	Peatland	0.827	0.818	0.822

roads. For example, in tile A36, the PFPs on the eastern edge of the image were small forest ditches along the treeline that were not picked up from the HPMF raster during digitizing (Figure 8a). In tile B12, most PFPs were along roads, where it was challenging to distinguish ditches due to general elevation changes along road embankments (Figure 8b). Nevertheless, some PFPs were likely true false positives, e.g. logging tracks in forested areas.

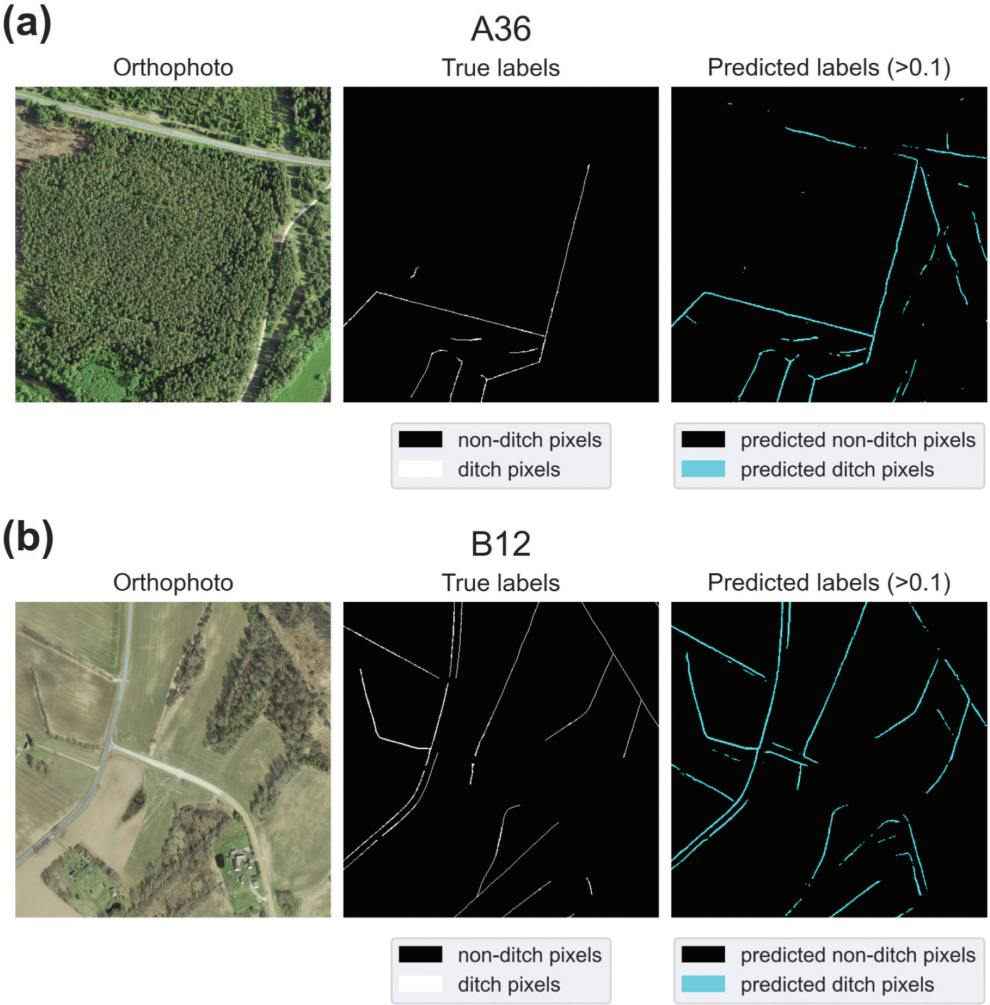


Figure 8. Examples of predicted roadside and forest ditches absent in true labels. (a) The orthophoto, true labels and predicted labels for tile A36. (b) The orthophoto, true labels and predicted labels for tile B12.

5. Discussion

5.1. Effectiveness of transfer learning for ditch detection

The fine-tuning results showed that transfer learning based on U-Net can be a viable method for detecting ditches in Estonia. We demonstrated that an existing labeled dataset from a data-rich region (Lidberg et al., 2023) could be successfully used to pre-train a model, which was then fine-tuned on a small (18 km^2) localized dataset from Estonia. This approach significantly reduced the amount of labels required from Estonia, where no suitable dataset existed. Training a similar DL model from scratch would have required more extensive labeling efforts, generally the most time-consuming and resource-intensive parts of DL projects (Flood et al., 2019; Waldner & Diakogiannis, 2020).

Transfer learning has shown to be a valuable method for image segmentation tasks in data-scarce areas (van Opbroek et al., 2015; Wurm et al., 2019). The comparison of fine-tuned models revealed that depending on model configuration, fine-tuning improved the performance of pre-trained models between 13% and 33%, indicating that reasonable gains were made by fine-tuning on only a limited number of Estonian HPMF-label pairs. Although fine-tuning provided measurable gains, we note that the loss function converged relatively quickly regardless of model configuration, which indicates that Swedish and Estonian datasets were sufficiently similar. Therefore, improvements made by fine-tuning were less pronounced than they would have been if differences between the hydrological and topographic conditions of the two countries were more significant. This aligns with results reported in Lidberg et al. (2023), where a U-Net model trained in Sweden performed reasonably in countries around the Baltic Sea, i.e. Finland, Latvia and Poland.

5.2. Effects of model configuration on performance

We further examined the effects of model configuration, i.e. kernel size and data augmentation on model performance. Although previous ditch detection studies have used both 3 × 3 (Lidberg et al., 2023) and 5 × 5 (Koski et al., 2023) pixel kernels, the effects of different kernel sizes have not been compared. Our experiments revealed no apparent connection between kernel size and performance, as loss values and F1 scores were similar in all cases. As larger kernels require more resources when training, the smallest kernel was sufficient for our use case.

In Robb et al. (2023) and Du et al. (2024), the authors applied basic data augmentation to increase the number of labeled ditches for training. We implemented similar augmentation techniques (rotations and horizontal flipping) and compared the performance of models pre-trained on unaugmented and augmented datasets. Pre-training results showed that augmentation had more impact on model performance than kernel size, with augmented models being slightly more accurate. DL models such as U-Net generally depend on the quantity and diversity of training samples to generalize better and reduce overfitting (Malik et al., 2021; Zhu et al., 2017). Our findings confirmed that basic augmentation can successfully supplement manually produced labels.

While augmenting the training data improved performance, it also significantly increased the pre-training time. For example, the pre-training of model PT3A took 20 hours, i.e. over five times longer than its unaugmented counterpart PT3U. We considered this trade-off acceptable as it resulted in a lower loss and deemed the corresponding fine-tuned model FT3A the best configuration for ditch detection. Still, increasing training data size might not always be feasible considering that training time and computing costs are often limiting factors (Flood et al., 2019).

5.3. Comparison to previous studies and practical implications

We used loss values and F1 scores to determine the optimal model configuration, ultimately identifying FT3A (F1 = 0.766) as the most effective model for detecting ditches in Estonia. In previous studies, the reported F1 scores were 0.71 (Du et al., 2024), 0.79 (Robb et al., 2023) and 0.812 (Koski et al., 2023), respectively. While the F1 score was not reported in Lidberg et al. (2023), their model's Matthews correlation coefficient (MCC) was 0.78. Therefore, FT3A achieved an accuracy similar to the models above.

We applied FT3A to Estonian training tiles to assess its predictive capabilities. We found that when using the 0.5 probability threshold for extracting labels, the model was relatively conservative in its predictions, producing minimal PFPs. This was further confirmed by lowering the probability threshold to 0.1, which increased the completeness of the ditch network significantly while not adding many PFPs. The accuracy was generally better in the peat extraction area, where most ditches were spaced regularly and more likely to have even width and depth, meaning they were more visible in the HPMF images. However, like in Robb et al. (2023), a certain level of under-segmentation was present, indicated by the discontinuity of some peatland ditches.

The model was less accurate in forests where many predicted labels became visible only after lowering the threshold to 0.1. As trees obscure forest ditches, the HPMF image in such areas is noisier, which hinders detecting the correct ditch channel, and the model is likely to under-segment. Previous studies have also noted that many forest ditches are partially overgrown, resulting in uneven depth throughout the channel (Koski et al., 2023; Lidberg et al., 2023). Although the model produced minimal PFPs, some true false positive predictions were likely made. In particular, roadside ditches have previously caused false positive predictions as detecting them is difficult due to abrupt elevation changes near road embankments (Du et al., 2024; Koski et al., 2023; Wang & Fareed, 2021). Nevertheless, predicted labels in problematic areas like forests and roadsides require fieldwork to be appropriately validated since distinguishing the causes of prediction errors from orthophotos or HPMF images is not always possible.

While our results demonstrate the effectiveness of transfer learning for ditch detection in data-scarce regions, certain limitations should be acknowledged. The Estonian fine-tuning dataset covered only a small geographic area, which can lead to overfitting and reducing performance outside of the selected study sites. We tried to mitigate this by applying data augmentation and validating performance across different land use classes. However, testing in geographic regions with more diverse ditch characteristics and environmental conditions is necessary to assess its generalizability fully. We used a relatively high resolution (1 m) DTM as model input, which might not be uniformly available in other countries. Therefore, future work could include integrating multi-source input data, such as orthophotos, to improve its applicability across large areas. Despite these limitations, our findings highlight the potential of transfer learning for scalable and automated ditch detection, particularly in areas with limited labeled training data.

6. Conclusions

Drainage ditches are essential for managing water flow and mitigating flood risks in temperate climates. However, accurately mapping ditches from remote sensing imagery is challenging due to their narrow and shallow nature, especially in forests where tree cover obstructs their visibility. Traditional ditch mapping methods are time-consuming and labor-intensive and often fail to capture the full extent of these features. We demonstrated that transfer learning, utilizing a U-Net model pre-trained on existing data from a region with similar topographic and hydrological conditions, can effectively reduce the need for extensive local training data, overcoming one major bottleneck for applying DL models in regions with limited labeled data availability. Our approach resulted in measurable accuracy gains after fine-tuning on a small Estonian dataset, underscoring the potential of transfer learning for environmental mapping tasks in regions with limited labeled data.

Our experiments with different kernel sizes and data augmentation techniques provided valuable insights into optimizing model configurations for ditch detection. We found that increasing the kernel size beyond the default 3×3 did not enhance accuracy, indicating that a smaller kernel was sufficient for the task. On the other hand, basic augmentation techniques helped improve model performance, highlighting the importance of diverse and plentiful training samples when training DL models. However, there was a trade-off between training time and improved accuracy, pointing to the need for a balanced approach that considers both model performance and computational efficiency.

Acknowledgements

We acknowledge the support of grant number LLTOM21539 from the Estonian Environmental Research Centre and grant number LLTOM22387 from the State Forest Management Centre. While these grants did not directly fund the present study, their recipients provided valuable advice during the research process. We hope that the outcomes of this study will contribute to the broader goals of these funding bodies.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was funded by the Estonian Research Agency [grant number PRG1764, PSG841], Estonian Ministry of Education and Research, Centre of Excellence for Sustainable Land Use [TK232], Estonian Environment Agency [grant MULD2] and by the European Union [ERC, WaterSmartLand, 101125476]. However, views and opinions expressed are those of the author(s) only and do not necessarily reflect those of the European Union or the European Research Council Executive Agency. Neither the European Union nor the granting authority can be held responsible.

Notes on contributors

Holger Virro, PhD, is a Research Fellow in Geoinformatics at the University of Tartu, Estonia. He works on applying machine learning methods to solve large-scale geospatial problems, focusing on modeling different environmental indicators (soil, water quality, land use). He also leverages deep learning for detecting landscape elements from remote sensing data

Alexander Kmoch, PhD, is an Associate Professor in Geoinformatics and a Distributed Spatial Systems Researcher with many years of experience in geospatial data management and web- and cloud-based geoprocessing with a particular focus on land use, soils, hydrology, hydrogeology and water quality data. His interests include OGC standards and web services for environmental and geo-scientific data sharing, modelling workflows and interactive geo-scientific visualization.

William Lidberg, PhD, is an Associate Senior Lecturer at the Department of Forest Ecology and Management. He leads an interdisciplinary research group that looks at opportunities and challenges with artificial intelligence in forests, mainly related to mapping small streams and shallow water from elevation data.

Merle Muru, PhD, is a Research Fellow in Geoinformatics with a primary interest in landscape modelling. Her PhD and postdoctoral studies were on palaeogeography, coastal landscape, and sea level change. She focuses on terrain modelling, digital elevation data, raster and point cloud analyses.

Wai Tik Chan, MSc, is a Junior Research Fellow in Geoinformatics and holds an MSc in Computer Science with a major in machine learning. His main focus is web application development and integration of geoinformatics data under the Discrete Global Grid System (DGGS).

Desalew Meseret Moges, PhD, is a Research Fellow in Applied Geoinformatics. His main research interest revolves around environmental modelling, with a particular emphasis on hydrology, soil, and land use. He also has a keen interest in remote sensing and geospatial visualization.

Evelyn Uuemaa, PhD, is a Professor in Geoinformatics and head of the Landscape Geoinformatics Lab at the University of Tartu. Her research interests revolve around the interactions of land use/land cover and water quality, landscape changes and geospatial modelling with machine learning. She is also interested in geospatial visualization.

ORCID

Holger Virro http://orcid.org/0000-0001-6110-5453

Data availability statement

The data (Virro, Kmoch, Lidberg, Muru, et al., 2025) and code (Virro, Kmoch, Lidberg, Chan, et al., 2025) that support the findings of this study are openly available on Zenodo. Data DOI: https://doi.org/10.5281/zenodo.14893003, code DOI: https://doi.org/10.5281/zenodo.14892850 (with an associated GitHub repository).

References

Bailly, J. S., Lagacherie, P., Millier, C., Puech, C., & Kosuth, P. (2008). Agrarian landscapes linear features detection from LiDAR: Application to artificial drainage networks. *International Journal of Remote Sensing*, *29*(12), 3489–3508. https://doi.org/10.1080/01431160701469057

Du, L., McCarty, G. W., Li, X., Zhang, X., Rabenhorst, M. C., Lang, M. W., Zou, Z., Zhang, X., & Hinson, A. L. (2024). Drainage ditch network extraction from lidar data using deep convolutional neural networks in a low relief landscape. *Journal of Hydrology*, *628*, 130591. https://doi.org/10.1016/j.jhydrol.2023.130591

Du, L., McCarty, G. W., Zhang, X., Lang, M. W., Vanderhoof, M. K., Li, X., Huang, C., Lee, S., & Zou, Z. (2020). Mapping forested wetland inundation in the Delmarva Peninsula, USA using deep convolutional neural networks. *Remote Sensing*, *12*(4), 644. https://doi.org/10.3390/rs12040644

Estonian Land Board. (2020). *Elevation data*.

Falk, T., Mai, D., Bensch, R., Çiçek, Ö., Abdulkadir, A., Marrakchi, Y., Böhm, A., Deubner, J., Jäckel, Z., Seiwald, K., Dovzhenko, A., Tietz, O., Dal Bosco, C., Walsh, S., Saltukoglu, D., Tay, T. L., Prinz, M., Palme, K., … Ronneberger, O. (2019). U-Net: Deep learning for cell counting, detection, and morphometry. *Nature Methods*, *16*(1), 67–70. https://doi.org/10.1038/s41592-018-0261-2

Flood, N., Watson, F., & Collett, L. (2019). Using a U-net convolutional neural network to map woody vegetation extent from high resolution satellite imagery across Queensland, Australia. *International Journal of Applied Earth Observation and Geoinformation*, *82*, 101897. https://doi.org/10.1016/j.jag.2019.101897

Flyckt, J., Andersson, F., Lavesson, N., Nilsson, L., & Gren, Å. (2022). Detecting ditches using supervised learning on high-resolution digital elevation models. *Expert Systems with Applications*, *201*, 116961. https://doi.org/10.1016/j.eswa.2022.116961

Huang, Z., Qian, H., Wang, X., Lin, D., Wang, J., & Xie, L. (2023). Graph neural network-based identification of ditch matching patterns across multi-scale geospatial data. *Geocarto International*, *38*(1), 2294900. https://doi.org/10.1080/10106049.2023.2294900

Kattenborn, T., Eichel, J., & Fassnacht, F. E. (2019). Convolutional neural networks enable efficient, accurate and fine-grained segmentation of plant species and communities from high-resolution UAV imagery. *Scientific Reports*, *9*(1), 17656. https://doi.org/10.1038/s41598-019-53797-9

Kiss, K., Malinen, J., & Tokola, T. (2015). Forest road quality control using ALS data. *Canadian Journal of Forest Research*, *45*(11), 1636–1642. https://doi.org/10.1139/cjfr-2015-0067

Koski, C., Kettunen, P., Poutanen, J., Zhu, L., & Oksanen, J. (2023). Mapping small watercourses from DEMs with deep learning—exploring the causes of false predictions. *Remote Sensing*, *15*(11), 2776. https://doi.org/10.3390/rs15112776

Längkvist, M., Kiselev, A., Alirezaie, M., & Loutfi, A. (2016). Classification and segmentation of satellite orthoimagery using convolutional neural networks. *Remote Sensing*, *8*(4), 329. https://doi.org/10.3390/rs8040329

Laurén, A., Palviainen, M., Launiainen, S., Leppä, K., Stenberg, L., Urzainki, I., Nieminen, M., Laiho, R., & Hökkä, H. (2021). Drainage and stand growth response in peatland forests—description, testing, and application of mechanistic peatland simulator SUSI. *Forests*, *12*(3), 293. https://doi.org/10.3390/f12030293

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Levavasseur, F., Lagacherie, P., Bailly, J., Biarnès, A., & Colin, F. (2015). Spatial modeling of man-made drainage density of agricultural landscapes. *Journal of Land Use Science*, *10*(3), 256–276. https://doi.org/10.1080/1747423X.2014.884644

Lidberg, W., Paul, S. S., Westphal, F., Richter, K. F., Lavesson, N., Melniks, R., Ivanovs, J., Ciesielski, M., Leinonen, A., & Ågren, A. M. (2023). Mapping drainage ditches in forested landscapes using deep learning and aerial laser scanning. *Journal of Irrigation & Drainage Engineering*, *149*(3), 04022051. https://doi.org/10.1061/JIDEDH.IRENG-9796

Lindsay, J. B. (2014). The Whitebox geospatial analysis tools project and open-access GIS. In Jane Drummond (Ed.), *Proceedings of the GIS research UK 22nd annual conference* (pp. 16–180). Geomatic Group.

Lõhmus, A., Remm, L., & Rannap, R. (2015). Just a ditch in forest? Reconsidering draining in the context of sustainable forest management. *BioScience*, *65*(11), 1066–1076. https://doi.org/10.1093/biosci/biv136

Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In D. Forsyth (Ed.), *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431–3440). Institute of Electrical and Electronics Engineers (IEEE).

Malik, K., Robertson, C., Braun, D., & Greig, C. (2021). U-Net convolutional neural network models for detecting and quantifying placer mining disturbances at watershed scales. *International Journal of Applied Earth Observation and Geoinformation*, *104*, 102510. https://doi.org/10.1016/j.jag.2021.102510

Marcel, S., & Rodriguez, Y. (2010). Torchvision the machine-vision package of torch. In A. del Bimbo (Ed.), *Proceedings of the 18th ACM international conference on multimedia* (pp. 1485–1488). Association for Computing Machinery. https://doi.org/10.1145/1873951.1874254

Melniks, R., Ivanovs, J., Lazdins, A., & Makovskis, K. (2022). Mapping drainage ditches in agricultural landscapes using LiDAR data. *Agronomy Research*, *20*, 318–325. https://doi.org/10.15159/AR.22.012

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., & Chintala, S. (2019). PyTorch: An imperative style, high-performance deep learning library. In H. M. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché, E. A. Fox, & Roman Garnett (Eds.), *Advances in neural information processing systems* (pp. 8024–8035). Curran Associates Inc.

Peacock, M., Granath, G., Wallin, M. B., Högbom, L., & Futter, M. N. (2021). Significant emissions from forest drainage ditches—an unaccounted term in anthropogenic greenhouse gas inventories? *Journal of Geophysical Research: Biogeosciences*, *126*(10), e2021JG006478. https://doi.org/10.1029/2021JG006478

Peacock, M., Ridley, L. M., Evans, C. D., & Gauci, V. (2017). Management effects on greenhouse gas dynamics in fen ditches. *Science of the Total Environment*, *578*, 601–612. https://doi.org/10.1016/j.scitotenv.2016.11.005

Pirotti, F., & Tarolli, P. (2010). Suitability of LiDAR point density and derived landform curvature maps for channel network extraction. *Hydrological Processes*, *24*(9), 1187–1197. https://doi.org/10.1002/hyp.7582

QGIS Development Team. (2024). *QGIS geographic information system*.

Rapinel, S., Hubert-Moy, L., Clément, B., Nabucet, J., & Cudennec, C. (2015). Ditch network extraction and hydrogeomorphological characterization using LiDAR-derived DTM in wetlands. *Hydrology Research*, *46*(2), 276–290. https://doi.org/10.2166/nh.2013.121

Ratt, A. (1985). *Mõnda maaviljeluse arengust Eestis läbi aegade*. Valgus.

Robb, C., Pickard, A., Williamson, J. L., Fitch, A., & Evans, C. (2023). Peat drainage ditch mapping from aerial imagery using a convolutional neural network. *Remote Sensing*, *15*(2), 499. https://doi.org/10.3390/rs15020499

Roelens, J., Höfle, B., Dondeyne, S., Van Orshoven, J., & Diels, J. (2018). Drainage ditch extraction from airborne LiDAR point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, *146*, 409–420. https://doi.org/10.1016/j.isprsjprs.2018.10.014

Ronneberger, O., Fischer, P., & Brox, T. (2015). *U-Net: Convolutional networks for biomedical image segmentation*. *arXiv:1505.04597*. https://doi.org/10.48550/arXiv.1505.04597

Stanislawski, L. V., Survila, K., Wendel, J., Liu, Y., & Buttenfield, B. P. (2018). An open source high-performance solution to extract surface water drainage networks from diverse terrain conditions. *Cartography and Geographic Information Science*, *45*(4), 319–328. https://doi.org/10.1080/15230406.2017.1337524

University of Tartu. (2018). *UT rocket*. https://doi.org/10.23673/PH6N-0144

van der Walt, S., Schönberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., & Yu, T. (2014). Scikit-image: Image processing in Python. *PeerJ*, *2*, e453. https://doi.org/10.7717/peerj.453

van Opbroek, A., Ikram, M. A., Vernooij, M. W., & de Bruijne, M. (2015). Transfer learning improves supervised image segmentation across imaging protocols. *IEEE Transactions on Medical Imaging*, *34*(5), 1018–1030. https://doi.org/10.1109/TMI.2014.2366792

Virro, H., Kmoch, A., Lidberg, W., Chan, W. T., & Uuemaa, E. (2025). Code supplement: Detection of drainage ditches from LiDAR DTM using U-Net and transfer learning. *Zenodo*. https://doi.org/10.5281/zenodo.14892851

Virro, H., Kmoch, A., Lidberg, W., Muru, M., Chan, W. T., Moges, D. M., & Uuemaa, E. (2025). *Data supplement: Detection of drainage ditches from LiDAR DTM using U-Net and transfer learning*. https://doi.org/10.5281/zenodo.14893004

Waldner, F., & Diakogiannis, F. I. (2020). *Deep learning on edge: Extracting field boundaries from satellite images with a convolutional neural network*. *arXiv: 1910.12023*.

Wang, C. K., & Fareed, N. (2021). Mapping drainage structures using airborne laser scanning by incorporating road centerline information. *Remote Sensing*, *13*(3), 463. https://doi.org/10.3390/rs13030463

Wu, W., Niu, X., Yan, Z., Li, S., Comer-Warner, S. A., Tian, H., Li, S. L., Zou, J., Yu, G., & Liu, C. Q. (2023). Agricultural ditches are hotspots of greenhouse gas emissions controlled by nutrient input. *Water Research*, *242*, 120271. https://doi.org/10.1016/j.watres.2023.120271

Wurm, M., Stark, T., Zhu, X. X., Weigand, M., & Taubenböck, H. (2019). Semantic segmentation of slums in satellite images using transfer learning on fully convolutional neural networks. *ISPRS Journal of Photogrammetry and Remote Sensing*, *150*, 59–69. https://doi.org/10.1016/j.isprsjprs.2019.02.006

Zhang, H., Loáiciga, H. A., Feng, L., He, J., & Du, Q. (2021). Setting the flow accumulation threshold based on environmental and morphologic features to extract river networks from digital elevation models. *ISPRS International Journal of Geo-Information*, *10*(3), 186. https://doi.org/10.3390/ijgi10030186

Zhu, X. X., Tuia, D., Mou, L., Xia, G. S., Zhang, L., Xu, F., & Fraundorfer, F. (2017). Deep learning in remote sensing: A comprehensive review and list of resources. *IEEE Geoscience and Remote Sensing Magazine*, *5*(4), 8–36. https://doi.org/10.1109/MGRS.2017.2762307