

# Improved Difference Images for Change Detection Classifiers in SAR Imagery Using Deep Learning

Janne Alatalo<sup>1</sup>, Tuomo Sipola<sup>1</sup>, and Mika Rantonen<sup>1</sup>

**Abstract**—Satellite-based synthetic aperture radar (SAR) images can be used as a source of remote sensed imagery regardless of cloud cover and day–night cycle. However, the speckle noise and varying image acquisition conditions pose a challenge for change detection classifiers. This article proposes a new method of improving SAR image processing to produce higher quality difference images (DIs) for the classification algorithms. The method is built on a neural network-based mapping transformation function that produces artificial SAR images from a location in the requested acquisition conditions. The inputs for the model are: previous SAR images from the location, imaging angle information from the SAR images, digital elevation model, and weather conditions. The method was tested with data from a location in North-East Finland by using Sentinel-1 SAR images from the European Space Agency (ESA), weather data from Finnish Meteorological Institute (FMI), and a digital elevation model from National Land Survey of Finland (NLS). In order to verify the method, changes to the SAR images were simulated, and the performance of the proposed method was measured using experimentation where it gave substantial improvements to performance when compared to a more conventional method of creating DIs.

**Index Terms**—Change detection, mapping transformation function, remote sensing, Sentinel-1, synthetic aperture radar (SAR), U-Net.

## I. INTRODUCTION

REMOTE sensing change detection can be used for many purposes, such as damage assessment after a natural disaster [1], [2], [3], detection of forest damages after a storm [4], [5], and monitoring deforestation and glacier melting [6], [7], to name only a few. Change detection works by comparing two images that have been captured at different dates in the same geographical location and finding the areas that have changed during the time between the acquisitions [8]. Different platforms can be used to image the terrain, such as airplanes and satellites, however, only satellites provide the advantage of continuously monitoring the whole planet [9]. The revisit time of some satellite systems can be as short as only a few days,

and the images are available from anywhere on the planet. This makes the satellite images a useful source of remote sensing data for change detection applications. Some space agencies, such as the European Space Agency (ESA), provide some of the satellite images for anybody to download and use [10]. The ease of acquiring the data further facilitates the development of change detection systems that are based on the satellite remote sensing techniques. The images from the satellites are captured using either optical or radar sensors, with radar having the advantage of piercing the cloud layer, thus enabling it to work in various weather conditions [9]. However, the radar satellites have their disadvantages as well. The resolution of the images is not as good as what the optical instruments can produce. The resolution of the radar images is defined by the antenna length and the frequency band of the radar signal. To enable higher resolution images, the satellites use the synthetic aperture radar (SAR) technique, where the satellite movement over the ground is utilized to synthesize virtual aperture that is longer than the physical antenna on the satellite [11]. However, even with the SAR technique the radar images are lower resolution when compared to the optical images. ESA has the Sentinel-1 mission with two SAR satellites that operate on the C-band and have a spatial resolution of around  $5 \times 20$  m [12]. Likewise, speckle noise reduces the quality of the SAR imagery. SAR images always have a grainy look from the speckle, which is random noise that is always present in the images. Despite the shortcomings of the SAR imagery, they are commonly used in remote sensing change detection [13], [14], [15], [16].

One approach to implement a change detection system, that is generally used in unsupervised change detection, is to proceed in steps [17]. Fig. 1 illustrates this method. The images are first preprocessed to make them comparable to each other. Then, two images from the same location, that are captured at different times, are used to produce a difference image (DI) using an algebraic operation like subtraction, ratio, or log ratio. Finally, the DI is analyzed by a classifier algorithm to produce a change map that indicates the changed regions. The preprocessing step is crucial for this method to work well. The issue with the speckle noise is a commonly recognized problem with change detection on SAR imagery [13], [15], [16], and to mitigate the issue, noise suppression algorithms are used in the image preprocessing step. However, it is impossible to remove the noise completely, thus the DI also includes noise that causes misclassifications in the classification step. Likewise, other image properties that influence image comparability have an effect on the quality of the DI.

Manuscript received 31 March 2023; revised 22 June 2023 and 26 September 2023; accepted 7 October 2023. Date of publication 16 October 2023; date of current version 27 October 2023. This work was supported in part by the Regional Council of Central Finland/Council of Tampere Region and European Regional Development Fund under Grant A76982, in part by the REACT-EU Instrument as part of the European Union's response to the COVID-19 pandemic under Grant A77973, and in part by the European Union and the Regional Council of Central Finland with the Just Transition Fund under Grant J10075. (Corresponding author: Janne Alatalo.)

The authors are with the Institute of Information Technology, JAMK University of Applied Sciences, 40100 Jyväskylä, Finland (e-mail: janne.alatalo@jamk.fi; tuomo.sipola@jamk.fi; mika.rantonen@jamk.fi).

Digital Object Identifier 10.1109/TGRS.2023.3324994

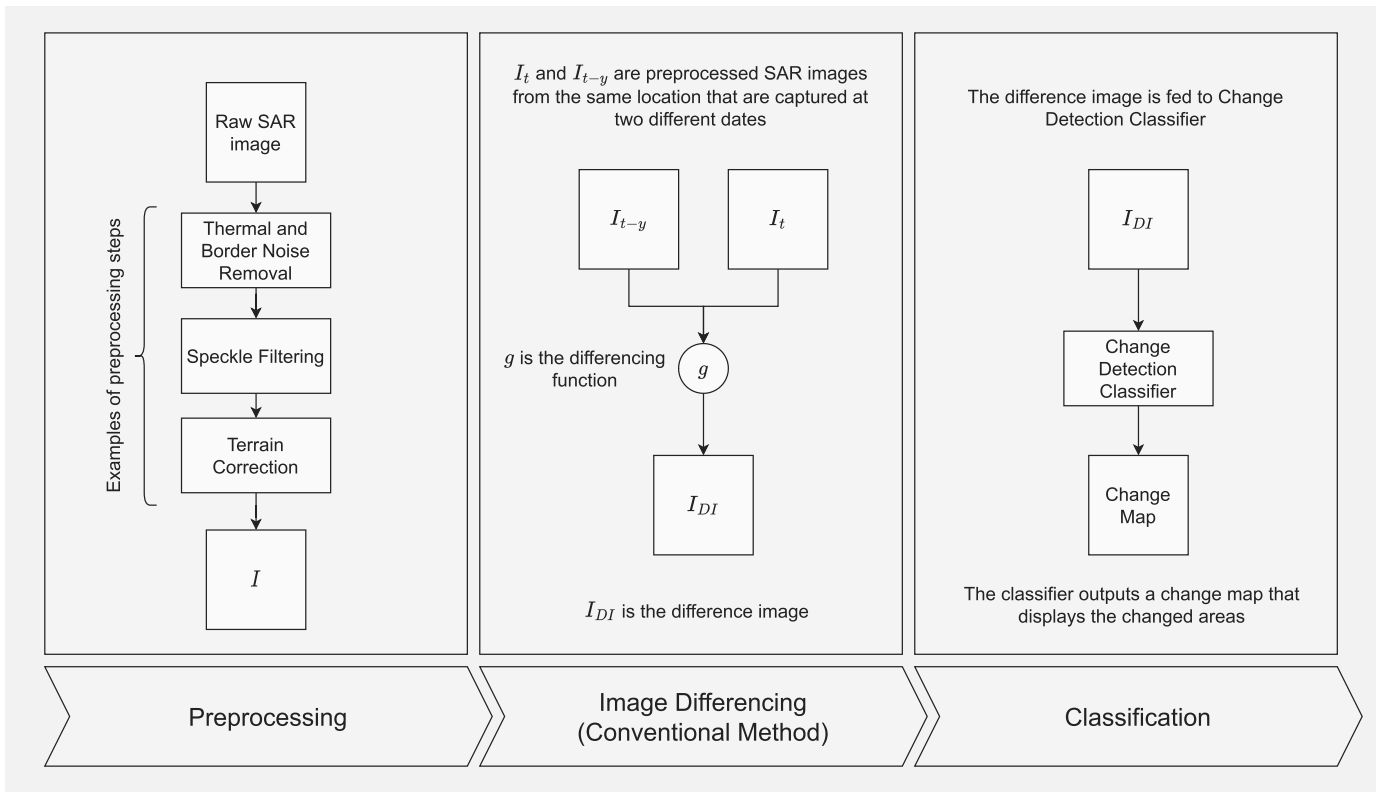


Fig. 1. Change detection is often implemented in three distinct steps. The first step is to make the images more comparable to each other using a preprocessing pipeline. The preprocessed images are then used to create DIs ( $I_{DI}$ ) using a function  $g$  that is often an algebraic operation, such as subtraction, ratio, or log ratio.  $I_{DI}$  is then used as an input to a change detection classifier that produces the change map that displays the changed areas. The figure illustrates the conventional method of producing the DIs by using two SAR images that are captured from the location on two different dates.

This includes properties such as the satellite orbit direction, incidence angle, and ground moisture content. The satellite does not capture the image from the same angle during every revisit. In the case of the ESA Sentinel-1 satellites, the satellite can be flying from North to South, or from South to North, during the image acquisition, and the satellite orbit can be higher or lower with respect to the horizon from the ground perspective between the overflies. The satellite imaging angle influences how the radar signal backscatters from the ground features [18], which results in images taken from different imaging angles likely producing lower quality DI than images taken from the same imaging angle. Likewise, ground weather conditions can influence the DI quality. Soil moisture content changes the dielectric constant of the soil, thus changing the backscatter intensity of the radar signal [19]. Images that are taken in similar weather conditions are likely to produce better quality DI when compared to images that are taken in different weather conditions. One solution to improve the DI quality is to favor images with similar acquisition conditions when selecting the images that are used to produce the DI. However, this is not always possible. Predicting vegetation properties from atmospheric conditions has been identified as one of the potential tasks to benefit from neural networks with spatio-temporal context. However, prescriptive assumptions could limit this use. The combination of process-based modeling and data-driven machine learning approaches could help when the physical models need support from data [20]. Adding domain knowledge to the physical layers is a step toward

hybrid modeling. Indeed, deep neural networks have been used to extract spatial and frequency features from SAR images. Using these features, the classification of objects or areas in SAR images is also suitable as a deep learning task [21].

The contribution of this article is a new method to produce better quality DIs. This is achieved by using a neural network-based mapping transformation function preprocessing step that factors in the image acquisition conditions of the SAR images, which improves the comparability of the SAR images. Existing research about SAR image preprocessing has focused on removing speckle noise from the images [22], [23], or correcting the incidence angle variation [24], [25]. However, to the best of the knowledge of the authors, this is the first time when the comparability of the SAR images is improved by taking into account the overall image acquisition conditions using a neural network-based preprocessing step. Project code is available on GitHub.<sup>1</sup>

## II. MATERIALS AND METHODS

### A. Proposed Method

Fig. 2 illustrates the overall architecture of the proposed method. It replaces the conventional method that is illustrated in Fig. 1 image differencing step. The idea of the proposed method is to improve the SAR image comparability by considering the acquisition conditions of the SAR images. The proposed method utilizes a mapping transformation function

<sup>1</sup><https://github.com/janne-alatalo/sar-change-detection>

that creates artificial SAR images in the requested acquisition conditions. The mapping transformation function  $\mathcal{F}$  is a neural network model that is trained to predict the SAR image  $I_t$  at the time  $t$ . The neural network output  $\hat{I}_t$  is the artificial SAR image that is created in the acquisition conditions of  $I_t$ , therefore it should be more comparable to  $I_t$  than previous SAR images from the location that might have been captured in different acquisition conditions.

The model input consists of three distinct features, which are: the previous SAR images from the location; the acquisition conditions of the SAR images (including at time  $t$ ); and the digital elevation map (DEM) from the location. The objective of the neural network model is to learn to replicate the SAR image at the time  $t$ . The only information from the time  $t$  in the model input are the image acquisition conditions of  $I_t$ . This means that for the model to be able to replicate  $I_t$ , it needs to learn to map the information contained in the previous SAR images and the DEM to the image acquisition conditions of  $I_t$ . With an ideal model that could perfectly replicate  $I_t$ ,  $\hat{I}_t$  and  $I_t$  would be identical if nothing has changed between the image acquisition of  $I_{t-1}$  and  $I_t$ , however,  $\hat{I}_t$  would be missing the change if something had changed after the previous image acquisition since the information of the change is not included in the model input data. In practice the SAR images include random noise that is impossible to replicate accurately, and the acquisition conditions are not accurate enough for perfect replication of  $I_t$ , therefore  $\hat{I}_t$  only approximates  $I_t$ .

The objective function of the model training to produce artificial SAR images is defined as follows:

$$\mathcal{L} = \sum_{i=0}^N (h(I_t)_i - h(\hat{I}_t)_i)^2$$

where  $h(I)$  flattens the image  $I$  by concatenating the pixel rows of the image to 1-D array.  $N$  is the length of the array. The objective function is the mean squared error (mse) between the most recent image and the predicted image.

The intuitive description of  $\hat{I}_t$  is that the neural network-based mapping transformation function produces a prediction of how  $I_t$  should look like based on previous information about the location and the actual imaging conditions of  $I_t$ . The produced image  $\hat{I}_t$  can be used with the actual image  $I_t$  to create the DI  $\hat{I}_{DI}$  by using a simple algebraic operation like subtraction, ratio, or log ratio. Generating the DI is the standard method of conducting change detection, especially when using unsupervised methods [17].

Conventional methods of producing the DI often use only one of the previously captured images with the most recent image to generate the image, e.g.,  $I_{DI} = g(I_t, I_{t-y})$  [26]. This method has the previously discussed drawbacks of noise and imaging conditions affecting the final DI quality. By using the proposed mapping transformation function, the predicted image  $\hat{I}_t$  is used in the place of the previously captured image to generate the DI, e.g.,  $\hat{I}_{DI} = g(I_t, \hat{I}_t)$ . Recall that  $\hat{I}_t$  is a representation of  $I_t$  based on geospatial information from the time  $t-1$  and earlier, therefore it is missing all the changes that have happened after that time. The predicted image  $\hat{I}_t$  does

not contain noise and the mapping transformation function can correct the acquisition condition mismatch between the images, therefore the proposed method should produce better quality DIs when comparing it to the conventional method.

SAR imaging is sensitive to the soil moisture content of the imaged area [19]. A change in the soil moisture level changes the dielectric constant of the soil, and that way changes the SAR backscatter intensity. Often the soil moisture content changes should be ignored by the change detection system. Otherwise, the system would notify changes after every rainy day. This is one of the advantages of the proposed method. By adding weather to the model input acquisition condition parameters, the mapping transformation function can learn to construct  $\hat{I}_t$  in the actual weather conditions of  $I_t$  and should correctly model the changes in the soil moisture changing the backscatter intensity. Therefore, the false positive changes, that are potentially caused by weather condition changes, should be reduced.

In addition to weather, the acquisition condition parameters also include the imaging angle and identify the satellite that captured the image. A location is imaged by one of the sentinel satellites with an interval ranging from a few days to about a week. The satellite does not capture the image from the same angle every time. The satellite can be in ascending or descending orbit during the image acquisition and the incidence angle can vary between the overpasses. The ascending or descending orbit changes the look direction of the satellite, and that way has a considerable effect to the resulting image. The Sentinel-1 satellites are right-looking. When the satellite is descending from North to South it is imaging to the direction of West, and for ascending passes it is imaging to the direction of East [27]. Various 3-D features, like forest edges, lake banks, and hills are sensitive to the look direction, therefore the imaging angle is an important parameter when computing the DI. When using an image differencing method where only one previous image is used for DI computation, the imaging angle of the most recent image can restrict what previous images can be used to produce the DI. Seasonal changes, like foliage growth or change in snow cover, means that the most optimal image for the differencing would be the most recent previous image, however different imaging angles can limit the usage of the most recent images. This problem is not present with the proposed method. The model input includes  $n$  previous images and their imaging angle information. The model output image  $\hat{I}_t$  is produced using the actual acquisition conditions of  $I_t$ . The model can use all the information from all  $n$  input images, despite the input including images from different look directions, and the produced image  $\hat{I}_t$  represents an image that is acquired from the same angle as  $I_t$ .

## B. Neural Network Architecture

Fig. 3 illustrates the architecture of the neural network-based mapping transformation function. The architecture is based on the well-known U-Net neural network architecture [28]. The previous  $n$  SAR images, and the DEM are stacked to construct the input. The previous images and the DEM are all from the

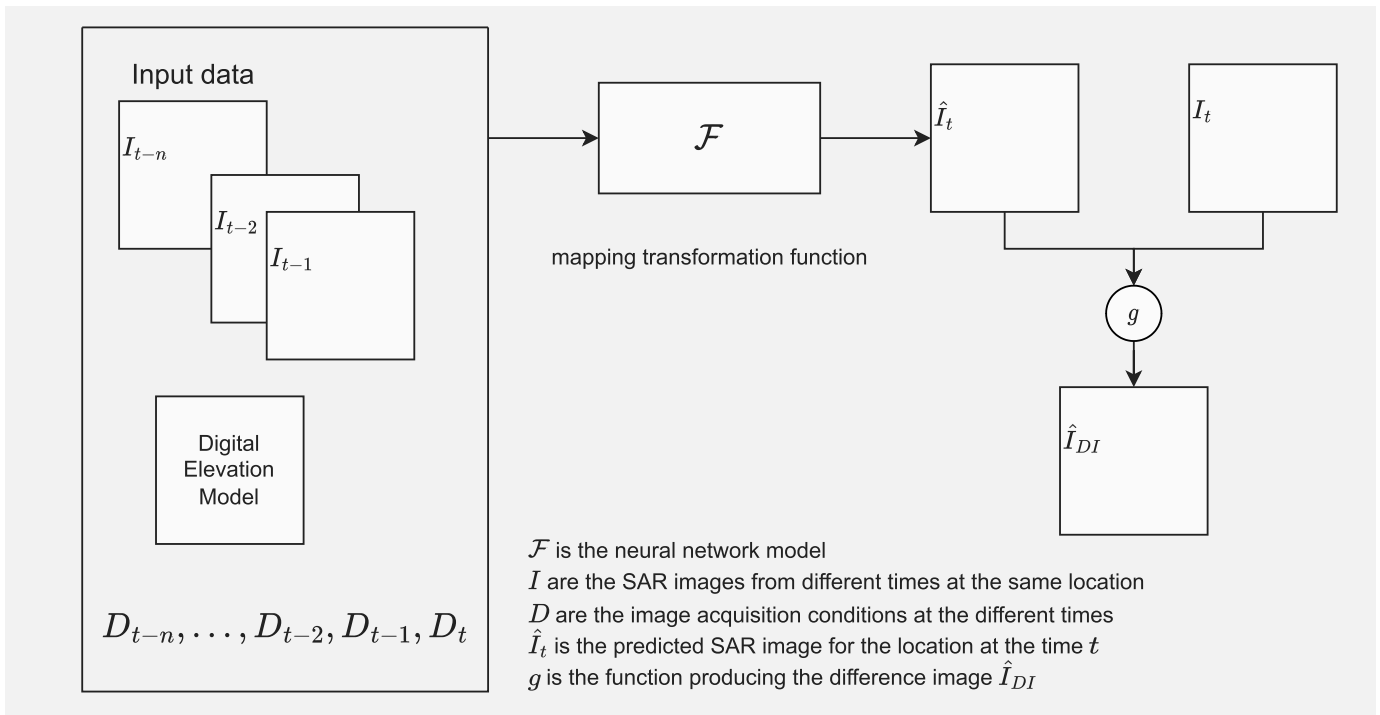


Fig. 2. Architectural overview of the proposed method. The neural network-based mapping transformation function fuses the information from previous image acquisitions and predicts what the scene should look like at the imaging conditions of  $I_t$ . The model output image  $\hat{I}_t$  and the actual image  $I_t$  is used to produce the DI  $\hat{I}_{DI}$ .

same location. The images are projected to the same resolution and the pixels across the different images are aligned to match the same geographical position. The U-Net architecture is constructed from encoder and decoder units. The encoder takes the input and compresses the input image stack to the latent space by using a set of downsampler blocks that half the input resolution using convolution layers with stride  $2 \times 2$ . The encoder stacks enough downsampler blocks so that the input image stack is compressed to  $1 \times 1$  resolution in image height and width dimensions. The image acquisition condition vector that contains the information of the acquisitions conditions for the  $n$  input images and the target image, is concatenated to the latent vector as described at the end of Section II-C. The resulting vector is then fed to the decoder that decodes the vector back to the dimensions of a normal SAR image outputting  $\hat{I}_t$ . The decoder is constructed from upsampler blocks that double the width and height dimensions using transposed convolution layers with stride  $2 \times 2$ . The decoder has same amount of upsampler blocks as the encoder has downsampler blocks. The number of filters, that are used in the upsampler and downsampler blocks, can be configured for every block individually, except for the final upsampler block that has the same number of filters as the SAR image has bands. The encoder and decoder layers are connected with skip connections that help the model in producing the output by not forcing the model to pack all the information into the latent vector. Instead, the information can flow from the input to the output by skipping most of the layers in the architecture. This is a standard method in U-Net style architectures.

### C. Dataset

A dataset is needed for the training of the neural network-based mapping transformation function. As discussed previously, the mapping transformation function input is composed of the previously taken SAR images; the acquisition conditions of the previous and the most recent SAR image; and the DEM from the location. The objective of the model is to learn to predict the most recent SAR image based on the input, therefore the most recent SAR image is the target in the training dataset. This means that the training dataset does not require any labeled data making the learning process of the proposed method unsupervised and economical to implement. The dataset can be generated directly from available data sources without the need of human labeling for the data. The dataset is available at the Fairdata.fi service [29].

The SAR images for the dataset were acquired from the ESA Copernicus Open Access Hub [30]. High-resolution interferometric wide swath (IW) ground range detected (GRD) products were used in this study [31]. The images were captured between March 2020 and August 2021 from the area illustrated in Fig. 4. All images from the time frame that included the area were downloaded from the Copernicus Open Access Hub. The images were preprocessed using the Sentinel-1 Toolbox from the Sentinel Application Platform (SNAP) [32], by applying the data preprocessing workflow described by Filipponi [33]. The optional noise filtering step was applied to the dataset using the Refined Lee filter from the SNAP toolkit. The more accurate AUX\_POEORB precise orbit files were used in the Apply Orbit File step.

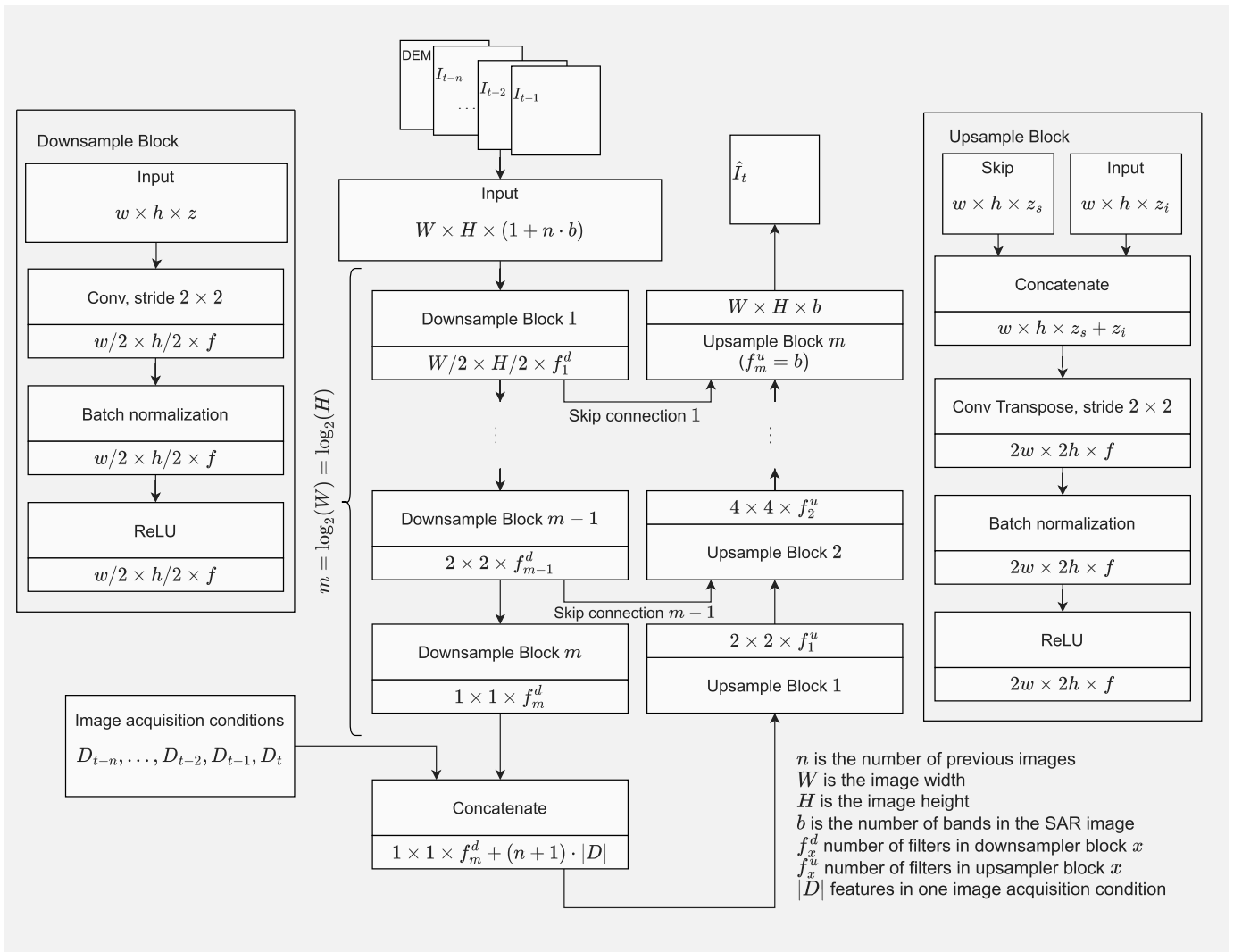


Fig. 3. Neural network architecture for the mapping transformation function. The architecture is based on the well-known U-Net neural network architecture. The image acquisition conditions are injected into the latent vector between the encoder and decoder.

The AUX\_POEORB files are available 20 days after the image acquisition [34], and since the processing was done in spring 2022, the more accurate orbit files were available for all images. The proposed workflow in [33] uses the SRTM Digital Elevation Database in the Range Doppler Terrain correction step, however, the database does not cover the area from where the dataset was created, therefore the Copernicus 30-m Global DEM was used that does cover the area. The SNAP toolkit can automatically download the required DEM files during preprocessing and the Terrain Correction step supports multiple different DEM sources, including the Copernicus 20-m Global DEM, thus the change was trivial to implement. The preprocessed images were saved as GeoTIFF files and uploaded to PostgreSQL<sup>2</sup> database that was using the PostGIS<sup>3</sup> extension. Using a relational database as the storage backend simplified the dataset generation process since all the data was available in one place and queryable with SQL.

Although the Copernicus 30-m Global DEM was used in the SAR image terrain correction preprocessing step, the product was not used for the mapping transformation input. Instead, we used more accurate DEM from NLS. NLS provides the DEM in multiple different resolutions of which the most accurate 2-m grid DEM was used [37]. The data is open access and distributed under Attribution 4.0 International (CC BY 4.0) license.<sup>4</sup> The DEM was downloaded in GeoTIFF format and uploaded to the same PostgreSQL database with the SAR images.

As discussed before, the image acquisition condition data included information about the weather when the images were captured. This data was acquired from the Finnish Meteorological Institute (FMI) that provides daily weather observations that are interpolated to  $1 \times 1$  km grid [38]. The interpolation method is described by Aalto et al. [39]. The data is distributed in NetCDF format and uploaded once a month. Daily mean temperature, daily precipitation sum, and snow depth data were downloaded from the time range. The daily observations were

<sup>2</sup><https://www.postgresql.org/>

<sup>3</sup><https://postgis.net/>

<sup>4</sup><https://creativecommons.org/licenses/by/4.0/>

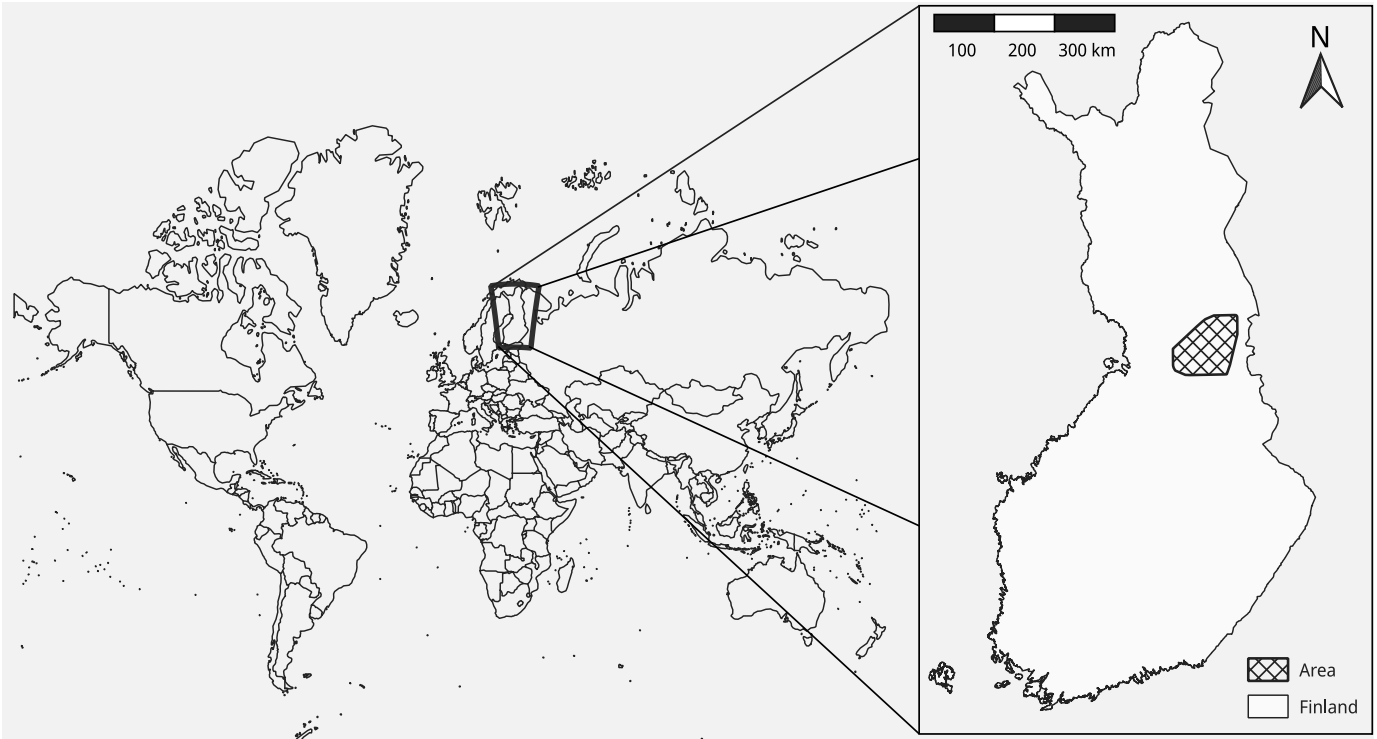


Fig. 4. Dataset was generated from images acquired from the marked area. The figure contains data from the National Land Survey of Finland (NLS) Topographic Database [35] and data from @EuroGeographics distributed by Eurostat [36].

extracted from the NetCDF files, converted to daily GeoTIFF rasters, and uploaded to the same PostgreSQL database with the SAR images and DEM.

The final data samples were created by sampling random locations from the area and random dates from the time range. For the training dataset, the time range was limited to the time before 20th of June in 2021, and for the test dataset the time was limited after the date. An assumption was made that the samples do not have any changes between the acquisitions  $I_{t-1}$  and  $I_t$ . This assumption is not likely true for all of the samples, however, the total dataset size is created to be large enough so that the samples that have changed between the two last acquisition should be marginally small fraction of the total dataset and neural networks can adapt to noisy data [40]. The image size was set to  $512 \times 512$  pixels, and number of previous images was set to 4. The spatial resolution of a high-resolution IW GRD product is  $20 \times 22$  m, and the images are distributed with  $10 \times 10$  m pixel spacing [41]. The geographical dimensions of the images were set to  $3 \times 3$  km making the pixel size  $3000/512 \approx 5.9$  m. This is higher resolution than the original 10-m pixel size, therefore the information loss is minimized during processing. For each random location and date, the target SAR image  $I_t$  was the next SAR image from the location that was available after the date. The input SAR images  $I_{t-4}, I_{t-3}, I_{t-2}, I_{t-1}$  were the SAR images from the four previous acquisitions from the location that were captured before  $I_t$ . The SAR images and the DEM were queried from the PostgreSQL database and the rasters were projected to the same projection window with the same  $512 \times 512$  resolution and  $3 \times 3$  km spatial dimensions using GDAL library [42]. The `gdal.Translate` function

was used for the projection with nearest neighbor resampling algorithm. After the projection, all pixels were geographically aligned across all images and the images could be stacked to construct the input image stack. The Sentinel-1 satellites use IW mode with dual polarization over the land areas thus one SAR image has two bands [12]. Both bands are used in all input images and the target image. That makes the input image stack have  $1 + 4 \cdot 2 = 9$  channels (DEM has one channel and every SAR image has two bands/channels), and the model output image has two bands.

The acquisition conditions were composed of the following features:

- 1) mean temperature of the acquisition date;
- 2) snow depth in the acquisition date;
- 3) satellite orbit direction during the acquisition (ascending/descending);
- 4) incidence angle;
- 5) satellite id (Sentinel-1A or Sentinel-1B);
- 6) precipitation amount in the acquisition date and three previous dates.

In addition to imaging conditions, such as weather and imaging angle, the satellite that captured the image is also added to the acquisition condition vector. The satellite id is encoded to 1 if the image is captured by the S1A satellite and 0 if the image is captured by the S1B satellite. The imaging instrumentation is not necessarily identical in both satellites and the model might learn to use this information to create more accurate images. All other features were scalar values from the acquisition date except for precipitation that is a vector with values for four different days. Since the moisture content of the soil has a known effect on the signal, and

moisture can linger long times in the soil, it was decided to include the precipitation amounts from multiple days to the acquisition conditions. Taking the precipitation amounts from the previous four days was a somewhat arbitrary decision with the reasoning that the neural network can learn to ignore the precipitation amounts from previous days if they have no use. The features were flattened to the final vector with a dimensionality of  $|D| = 9$ .

The final generated dataset had had around 230 000 training samples, and around 9000 test samples.

#### D. Experiment Setup

The performance of the proposed method was measured using experimentation. The main contribution of this article is to offer a new strategy for computing the DI. Existing methods generally use a strategy where the DI is computed using  $I_{DI} = g(I_{t-y}, I_t)$ , where  $g$  is the differencing function,  $I_{t-y}$  is one of the previous images from the location captured at some previous date, and  $I_t$  is the most recent image from the location. The proposed method uses the neural network output  $\hat{I}_t$  in place of  $I_{t-y}$  to compute the DI  $\hat{I}_{DI} = g(\hat{I}_t, I_t)$ . The mapping transformation function factors in the imaging conditions of  $I_t$  when generating  $\hat{I}_t$ , therefore  $\hat{I}_{DI}$  should be higher quality when compared to  $I_{DI}$ . The DI is generally further used in the change detection system to detect the changes by applying a classifier to the DI. The classifier outputs a change map indicating the pixels that contain the detected changes. By using identical classifier to classify the DIs generated by the two different methods and comparing the classifying accuracy of the resulting change maps, the quality of the two DIs can be measured.

1) *Change Simulation*: The experiment needs a dataset with known changes so that the accuracy of the change detection classifier can be determined. This is a challenge since only a small number of datasets exists for remote sensing change detection even for optical satellite images [43]. For SAR images there are only a few datasets such as the ones used in the following publications [44], [45], however, they consist of only a few SAR image pairs with a hand labeled change map. Currently, there are no large enough SAR datasets for deep learning applications available online [46].

To avoid the problem of the lack of change detection datasets for SAR images, the decision was made to use simulation to add changes to real SAR images. This technique was used by Inglada and Mercier [47] where they measured the performance of their statistical similarity measure change detection algorithm using simulated changes. The authors used three different methods for change simulation. The techniques were: *offset change*, where the original value was shifted by a value; *Gaussian change*, where the original value was changed by adding zero mean Gaussian noise to the value; and *deterministic change*, where a value was copied from some other location in the image. Likewise, Cui et al. [48] used change simulation for SAR images when they introduced an evaluation benchmark for SAR change detection algorithms. The change simulation methods in this article try to replicate changes that are commonly seen in the real world using

techniques that correctly resemble the statistical properties of the real-world changes. Based on these papers two change simulation methods were devised for this study.

- 1) *Offset Change*: A value is added to the original pixel value. The simulation does not try to replicate any real-world change, however, it is trivial to implement, and the offset value can be changed to test different offsets.
- 2) *First-Order Statistical Change*: The statistical distribution of the change area is converted to the statistical distribution of some other nearby geographical feature. This replicates the real-world changes more accurately.

Fig. 5 illustrates the simulated change methods when applied to an example SAR image. The changes were added to the SAR images by creating a random shape mask and positioning the mask to a random location in the SAR image. The pixel values inside the mask were changed using the selected method. The location of the mask was restricted to forested geographical areas in the SAR image. If the mask location was at the forest edge, the mask part that landed outside of the forested area was not changed. The information about different geographical features was acquired from the NLS Topographic Database [49]. The database was also utilized in first-order statistical change implementation where the forest area pixel values were changed to follow the statistical distribution of some other geographical feature. The nearest areas of the desired geographical feature type were queried from the database, and the statistical distribution of the pixel values was estimated using a univariate kernel density estimator (KDE) from the statsmodels Python library [50]. A second univariate KDE model was fit to the pixel values of all forested area pixels in the SAR image. The mapping of the pixel values was implemented using the method of modifying first-order statistical distribution described in [48]. The change area pixel values were first mapped to uniform distribution in the interval  $[0, 1]$  by using the cumulative distribution function (cdf) of the forest area KDE. After that, the inverse cdf of the second KDE model is applied to the uniformly distributed values, thus mapping them to the distribution of the desired geographical feature.

The simulated change dataset is only needed for validating that the DIs that are generated using the proposed method are of higher quality when compared to the DIs that are generated using the conventional method. The simulated changes are not used for training the neural network. Therefore, the simulated changes are added only to the neural network test dataset samples. A random number of changes, ranging from 0 to 3, were added to each of the samples.

2) *Difference Image Generation and Change Classifiers*: The quality of the DIs was measured using two different classifiers. The first method is a simple threshold method. A thresholding value is chosen, and the pixels are classified to changed or unchanged based on if the value is smaller or greater than the threshold. This requires that the pixels have scalar values. The scalar valued DIs were produced using the following equations:

$$\hat{I}_{DI}(x, y) = \sqrt{\sum_b (I_t(x, y, b) - \hat{I}_t(x, y, b))^2} \quad (1)$$

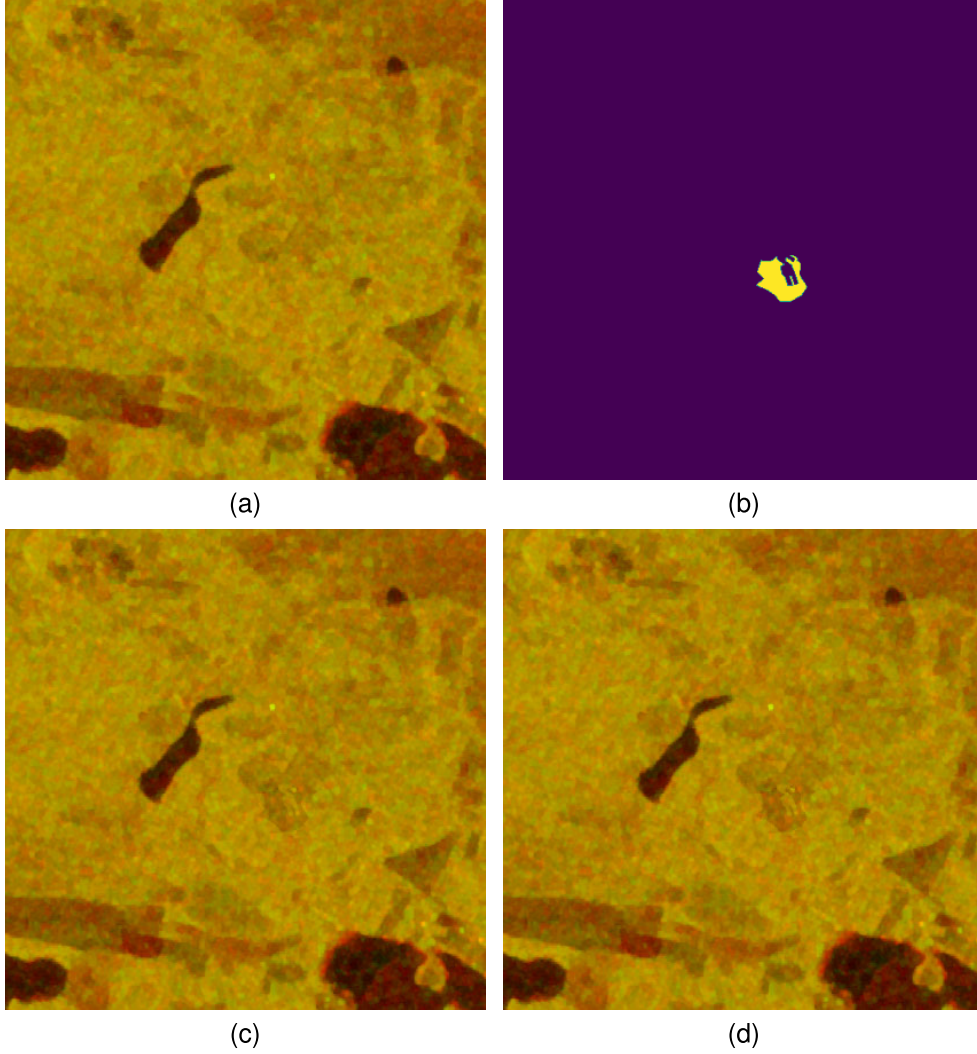


Fig. 5. Example of the two simulated change methods. (a) Original image where the SAR images are visualized as an RGB image by using red and green channels for the two bands and where the blue channel is set to zero. (b) Change mask where the simulated change is applied. (c) Offset change image where  $-2.5$  dB shift is applied to the change location. (d) First-order statistical change where the resulting change mean is close to the change in the offset change image.

$$I_{DI}(x, y) = \sqrt{\sum_b (I_t(x, y, b) - I_{t-y}(x, y, b))^2}. \quad (2)$$

In the equations,  $\hat{I}_{DI}$  is the DI that is computed using the proposed method,  $I_{DI}$  is the DI that is computed using the conventional method,  $b$  is the band, and the  $x$  and  $y$  define the pixel location. The different bands are considered as vector dimensions. Pythagorean theorem is used to compute the vector length that is used as the value for the DI pixel. The threshold method was used as an example of an unsupervised classifier algorithm [46]. The performance of the threshold classifiers was measured using the well known area under curve (AUC) metric that is computed from the receiver operating characteristic (ROC) curve. The metrics were computed to the test partition of the neural network mapping function dataset. The  $\hat{I}_{DI}$  and  $I_{DI}$  DIs were computed for every sample in the test dataset, and the pixels from all samples were used to generate the two datasets that were used to compute the ROC curves and AUC metrics.

The second classifier was the linear support vector classifier (SVC). The SVC was used as an example of supervised machine learning algorithm. The support vector models work with multidimensional data, therefore the DIs were produced using simple subtraction

$$\hat{I}_{DI}(x, y, b) = I_t(x, y, b) - \hat{I}_t(x, y, b) \quad (3)$$

$$I_{DI}(x, y, b) = I_t(x, y, b) - I_{t-y}(x, y, b). \quad (4)$$

The test dataset from the mapping transformation function training was used to train the classifiers. For each sample, the two DIs were computed, and the pixels from all DI samples were used to create the two datasets. The first dataset was generated using the pixels from the  $\hat{I}_{DI}$  samples, and the second dataset was generated using the pixels from the  $I_{DI}$  samples. The two datasets were further divided to train and test datasets with a rule that all pixels originating from one image sample end up on the same side of the split. The train test split was also identical for both datasets. The datasets were



TABLE I  
PARAMETERS FOR THE NEURAL NETWORK ARCHITECTURE

Block number	Downsampler (filter size, kernel size)	Upsampler (filter size, kernel size)
1	64, 4	512, 4
2	128, 4	512, 4
3	256, 4	512, 4
4	512, 4	512, 4
5	512, 4	512, 4
6	512, 4	512, 4
7	512, 4	256, 4
8	512, 4	128, 4
9	512, 2	2, 4

used to train two instances of the classifier and measure their accuracy.

### III. RESULTS

#### A. Training the Neural Network-Based Mapping Transformation Function

Different neural network parameters were experimented with, and the best results were achieved with the parameters shown in Table I. The mse was used as the loss function, and AdamW [51] was used as the optimizer. The neural network architecture was implemented using TensorFlow deep learning framework [52]. The training was conducted on one NVIDIA V100 GPU with a batch size of 200, and a training time of around 30 h.

Fig. 6(c) demonstrates the model performance for one of the test samples. Fig. 6(a) shows the real SAR image that the model tries to predict. Fig. 6(b) illustrates the difference between the real SAR image and the model output with a heat map where lighter color indicates a greater error. The predicted image is very close to the real SAR image except for lack of noise that is purely random and impossible for the model to predict. Likewise, the bottom-right corner of the image has an area that has greater error in the prediction. The error is located in a lake, therefore the error can be a result of waves that are likewise impossible to predict.

The proposed method depends on that the mapping transformation function adapts the predicted image  $\hat{I}_t$  based on the imaging conditions of  $I_t$ . To verify that the model genuinely uses the image acquisition conditions to produce  $\hat{I}_t$ , the model was experimented to produce outputs with manually modified imaging condition vector  $D_t$ . Fig. 6(d) and (f) image pair illustrates model outputs where  $D_t$  is modified to have opposite orbit directions. Fig. 6(e) illustrates the difference between the images. The lake banks and the top-left corner of the image, where there is a small hill, have large differences between the two generated images. All locations, where there are greater differences between the images, are 3-D features. The Sentinel-1 satellites have different look directions on ascending and descending orbit directions. Therefore, the scattering of the radar signal is different and the difference is most noticeable on 3-D features. Since the differences are so clearly located on the 3-D features in the image the model is clearly factored in the orbit direction when generating the output. This verifies that the imaging conditions are used by the model to produce  $\hat{I}_t$  in the imaging conditions of  $I_t$ .

TABLE II  
CONVENTIONAL DI STRATEGY RESULTS

$I_{t-y}$ Selection Method	AUC
Method 1	0.79
Method 2	0.75
Method 3	0.71

The same experiment was conducted by modifying the precipitation amounts in Fig. 6(g) and (i). The difference between the generated images is shown in Fig. 6(h). This time the difference between the generated images is focused on swamp, meadow, and agricultural land areas in the image. The forest areas have only small differences between the images. In forest areas, the radar signal is scattered back by the forest foliage where the moisture does not affect the scattering properties as much as the open areas. In open areas, the radar signal hits the ground where the soil moisture content is altered more by the rain, thus changing the backscatter intensity. This experiment suggests that the model uses the precipitation information correctly when generating the output image.

#### B. Identifying the Best Conventional DI Strategy

The conventional method of computing the DI is to use one of the previous SAR images that is captured at some preceding date with the most recent image to produce the DI  $I_{DI} = g(I_{t-y}, I_t)$ . There are multiple different strategies when selecting the previous image. The simplest strategy is to select the previous image that is preceding the image that was captured most recently. This strategy has the advantage that the least amount of time has elapsed between the images, therefore the number of natural changes, like foliage growth or soil moisture changes, are minimized. However, the problem is that the previous image has very likely different incidence angle and it might have been captured from a different orbit direction (ascending/descending). To make sure that we compare the proposed method to the best conventional method, three different previous image selection strategies were compared to identify the best strategy. The threshold classifier was used to compare the quality of the DIs that were produced using the different strategies. The strategies have different trade offs between the elapsed time and imaging angle.

*Method 1:* Closest incidence angle and the same orbit direction.

*Method 2:* Most recent previous image with the same orbit direction.

*Method 3:* Most recent previous image preceding the target image ( $I_{t-1}$ ).

Fig. 7 illustrates the comparison of the three different methods using ROC curve plots. Table II shows the results in a list format by displaying the AUC metrics. The strategy where the previous image is captured from the same orbit direction and has the closest incidence angle with  $I_t$  is the best  $I_{t-y}$  selection strategy. From this on forward, the Method 1 is always used when referring to the conventional method of computing DI.

#### C. Proposed Method Versus Conventional Method

1) *Threshold Classifier:* Fig. 8 illustrates the ROC curve plots for the two threshold classifiers when measuring the

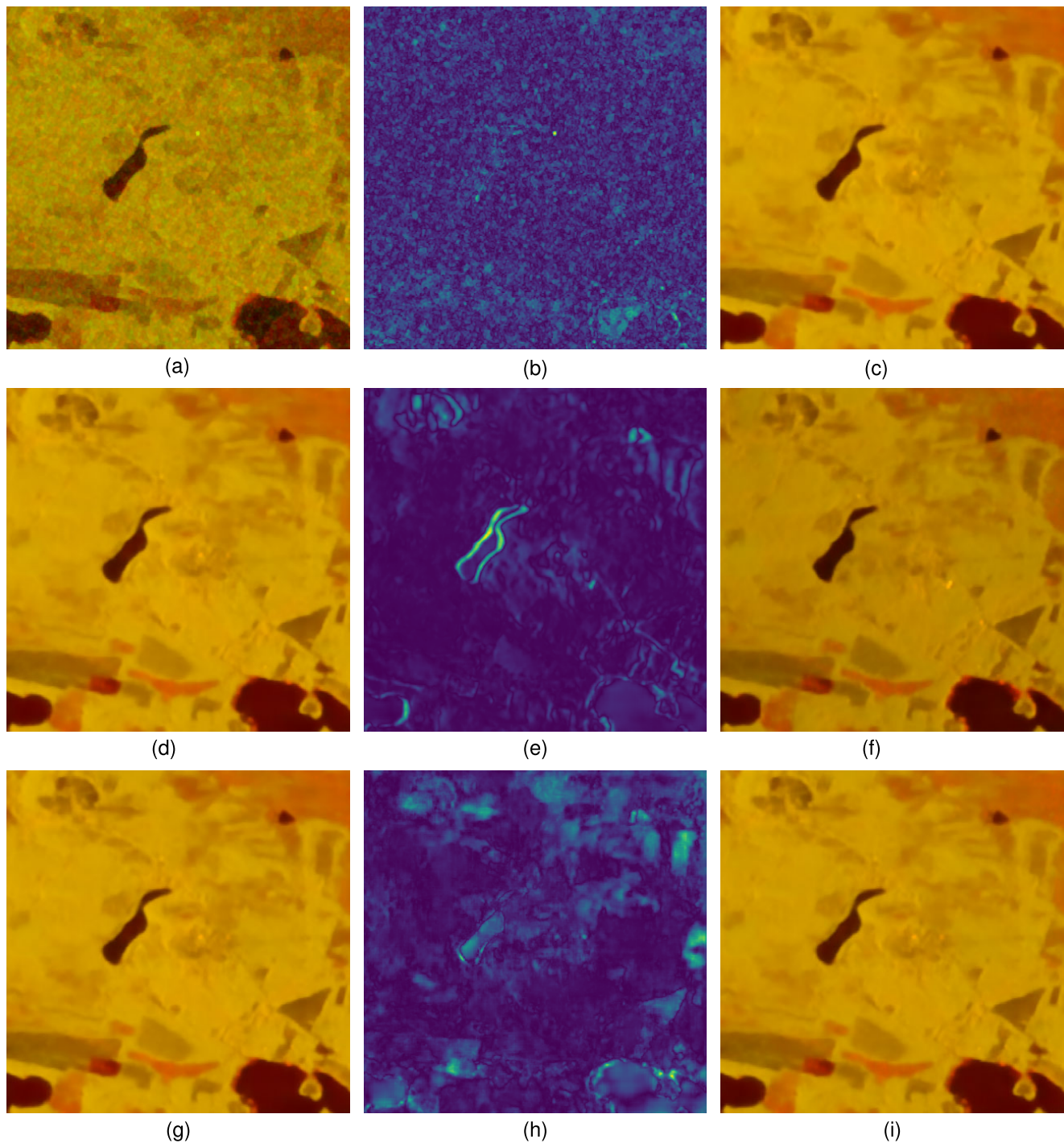


Fig. 6. Mapping transformation function outputs with different imaging conditions. (a) Original SAR image that is captured from coordinates 64.919 lat, 28.124 lon on July 7, 2021 (target image  $I_t$ ). (b) Difference between the true image  $I_t$  and the predicted image  $\hat{I}_t$ . (c) Model output  $\hat{I}_t$  when it is trying to predict  $I_t$ . (d), (f), (g), and (i) Images are generated by manually modifying the imaging condition vector  $D_t$ . (d) and (f) Ascending and descending orbit direction, respectively. (e) Difference between the different orbit direction images. (g) and (i) Identical experiment was conducted by varying the precipitation amount, i.e., zero rain and Substantial amount of rain, respectively. (h) Difference between the images with the different precipitation amounts.

quality of the DIs generated with the two methods. In this experiment, the changes are simulated to the dataset using the offset change method. The AUC metrics from the experiment is also shown in a list format in the first row of Table III. The simulated shift is  $-2.5$  dB in the change area, which represents a considerable change. In the real world, this could be a change where the forest is clear cut, making it smoother, and that way

reducing the backscatter intensity. The threshold classifier that uses the DIs that are produced using the proposed method is clearly better. This indicates that the proposed method generates better quality DIs.

Fig. 9 illustrates the results of the same experiment when it is repeated to the simulated change dataset using the statistical change method. The change areas are simulated to emit the

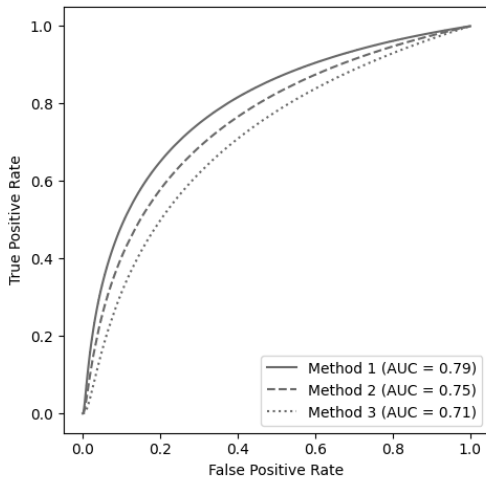


Fig. 7. Comparison of different previous image selection strategies when using the traditional method of computing the DI.

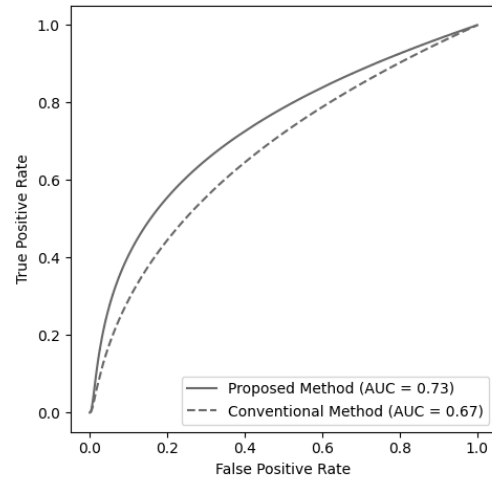


Fig. 9. ROC curve for the two threshold classifiers when applied to the dataset with simulated statistical changes.

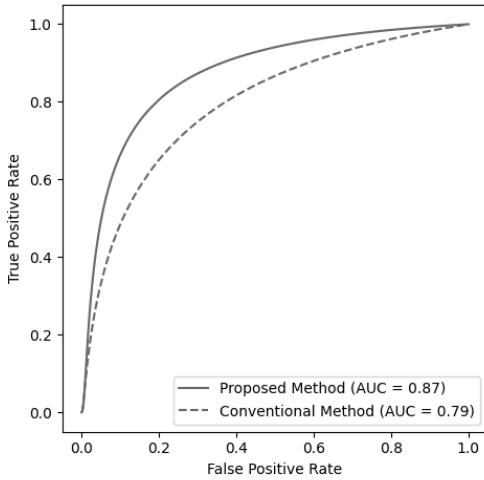


Fig. 8. ROC curve for the two threshold classifiers when applied to the dataset with simulated changes using the offset change method.

TABLE III  
EXPERIMENT RESULTS FOR THE THRESHOLD MODELS

Dataset	Proposed method AUC	Conventional method AUC
Shift change	0.87	0.79
Statistical change	0.73	0.67

backscatter intensity of nearby forest areas that are not as densely wooded making this a more realistic representation of real changes in the forest. The mean backscatter intensity change varied from around  $-0.5$  to  $-2.5$  dB in the change areas depending on the sample. The AUC metrics from the experiment is also shown in a list format in the second row of Table III. Both classifiers have considerably worse performance, however, the proposed method is still better performing. The overall poor performance is to be expected with the threshold classifiers. It is the simplest possible classifier working in a single pixel level without having any kind of visibility to the neighboring pixels. Furthermore, the changes can be small in the simulated change dataset that is created using the statistical change method.

TABLE IV  
EXPERIMENT RESULTS FOR THE SVC MODELS

Dataset	Proposed method accuracy	Conventional method accuracy
Shift change	0.89	0.81
Statistical change	0.75	0.70

2) *Support Vector Classifier*: The experiments were repeated with the SVC model to the same two datasets. The linear kernel SVC implementation `LinearSVC` from Scikit-learn library [53] was used to conduct the experiment. Linear kernel SVC was chosen due to the large dataset size. Other kernel types were tested, however, they did not scale to a large number of samples. The samples were normalized using the Scikit-learn `StandardScaler` to ease the model convergence. Table IV displays the results from the experiments. The proposed method is clearly superior to the conventional method in both experiments. The performance in the statistical change dataset is considerably worse when compared to the shift change dataset. However, this is to be expected with a similar loss of accuracy in the threshold classifier experiments. This experiment uses supervised learning with a labeled dataset which should improve the results when compared to the threshold classifier. However the SVC is still a very simple classifier that performs the classification at pixel level without any visibility to the neighboring pixels, thus the accuracy scores are mediocre at best. Still, achieving a high accuracy score was not the goal of the experiment. Instead, the experiment compares the accuracies of the two classifiers and the results from this experiment support the findings from the threshold classifier experiments. The proposed method clearly produces higher quality DIs.

3) *Model Without the Weather Data*: The dataset creation for this project was a major undertaking that complicates the adaption of the proposed methodology since the model needs to be trained to every location where it is used. FMI provides the interpolated weather data for the features we used in this study that are available in locations inside the borders of Finland. However, equivalent data sources are not necessary

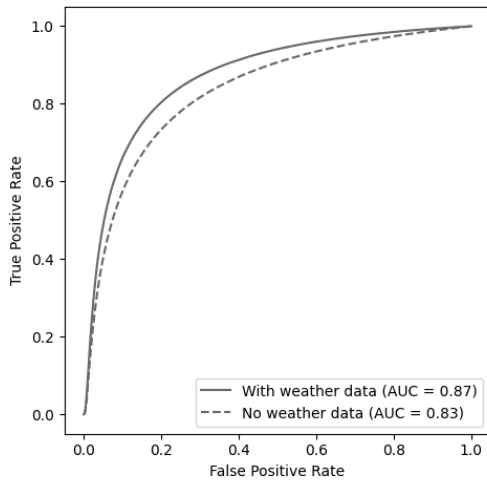


Fig. 10. Threshold classifier ROC when used with mapping transformation function that is trained without the weather data. The solid line is the same line from Fig. 8 that is added to the plot to help visual comparison of the results.

TABLE V  
ABLATION STUDY IMPACT TO SVC MODEL ACCURACIES

Dropped feature	Shift change dataset	Statistical change dataset
Mean temperature	0.88	0.75
Snow depth	0.89	0.75
Satellite orbit direction	0.87	0.74
Incidence angle	0.88	0.75
Satellite id	0.89	0.76
Precipitation	0.87	0.74

available in other countries. Therefore, we experimented with how the neural network-based mapping transformation function works without the weather data. The model training pipeline was modified to drop the weather data during training and inference, thus the acquisition conditions consisted only from incidence angle, satellite orbit direction, and satellite id. Fig. 10 illustrates the results from the experiment. The experiment used simulated changes with  $-2.5$ -dB shift and the exact same model hyperparameters with the results that are illustrated in Fig. 8, thus the result is directly comparable. The resulting AUC metric is higher at 0.83 when compared to the conventional method at 0.79, however, the result is worse when compared to the model that has visibility to the weather data with AUC metric of 0.87. We can conclude that the proposed methodology can be used also without weather data, and it achieves measurable improvement over the conventional method. However, to achieve the best performance, the model requires the weather data in addition to the other imaging condition features.

4) *Feature Ablation Study*: An ablation study measured the impact of each feature to the accuracy of the SVC using both the shift change dataset and the statistical change dataset. Table V shows the accuracies of the classifier when the neural network  $\mathcal{F}$  is trained with a dataset where one of the features is removed. Satellite orbit direction and precipitation are two of the most important features, because dropping them decreased accuracy. Satellite id and snow depth were the least important.

## IV. DISCUSSION

The experiment results show that the proposed method produces higher quality DIs than the conventional method. Since the output from the proposed method is a DI, many of the existing change classification techniques may benefit from the method without any modifications. The techniques generally use the conventional method for producing the DI, however, it is a completely separate step from classification and thus could be replaced with the proposed method without changes to the classification step. Some methods do not use the DI computation step, instead, they accept the two images directly to the model to carry out the classification. Even with these techniques, the usage of the proposed method could be beneficial. In these cases, the earlier image ( $I_{t-y}$ ) is replaced with  $\hat{I}_t$ , thus giving the classification model better understanding of what the scene should look like in the correct image acquisition conditions.

This study did not experiment with the more advanced change detection classifiers since the simple classifiers were enough to prove that the proposed method is better than the conventional method. However, the clear improvement in classification accuracy with the simple methods could indicate that similar improvement can be achieved with the more advanced methods.

The use of simulated changes to measure the performance of the method was a necessary compromise caused by the lack of existing change detection datasets suitable for training the neural network. The simulated changes are not realistic enough to draw conclusions about how much the proposed method would improve the change detection performance in real world. However, the experiments with the simulated changes indicate a substantial performance improvement potential.

The downside of the proposed method is that the mapping transformation function is a neural network model that requires a training dataset and a considerable amount of processing power for training. The dataset creation is a complex operation that combines data from multiple data sources. Some of the sources that were used in this study are available only for geographical locations inside Finland, such as the interpolated weather data from FMI. The model requires training data from the locations it is used at inference time which complicates the adaption of the method outside of Finland. However, many of the data sources very likely have equivalents available in other geographical locations, therefore the adoption is not impossible. Even a global training dataset could potentially be constructed, which could make the training of a universal model possible. The recent advances in neural network architectures with natural language processing and image generation have shown that the models can learn from impressive amounts of data. The model training is unsupervised, meaning it does not require labeled data, thus the creation of such a dataset could be possible. Our experiment with a model that did not see the weather data in the input shows that the method achieves measurable improvement over the conventional method even when the model has information only about the imaging angle and the satellite. That data is available in the SAR images when they are downloaded from the ESA open access portal, thus simplifying the dataset creation

considerably. However, without the weather data the mapping transformation function cannot generate accurate enough SAR images to achieve the same accuracy metrics as the model with the weather information. The ablation study suggests that model training could be simplified since some of the features are found to be less important, and therefore can be dropped from the training data.

#### DATA AVAILABILITY

The Sentinel-1 SAR imagery is available to download free of charge from Copernicus Open Access Hub [30]. The weather data is available free of charge from FMI [38]. The DEM and topographic database are available free of charge from the NLS open data service [37], [49]. Links to the download sites are listed in the references. The derived dataset can be downloaded from the Fairdata.fi service [29]. The computer code to produce the results is available at <https://github.com/janne-alatalo/sar-change-detection>.

#### ACKNOWLEDGMENT

This research was conducted at the Institute of Information Technology of JAMK University of Applied Sciences as part of *Data for Utilisation – Leveraging digitalisation through modern artificial intelligence solutions and cybersecurity project*, *coADDVA - ADDing Value by Computing in Manufacturing project*, and *Finnish Future Farm project*. This work uses modified Copernicus Sentinel data 2020-2021. The authors thank Eppu Heilimo for feedback on the draft manuscript. They wish to acknowledge CSC - IT Center for Science, Finland, for hosting the dataset.

#### REFERENCES

- [1] J. Sublime and E. Kalinicheva, "Automatic post-disaster damage mapping using deep-learning techniques for change detection: Case study of the Tohoku tsunami," *Remote Sens.*, vol. 11, no. 9, p. 1123, May 2019. [Online]. Available: <https://www.mdpi.com/2072-4292/11/9/1123>
- [2] L. Dong and J. Shan, "A comprehensive review of earthquake-induced building damage detection with remote sensing techniques," *ISPRS J. Photogramm. Remote Sens.*, vol. 84, pp. 85–99, Oct. 2013. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0924271613001627>
- [3] S. Long, T. E. Fatoyinbo, and F. Policelli, "Flood extent mapping for Namibia using change detection and thresholding with SAR," *Environ. Res. Lett.*, vol. 9, no. 3, Mar. 2014, Art. no. 035002.
- [4] E. Tomppo, G. Ronoud, O. Antropov, H. Hytönen, and J. Praks, "Detection of forest windstorm damages with multitemporal SAR data—A case study: Finland," *Remote Sens.*, vol. 13, no. 3, p. 383, Jan. 2021. [Online]. Available: <https://www.mdpi.com/2072-4292/13/3/383>
- [5] M. Rüetschi, D. Small, and L. Waser, "Rapid detection of windthrows using Sentinel-1 C-band SAR data," *Remote Sens.*, vol. 11, no. 2, p. 115, Jan. 2019. [Online]. Available: <https://www.mdpi.com/2072-4292/11/2/115>
- [6] P. de Bem, O. de Carvalho Junior, R. F. Guimarães, and R. T. Gomes, "Change detection of deforestation in the Brazilian Amazon using Landsat data and convolutional neural networks," *Remote Sens.*, vol. 12, no. 6, p. 901, Mar. 2020. [Online]. Available: <https://www.mdpi.com/2072-4292/12/6/901>
- [7] V. Sood, H. S. Gusain, S. Gupta, A. K. Taloor, and S. Singh, "Detection of snow/ice cover changes using subpixel-based change detection approach over Chhota-Shigri glacier, Western Himalaya, India," *Quaternary Int.*, vols. 575–576, pp. 204–212, Feb. 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1040618220302494>
- [8] D. Lu, P. Mausel, E. Brondizio, and E. Moran, "Change detection techniques," *Int. J. Remote Sens.*, vol. 25, no. 12, pp. 2365–2401, Jun. 2004.
- [9] W. Fu, J. Ma, P. Chen, and F. Chen, "Remote sensing satellites for Digital Earth," in *Manual of Digital Earth*. Singapore: Springer, 2020, pp. 55–123.
- [10] A. G. Castriotta. (2022). *Copernicus Sentinel Data Access Annual Report Annual Report*. [Online]. Available: [https://scihub.copernicus.eu/twiki/pub/SciHubWebPortal/AnnualReport2021/COPE-SERCO-RP-22-1312\\_-\\_Sentinel\\_Data\\_Access\\_Annual\\_Report\\_Y2021\\_merged\\_v1.1.pdf](https://scihub.copernicus.eu/twiki/pub/SciHubWebPortal/AnnualReport2021/COPE-SERCO-RP-22-1312_-_Sentinel_Data_Access_Annual_Report_Y2021_merged_v1.1.pdf)
- [11] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou, "A tutorial on synthetic aperture radar," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 1, pp. 6–43, Mar. 2013.
- [12] European Space Agency. *Sentinel-1 SAR Technical Guide—Interferometric Wide Swath*. Accessed: Feb. 7, 2023. [Online]. Available: <https://sentinels.copernicus.eu/web/sentinel/user-guides/sentinel-1-sar/acquisition-modes/interferometric-wide-swath>
- [13] M. Gong, Y. Li, L. Jiao, M. Jia, and L. Su, "SAR change detection based on intensity and texture changes," *ISPRS J. Photogramm. Remote Sens.*, vol. 93, pp. 123–135, Jul. 2014. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0924271614001051>
- [14] H. Zhu, Y. Cao, Z. Zhou, M. Gong, and L. Jiao, "Parallel unsupervised synthetic aperture radar image change detection on a graphics processing unit," *Int. J. High Perform. Comput. Appl.*, vol. 27, no. 2, pp. 109–122, May 2013.
- [15] W. Zhang, L. Jiao, F. Liu, S. Yang, W. Song, and J. Liu, "Sparse feature clustering network for unsupervised SAR image change detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5226713.
- [16] Y. Li, C. Peng, Y. Chen, L. Jiao, L. Zhou, and R. Shang, "A deep learning method for change detection in synthetic aperture radar images," *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 8, pp. 5751–5763, Aug. 2019.
- [17] L. Bruzzone and D. F. Prieto, "An adaptive semiparametric and context-based approach to unsupervised change detection in multitemporal remote-sensing images," *IEEE Trans. Image Process.*, vol. 11, no. 4, pp. 452–466, Apr. 2002.
- [18] Y. Gauthier, M. Bernier, and J.-P. Fortin, "Aspect and incidence angle sensitivity in ERS-1 SAR data," *Int. J. Remote Sens.*, vol. 19, no. 10, pp. 2001–2006, Jan. 1998.
- [19] H. S. Srivastava, P. Patel, Y. Sharma, and R. R. Navalgund, "Large-area soil moisture estimation using multi-incidence-angle RADARSAT-1 SAR data," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 8, pp. 2528–2535, Aug. 2009.
- [20] M. Reichstein et al., "Deep learning and process understanding for data-driven Earth system science," *Nature*, vol. 566, no. 7743, pp. 195–204, Feb. 2019.
- [21] M. Datcu, Z. Huang, A. Anghel, J. Zhao, and R. Căcoveanu, "Explainable, physics-aware, trustworthy artificial intelligence: A paradigm shift for synthetic aperture radar," *IEEE Geosci. Remote Sens. Mag.*, vol. 11, no. 1, pp. 8–25, Mar. 2023.
- [22] R. Touzi, "A review of speckle filtering in the context of estimation theory," *IEEE Trans. Geosci. Remote Sens.*, vol. 40, no. 11, pp. 2392–2404, Nov. 2002.
- [23] F. Argenti, A. Lapini, T. Bianchi, and L. Alparone, "A tutorial on speckle reduction in synthetic aperture radar images," *IEEE Geosci. Remote Sens. Mag.*, vol. 1, no. 3, pp. 6–35, Sep. 2013.
- [24] X. Chen, G. Li, Z. Chen, Q. Ju, and X. Cheng, "Incidence angle normalization of dual-polarized Sentinel-1 backscatter data on Greenland Ice Sheet," *Remote Sens.*, vol. 14, no. 21, p. 5534, 2022. [Online]. Available: <https://www.mdpi.com/2072-4292/14/21/5534>
- [25] G. Kaplan, L. Fine, V. Lukyanov, V. S. Manivasagam, J. Tanny, and O. Rozenstein, "Normalizing the local incidence angle in Sentinel-1 imagery to improve leaf area index, vegetation height, and crop coefficient estimations," *Land*, vol. 10, no. 7, p. 680, Jun. 2021. [Online]. Available: <https://www.mdpi.com/2073-445X/10/7/680>
- [26] H. Zhuang, Z. Tan, K. Deng, and H. Fan, "It is a misunderstanding that log ratio outperforms ratio in change detection of SAR images," *Eur. J. Remote Sens.*, vol. 52, no. 1, pp. 484–492, Jan. 2019.
- [27] O. Mora, P. Ordoqui, R. Iglesias, and P. Blanco, "Earthquake rapid mapping using ascending and descending Sentinel-1 TOPSAR interferograms," *Proc. Comput. Sci.*, vol. 100, pp. 1135–1140, Jan. 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050916324358>
- [28] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention—MICCAI*, N. Navab, J. Hornegger, W. M. Wells, and A. F. Frangi, Eds. Cham, Switzerland: Springer, 2015, pp. 234–241.

- [29] *Simulated Change Detection Dataset for Neural Networks*, JAMK Univ. Appl. Sci., Jyväskylä, Finland, School Technol., 2023, doi: 10.23729/7b22c271-5e25-40fe-aa6a-f5d0330b1872.
- [30] European Space Agency. *Copernicus Sentinel Data 2020–2021*. Accessed: Jan. 25, 2023. [Online]. Available: <https://scihub.copernicus.eu/>
- [31] *Sentinel-1 SAR Technical Guide—Level-1 GRD Products*. Accessed: Jan. 26, 2023. [Online]. Available: <https://sentinels.copernicus.eu/web/sentinel/technical-guides/sentinel-1-sar/products-algorithms/level-1-algorithms/ground-range-detected>
- [32] *SNAP—ESA Sentinel Application Platform v8.0.0*. Accessed: Jan. 25, 2023. [Online]. Available: <https://step.esa.int/main/toolboxes/snap/>
- [33] F. Filippini, “Sentinel-1 GRD preprocessing workflow,” *Proceedings*, vol. 18, no. 1, p. 11, Jun. 2019. [Online]. Available: <https://www.mdpi.com/2504-3900/18/1/11>
- [34] *Copernicus POD Service File Format Specification*, document Version 2.0, GMV, 2022. [Online]. Available: [https://sentinel.esa.int/documents/247904/351187/Copernicus\\_Sentinels\\_POD\\_Service\\_File\\_Format\\_Specification](https://sentinel.esa.int/documents/247904/351187/Copernicus_Sentinels_POD_Service_File_Format_Specification)
- [35] National Land Survey of Finland. *Division Into Administrative Areas (Vector)*. Accessed: Jan. 31, 2023. [Online]. Available: <https://www.maannittauslaitos.fi/en/maps-and-spatial-data/professionals/product-descriptions/division-administrative-areas-vector>
- [36] European Commission, Eurostat (ESTAT), GISCO. *Countries 2020—Administrative Units—Dataset*. Accessed: Jan. 31, 2023. [Online]. Available: <https://ec.europa.eu/eurostat/web/gisco/geodata/reference-data/administrative-units-statistical-units/countries>
- [37] National Land Survey of Finland. *Elevation Model 2 m*. Accessed: Jan. 24, 2023. [Online]. Available: <https://www.maannittauslaitos.fi/en/maps-and-spatial-data/expert-users/product-descriptions/elevation-model-2-m>
- [38] Finnish Meteorological Institute. *Daily Observations in 1km\*1km Grid*. Accessed: Jan. 24, 2023. [Online]. Available: <https://en.ilmatiiteenlaitos.fi/gridded-observations-on-aws-s3>
- [39] J. Aalto, P. Pirinen, and K. Jylhä, “New gridded daily climatology of Finland: Permutation-based uncertainty estimates and temporal trends in climate,” *J. Geophys. Res., Atmos.*, vol. 121, no. 8, pp. 3807–3823, Apr. 2016. [Online]. Available: <https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2015JD024651>
- [40] H. Song, M. Kim, D. Park, Y. Shin, and J.-G. Lee, “Learning from noisy labels with deep neural networks: A survey,” *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Mar. 7, 2022, doi: 10.1109/TNNLS.2022.3152527.
- [41] *Sentinel-1 Product Definition*, document S1-RS-MDA-52-7440, European Space Agency, Paris, France, 2016. [Online]. Available: <https://sentinel.esa.int/documents/247904/1877131/Sentinel-1-Product-Definition>
- [42] GDAL/OGR Contributors. (2022). *GDAL/OGR Geospatial Data Abstraction software Library*. Open Source Geospatial Foundation. [Online]. Available: <https://gdal.org>
- [43] W. Shi, M. Zhang, R. Zhang, S. Chen, and Z. Zhan, “Change detection based on artificial intelligence: State-of-the-art and challenges,” *Remote Sens.*, vol. 12, no. 10, p. 1688, May 2020. [Online]. Available: <https://www.mdpi.com/2072-4292/12/10/1688>
- [44] L. T. Luppino, F. M. Bianchi, G. Moser, and S. N. Anfinsen, “Unsupervised image regression for heterogeneous change detection,” *IEEE Trans. Geosci. Remote Sens.*, vol. 57, no. 12, pp. 9960–9975, Dec. 2019.
- [45] J. Wang, F. Gao, J. Dong, S. Zhang, and Q. Du, “Change detection from synthetic aperture radar images via graph-based knowledge supplement network,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 1823–1836, 2022.
- [46] Y. Du, R. Zhong, Q. Li, and F. Zhang, “TransUNet++SAR: Change detection with deep learning about architectural ensemble in SAR images,” *Remote Sens.*, vol. 15, no. 1, p. 6, Dec. 2022. [Online]. Available: <https://www.mdpi.com/2072-4292/15/1/6>
- [47] J. Inglada and G. Mercier, “A new statistical similarity measure for change detection in multitemporal SAR images and its extension to multiscale change analysis,” *IEEE Trans. Geosci. Remote Sens.*, vol. 45, no. 5, pp. 1432–1445, May 2007.
- [48] S. Cui, G. Schwarz, and M. Datcu, “A benchmark evaluation of similarity measures for multitemporal SAR image change detection,” *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 9, no. 3, pp. 1101–1118, Mar. 2016.
- [49] National Land Survey of Finland. *Topographic Database*. Accessed: Jan. 27, 2023. [Online]. Available: <https://www.maannittauslaitos.fi/en/maps-and-spatial-data/expert-users/product-descriptions/topographic-database>
- [50] S. Seabold and J. Perktold, “Statsmodels: Econometric and statistical modeling with Python,” in *Proc. 9th Python Sci. Conf.*, 2010, pp. 92–96.
- [51] I. Loshchilov and F. Hutter, “Decoupled weight decay regularization,” 2017, *arXiv:1711.05101*.
- [52] M. Abadi et al., (2015). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. [Online]. Available: <https://www.tensorflow.org/>
- [53] F. Pedregosa et al., “Scikit-learn: Machine learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.



**Janne Alatalo** received the B.Eng. and M.Eng. degrees in information technology from the JAMK University of Applied Sciences, Jyväskylä, Finland, in 2014 and 2023, respectively.

He is currently a Specialist at the JAMK University of Applied Sciences. His research topics include applied artificial intelligence and cybersecurity. The working background includes multiple research and development projects at the JAMK University of Applied Sciences, where he has applied his software engineering, data analytics, and artificial intelligence skills to many real-world problems. Most recently, he has worked in the field of remote sensing, where radar satellite images were used to detect storm damages to forested areas. His research interests include artificial intelligence and data analytics.



**Tuomo Sipola** received the M.Sc. degree in information technology, the Ph.D. degree in mathematical information technology, and the M.A. degree in Latin from the University of Jyväskylä, Jyväskylä, Finland, in 2009, 2013, and 2020, respectively.

His earlier work experience includes research positions at the University of Jyväskylä and CEO duties in his startup company, CAP Data Technologies, Jyväskylä. He currently works as a Senior Researcher at the Institute of Information Technology, JAMK University of Applied Sciences, Jyväskylä. He was an Editor of the book *Artificial Intelligence and Cybersecurity: Theory and Applications* (Springer, 2023). He has worked in both research and commercial projects in many application areas, including predictive maintenance, network security, and medical imaging. His research interests include machine learning, data analytics, and cybersecurity.



**Mika Rantonen** received the M.S. degree in mathematics and the Licentiate of Technology degree in system engineering from the University of Oulu, Oulu, Finland, in 2000 and 2008, respectively.

He was a Software Architect at Nokia Mobile Phones, Oulu, Finland, and a Research Scientist at VTT Technical Research Centre of Finland, Oulu. He is currently a Principal Lecturer of artificial intelligence and data analytics, the Head of the master's degree program, the Project Manager of several AI- and DA-based research and development projects at the JAMK University of Applied Sciences, Jyväskylä, Finland. He has been working in the international and domestic research and development projects in various roles and has managed multiple projects in different domains and has an extensive knowledge of different funding programs. His research interests include artificial intelligence, data analytics, big data, machine learning, and deep learning.