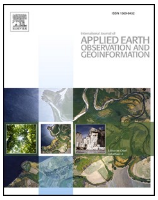
Contents lists available at ScienceDirect

International Journal of Applied Earth Observations and Geoinformation

journal homepage: www.elsevier.com/locate/jag



Deep learning based distributed scatterers acceleration approach: Distributed scatterers prediction Net

Duo Wang [*], Markus Even, Hansjörg Kutterer

Geodetic Institute, Karlsruhe Institute of Technology, Karlsruhe 76131, Germany

ARTICLE INFO

Keywords:
Deep Learning
InSAR
Distributed Scatterer
CNN
DSPN

ABSTRACT

Distributed scatter (DS) interferometric synthetic aperture radar is a powerful technology for analyzing displacements of the earth's surface. Unfortunately, the preparatory step of DS pre-processing is enormously time consuming. The present research puts forward a deep learning-based approach called Distributed Scatterers Prediction Net (DSPN), that can reduce the computational load considerably. DSPN is a convolutional neural network, which generates DS candidate masks based on nine input layers. Masked pixels with low prospect of being DS are omitted during DS pre-processing. Tests on 6 different terrains in North Rhine-Westphalia and Sicily with Sentinel-1 data show that DSPN saves 11% to 87% computation time depending on the scene without significantly reducing coverage with information. Our experiments show that the proposed approach can effectively predict DS candidates and speeds up processing, indicating its potential for analyzing the big data of remote sensing. To the best of our knowledge, this is the first attempt to do a classification in DS candidates and non-DS candidates as a preparatory step to DS pre-processing.

1. Introduction

Distributed scatterer (DS) pre-processing based on grouping statistically homogeneous pixels, estimating the coherence matrix and extracting the optimized phase history has been an object of research efforts of the InSAR community since more than a decade now (De Zan and Rocca, 2005; Guarnieri and Tebaldini, 2007; Samiei-Esfahany et al., 2016; Jiang et al., 2017; Lin and Perissin, 2017; Ansari et al., 2018; Even and Schulz, 2018; Michaelides et al., 2019; Zhao et al., 2019; Even, 2021; Ansari et al., 2021; Even, 2022; Ferretti et al., 2011; Zheng et al., 2022; Zwieback and Meyer, 2022; Perissin and Wang, 2012; Rocca et al., 2013; Jiang et al., 2014; Jiang et al., 2015; Fornaro et al., 2015; Zwieback, 2016; Wang and Zhu, 2016; Jiang and Guarnieri, 2020; Ansari et al., 2017). A review is (Even and Schulz, 2018). Put in a nutshell, one might think of it as transforming DS into persistent scatterers (PS), which then can be used together with actual PS in any PS algorithm. It allows, depending on the physical characteristics of the earth's surface, to increase significantly the coverage with valuable information. But the benefits of DS pre-processing are paid with an enormous computational load, in particular considering that data stacks become ever bigger due to, e.g., the Sentinel-1 mission. Hence, ways to accelerate processing are sought. On the technical side, more powerful computers, parallelization, and optimized implementations help to achieve that. On the side of selection of building blocks of the DS pre-processing algorithm, grouping methods like FSHP (Jiang et al., 2015) or the likelihood ratio test used in (Jiang and Guarnieri, 2020) outperform the Kolmgorov-Smirnov two sample test regarding speed and quality of results and the Eigendecomposition-based Maximum-likelihood-estimator of Interferometric phase (EMI) (Ansari et al., 2018) allows for a fast estimation of the DS signal, while retaining good quality of the estimation. In addition, sequential estimation is an innovative algorithm, which uses subdivision and compression of mini-stacks (Ansari et al., 2017) and is able to speed up processing significantly. Trivially, also parameter settings influence processing time. An obvious example is the size of the search window. In the present work, a new approach to accelerate DS pre-processing by DS Candidates (DSC) pre-selection is proposed. The idea is to employ deep learning in order to predict which pixels are non-DS before DS pre-processing is performed. As during DS pre-processing each pixel is analyzed separately, the non-DS identified beforehand do not need to be analyzed and their processing time can be saved. Depending on the data stack, this can significantly reduce the computational load and justify the costs of a classification step. The prediction does not have to be perfectly precise to be useful. It only needs to avoid discarding DS and needs to discard sufficiently many non-DS. Even the

* Corresponding author.
E-mail address: duo.wang@kit.edu (D. Wang).

https://doi.org/10.1016/j.jag.2022.103112
Received 22 July 2022; Received in revised form 3 November 2022; Accepted 14 November 2022
Available online 18 November 2022
1569-8432/© 2022 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

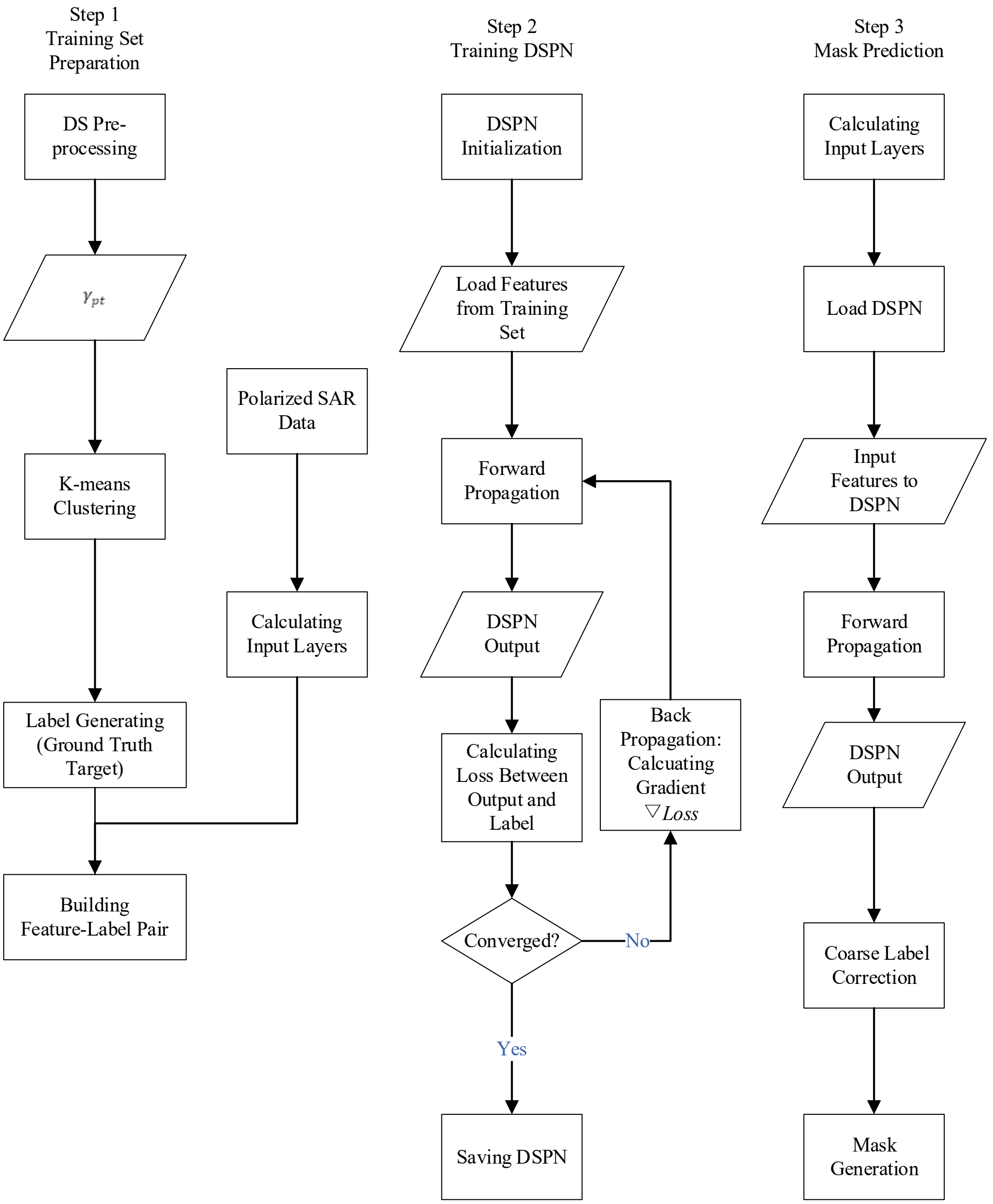


Fig. 1. The general flow chart of DSPN approach.

loss of DS pixels can be tolerated if the spatial coverage is not significantly affected. Indeed, because DS pixels naturally occur in groups, it is common to thin out pixels after combining PS and DS in order to reduce the computational load during subsequent processing steps.

Although important characteristics (e.g. phase triangulation coherence γ_{pt} (Ferretti et al., 2011) and size of neighborhood) that serve to identify good DSC are only available after DS pre-processing and although information from all possible interferograms is used to determine them, it is not necessary to feed the complete data stack in the deep artificial neural network (ANN). A selection of a modest number of input layers will be sufficient for two reasons: First, important classes of land cover, e.g. forest or water, will never constitute DS, at least not for smaller wavelengths as X- or C-band. Second, land cover classification (LCC) based on SAR data is a well-developed field of research that achieves good results; all the more, as deep learning approaches have lately set new standards in SAR image processing (Zhu et al., 2021). Hence it should be possible to recognize at least those non-DS pixels associated with land cover classes that can be successfully discerned like forest (Geng et al., 2017; Mazza et al., 2019; Ban et al., 2020), and water (Chen et al., 2020; Mayer et al., 2021; Guo et al., 2022). While LCC is related to this work, our goal is a direct classification in DSC and non-DSC that does not require a LCC as intermediate result. Only the selection of some of the input layers is motivated by work on LCC. The input layers will be discussed in section 2.2.2 in more detail.

Deep learning is a machine learning technique that uses deep Artificial Neural Networks (ANN). With the development of GPU technology and parallel computing, it has become successful in computer vision, speech recognition and other domains (Lecun et al., 2015; Goodfellow et al., 2016), often representing the state of the art in these fields. Since several years, also the remote sensing community is increasing research activity regarding deep learning (Zhu et al., 2021; Zhu, 2017; Yuan et al., 2020). In particular, deep learning has been applied to SAR data for different purposes: to SAR intensities for speckle filtering, e.g. (Lattari et al., 2019; Cozzolino et al., 2020), for image classification (Geng et al., 2017), for vegetation mapping (Ho Tong Minh et al., 2018), waterline detection (Zhang et al., 2022), to PolSAR data for LCC, e.g. (Xie et al., 2014; D. Ho Tong Minh, E. Ndikumana, N. Baghdadi, D. Courault, and L. Hossard, “Applying deep learning for agricultural classification using multitemporal SAR Sentinel-1 for Camargue, France,” vol., 2018; Ma et al., 2019; Parikh et al., 2020)characterizing arctic tundra hydro-ecological conditions (Merchant et al., 2022), or for near real time wildfire monitoring (Ban et al., 2020), for road extraction e.g. (Henry et al., 2018; Zhang et al., 2019), to InSAR data for detection of deformation, e.g. (Gaddes et al., 2019; Sun et al., 2020; Anantrasirichai et al., 2021), for phase filtering and coherence estimation, e.g. (Sun et al., 2020; Pu et al., 2020; Mukherjee et al., 2021), for PS selection (Tiwari et al., 2020), and for removing atmospheric phase from InSAR time series (Zhao et al., 2021).

Deep learning means that a function, which performs a certain task, is obtained by approximately presenting it as ANN with many weights/parameters that are determined by training with example data. For the DS Prediction Network (DSPN) proposed in this paper, supervised learning was applied. This means that the training data consisted of input data plus (as ground truth) the wished output of the ANN. During training, a so-called loss function measures how well the prediction of the ANN agrees with the ground truth for the training data. The training is formally a mathematical optimization that adjusts the weights of the ANN until the loss function attains small values. However, other than for optimization problems, the goal of training is not to find the minimum loss, but instead to perform reliably on new, unseen inputs.

In our case, we want to approximate a function that is hard to be formulated in simple mathematical terms. It would make the decision DS or non-DS based on the chosen input layers. Nevertheless, we can make DSPN an approximation to this function via the training data based on γ_{pt}. The calculation of γ_{pt} with the DS pre-processing algorithm itself can be seen as an inspiration for such a function that uses the whole data stack as input. As volume and variability of number of layers makes the whole data stack unsuitable for our purpose, we restrict the input to some selected layers. Above, we argued that these should suffice to discern DS and non-DS.

For this work, the ground truth is produced based on DS pre-processing results. This requires an answer to the question which pixels are considered being a DSC. We decided to solely use phase triangulation coherence γ_{pt} as the basis of characterization, from which the ground truth is generated using K-means clustering. Nevertheless, γ_{pt} alone is not sufficient to characterize DS. Presumably, this is the cause that a considerable number of false negatives are observed. When plotted in Google Earth, most of them appear over vegetated areas and are hence unlikely to be DS. But what matters in the end regarding applicability of DSPN, is the loss of coverage of an InSAR result generated discarding non-DS according to DSPN compared to a result, where DS pre-processing was run on the full dataset. In order to evaluate the loss of coverage due to discarding false negatives, such InSAR analyses were performed with a modified version of StaMPS (Even et al., 2020) which allows to jointly process PS Candidates and DS Candidates.

In chapter 2, the formal framework of DSPN, coarse label learning, training of the network and coarse label correction are explained. Furthermore, data and the computing environment are described. Chapter 3 analyses the results obtained for several test cases from North Rhine-Westfalia (Germany) and Sicily (Italy) regarding quality and runtime, showing a significant gain in speed. The discussion in chapter 4 is devoted to the question of efficiency, an analysis of false negatives, and possible future work. The efficiency proves to be good and an acceptable loss in coverage of less than 3 % is observed. Finally, our conclusions are presented.

2. Material and methods

2.1. Formal framework of DSPN

The framework of DSPN contains three parts, which are “training set preparation”, “training DSPN”, and “mask prediction“, as shown in Fig. 1. In section 2.2, the method for preparing the training set will be explained. The composition of the input layers and the generation of coarse labels (ground truth) are the focus of this part of the research. Then the DSPN is trained iteratively as step 2 showed in Fig. 1; the architecture of DSPN, training strategy, and loss function are provided in section 2.3. When the training is finished, the model can be used to predict the mask of DSC. The approach for converting the output of DSPN into the mask available for StaMPS is called coarse label correction. It is given in section 2.4.

2.2. Coarse label learning

2.2.1. Label generation

Since there is currently no dataset that provides supervised information for DS recognition, this research proposes a method of clustering γ_{pt} to generate labels for training. The label of a pixel contains the information of whether this pixel belongs to DSC or not. Even though setting γ_{pt} as the label directly or setting a hard threshold of γ_{pt} as the label is an uncomplicated way, experiments prove the network will hardly converge or it will overfit if the value of γ_{pt} is used as the label and will also miss details with low γ_{pt} like roads or motorways (Appendix B). After investigating the γ_{pt} distribution in four different regions (Ruhrgebiet, Saarland, Ibbenbüren and Sicily), a hypothesis is worth to consider: It is supposed that the patterns of DSC follow a similar distribution for different scenarios. For example, DSCs can be roughly classified into weak DS (road, motorway, etc.) and strong DS (urban area) according to their γ_{pt} values. Although the γ_{pt} value of the same type of DSCs from different stacks and different parameters are variable, their quantile values are close to each other.

Based on this hypothesis, an idea is to let the network learn the categories of DSCs but not the value of γ_{pt}. The categories of DSC are derived from γ_{pt} maps. From prior experience, for points with γ_{pt} less than a threshold (here 0.4), the possibility of becoming DSCs is very small (Even and Schulz, 2018), section 3.3. So an idea is to set such points as non-DS labels and eliminate them in the mask. For those DSCs ($\gamma_{pt} > 0.4$), the K-means algorithm was used to generate multi-category coarse labels for training because we want the network to learn a decision hyperplane as clearly as possible to classify the diverse types of DSCs and enhance the robustness and the sensitivity to weak DSCs. Although the threshold 0.4 is a bit low, it is appropriate for this setting because more DSC is unharmful, while losing coverage is harmful for post-processing in StaMPS.

Since K-means is a non-convex algorithm, the best categories number K is found by a systematic “trial and error” approach. For γ_{pt}, it will hardly converge if the cluster number K is higher than a threshold K_{max} (from experiment we recommend $K_{max} \leq 6$). Hence, the best K can be searched from 2 to K_{max}. In order to determine the best K, we applied three cluster evaluation methods: Calinski-Harabasz index (Caliñski and Harabasz, 1974), Silhouette analysis (Rousseeuw, 1987), and Elbow Curve Method (Ketchen and Shook, 1996). In case that the best K from different cluster evaluation methods is different, we selected the K, for which the histogram counting points in the classes retains best the shape of the distribution of values of γ_{pt}. Furthermore, if the size of a cluster is much smaller than that of the others, it is merged into the nearest one. This is helpful for the neural network to fully learn the features of different classes.

2.2.2. Input layers

The Distributed Scatterers Prediction Net (DSPN) learns the DS

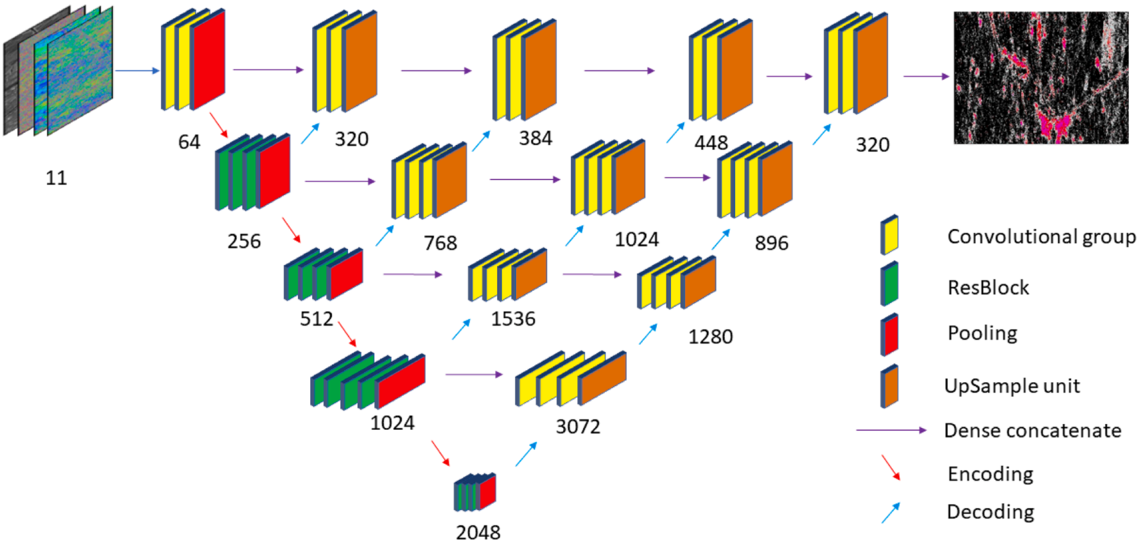


Fig. 2. DSPN structure. The number below each block indicates the input channel for that block.

Table 1
SAR images stacks details for North Rhine-Westphalia and Sicily.

Dataset	*Path*	*Master date*	*Period*	*Low coherence date*	*High coherence date*	*Size*
North Rhine-Westphalia	*15 Ascending*	*2018.3.01*	*2018.1.6–2021.3.21*	*2017.5.11*	*2018.2.17*	*4000*9000*
Sicily	*44 Ascending*	*2017.1.07*	*2017.10.08–2018.9.29*	*2017.4.19*	*2017.1.13*	*15000*11750*

Table 2
Test regions details for North Rhine-Westphalia and Sicily.

Dataset	*Test region*	*Topography*	*Size*
North Rhine-Westphalia	*Wickede (Ruhr)*	*River, Countryside*	*500*1000*
North Rhine-Westphalia	*Hamm*	*Urban*	*500*2000*
North Rhine-Westphalia	*Münster*	*Motorways*	*500*1000*
Sicily	*Arenella*	*Urban, Harbor, Beach, Sea*	*1500*500*
Sicily	*South side of Etna*	*Crater-volcano (slope and shadow)*	*3500*3000*
Sicily	*Rocche d'Argimusco*	*Forests (slope and shadow)*	*1500*1500*

patterns from the statistical characteristics of the raw signal. In the beginning, an initial choice of candidate input layers for DSPN was done, inspired by the input layers that other authors used for similar tasks. E. g., (Mestre-Quereda et al., 2020; Nikaein et al., 2021) compare different constellations and come to the conclusion that coherence and backscatter are complementary and that use of dual-polarization yields better results than only single polarization. Both papers concern crop classification and use time series to achieve multi-class categorization. Another type of land cover that is relevant with regard to DSCs are roads. They often provide DSCs of lower quality that nevertheless are valuable as they often provide additional coverage, e.g. in Central Europe. (Zhang et al., 2019) found that the use of dual-pol data can also improve road extraction. (Xiao et al., 2019) used the coefficient of variation as one of the input layers for this purpose. For such candidate input layers, we experimented with expert rules to distinguish between different types of land cover (having in mind the generation of training data). Based on this experience, the final selection for DSPN comprised the following nine input layers: speckle filtered intensities of the master regarding VV and VH; real and imaginary parts of coherences from an acquisition of a month with low coherence and a month with high coherence; median amplitude over the stack; filtered amplitude dispersion image of the stack; and coefficient of variation of median amplitude.

The intensities are the output of the polarimetric speckle filtering function of the Sentinel 1 toolbox (Improved Lee Sigma filter using default parameters one look, window size 7 × 7, sigma 0.9, target window size 3 × 3). The coherences were calculated for each acquisition from a month with low coherence (strong vegetation) and a month with high coherence (weak vegetation, short temporal baseline). For the calculation of the coherences, an 11 × 3 window was used. The median amplitude and the amplitude dispersion image of the stack do not pose an additional computational burden as they are used as the background image for plotting or are required for PS selection and are calculated either way. The median amplitude gives a low noise version of backscatter. We intended a pan-sharpening effect in connection with the ANN. Low amplitude dispersion is often used as an indicator of phase stability that helps detect PSC. Higher amplitude dispersion is found in fields and meadows. In order to obtain a layer with easier to classify information a median filter (window size 5 × 3) that exempts pixels with amplitude dispersion smaller than 0.4 was applied. The size of the median filter is a trade-off between losing resolution and being large enough to observe the desired smoothing effect. The coefficient of variation of median amplitude (7 × 3 window) was calculated to highlight in particular roads.

2.3. Training DSPN

2.3.1. Distributed scatterers prediction Net architecture

Since the network needs to identify the class of each pixel and its contribution to the loss function, the use of an Encoder-Decoder network structure is self-evident. The structure of DSPN is composed of two parts, the encoder branch, and the decoder branches. For the encoder, the backbone of a convolutional network with multiple level feature extractors can be considered used such as AlexNet (Krizhevsky et al., 2017); GoogLeNet (Szegedy, et al., 2015), or VGG (Simonyan and Zisserman, 2015). We recommend to use ResNet (He et al., 2016) as the backbone of the feature extractor because it provides four different levels of features from global to local by four ResBlocks (implemented

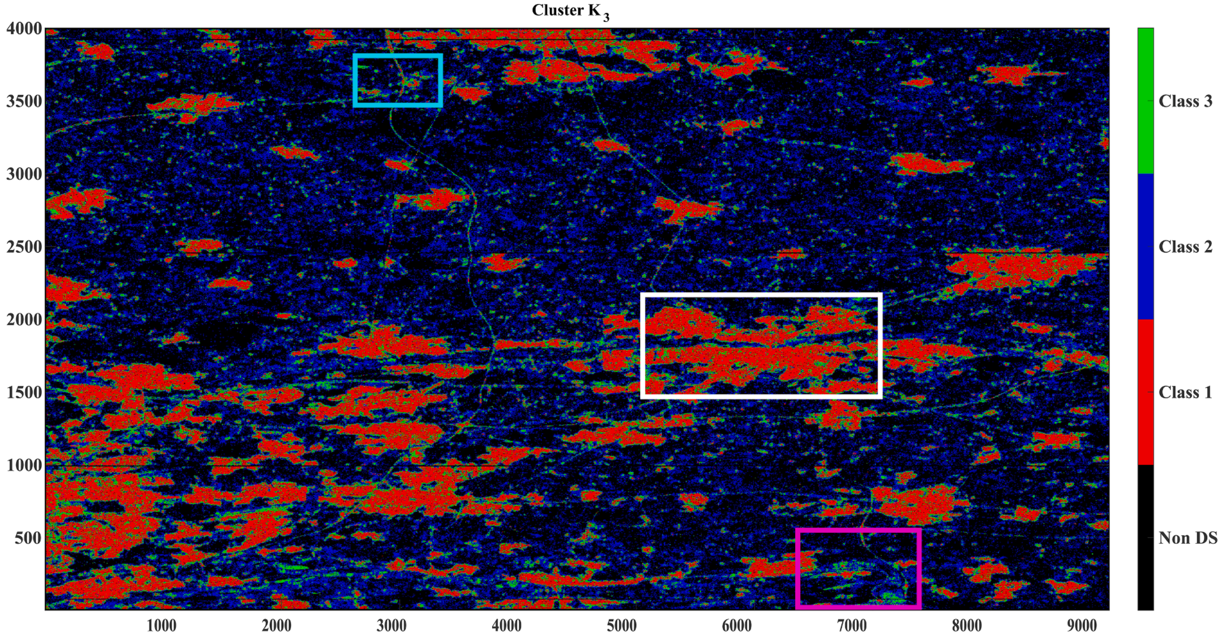


Fig. 3. The clustering result for North Rhine-Westphalia; the white rectangle marks the test region in city Hamm, the cyan rectangle marks the test region in the motorway intersection near Münster and the purple rectangle marks the test region in Wickede (Ruhr). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

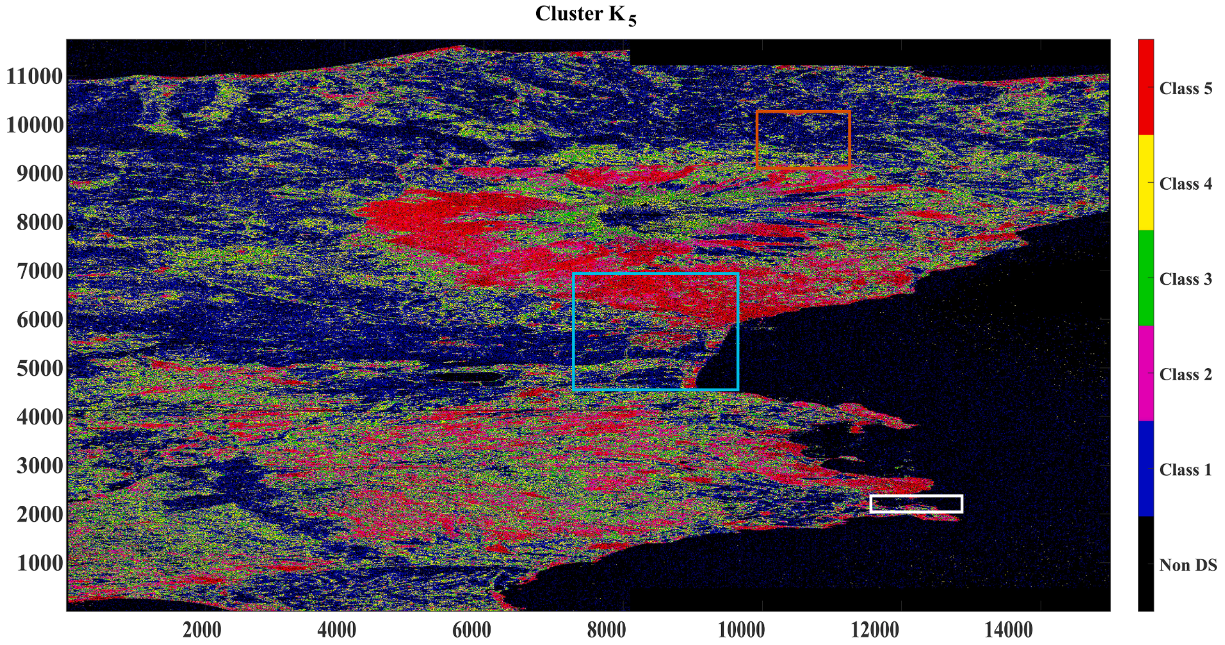


Fig. 4. The clustering result for Sicily; the white rectangle marks the test region in Arenella, the cyan rectangle marks the test region in the south side of Etna and the brown rectangle marks the test region in Riserva Naturale Orientata Bosco di Malabotta. Rocche d'Argimusco. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 3
Evaluation results of North Rhine-Westphalia.

Region	TP	TN	FP	FN	Accuracy(%)	Precision (%)	Recall (%)	NPV (%)	FNR (%)
Wickede (Ruhr)	172,830	295,501	0	31,669	93.67	100	84.51	90.32	15.49
Münster	181,097	292,371	0	26,532	94.69	100	87.22	91.68	12.78
Hamm	743,338	240,795	0	15,867	98.41	100	97.91	93.82	2.09

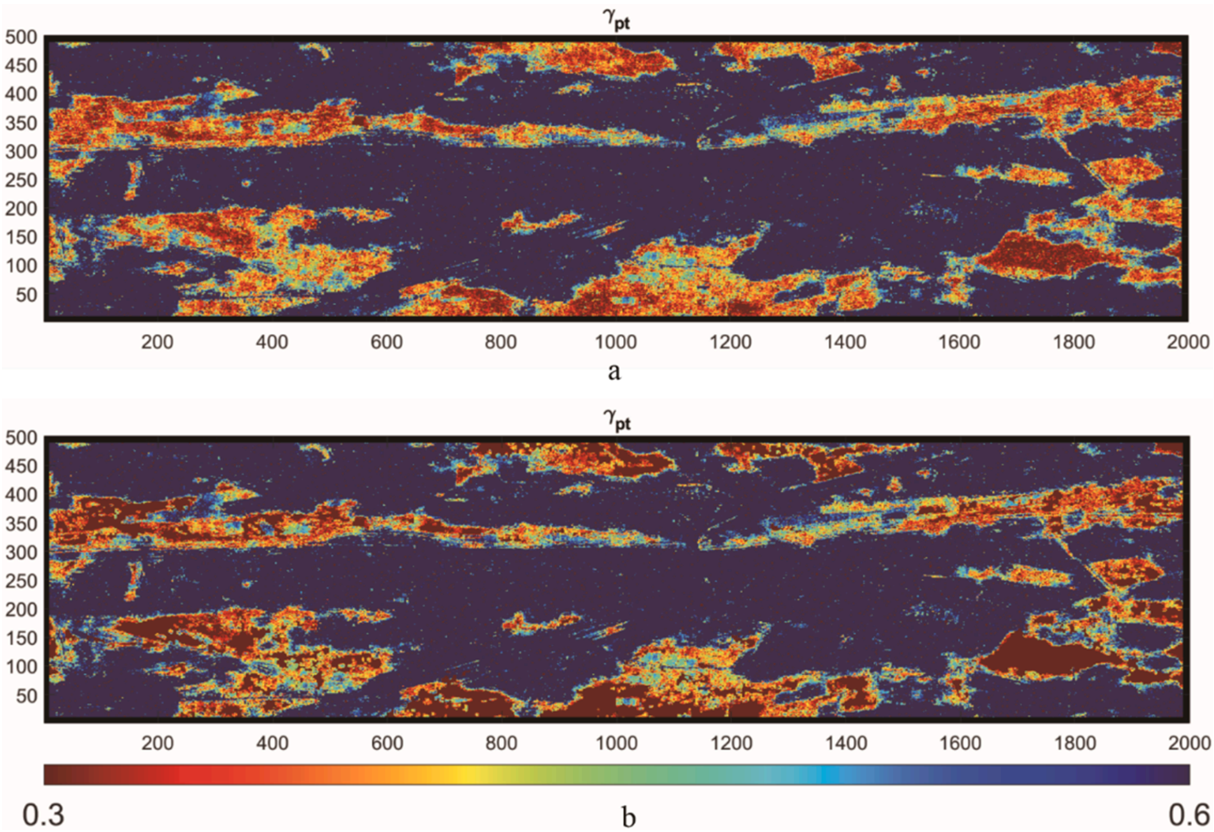


Fig. 5. γ_{pt} after pre-processing in Hamm – (a) without DSPN; (b) with DSPN.

with Bottleneck (He et al., 2016), which mitigate the vanishing gradient problem. The decoding path consists of several decoding units, each of which consists of a set of 3*3 convolutions and an up-sampling unit. Since up-sampling with deconvolution will cause "checkerboard artifacts" (Odena et al., 2017), we use bilinear interpolation and a 1*1 convolutional layer instead of deconvolution to construct the up-sampling unit. Different from V-net (Milletari et al., 2016) and U-net (He et al., 2015), we use dense concatenate (Zhou et al., 2018) to merge the features from different decoding units in different decoding paths but not concatenate them directly. Since the features used by different decoding paths have different receptive fields, using dense skipping to merge decoding results can more completely use the encoding information of different scales to improve the network's ability to perceive details (e.g. small roads or motorways). The structure of DSPN was shown in Fig. 2 and the detail of the layers was listed in the Appendix A.

2.3.2. *Training strategy and optimizer*

The DSPN is trained by the Back Propagation algorithm (Rumelhart and Hintont, 1986), which uses the gradient information from loss to update the weight of each layer of the network. For the optimizer selection, (Keskar and Socher, 2017) provides a good strategy to combine the advantage between Adam (Kingma and Ba, 2015) and SGD (Rumelhart and Hintont, 1986). Although the adaptive learning rate allows the loss function to converge quickly, some experiments from (Keskar and Socher, 2017) showed that the generalization ability of Adam is not better than SGD. Hence, in the early stage of training, we use the Adam optimizer for warm-up, then we use Momentum SGD (Rumelhart and Hintont, 1986) to do the fine tuning in order to get better accuracy and better generalization performance. If a pre-trained model is available, just use SGD for transfer learning without having to reinitialize the network and retrain when a new region needs to be processed. Compared to retraining, transfer learning only requires less time to converge. This improves the generalization ability of the DSPN while reducing the training time.

2.3.3. *Loss function*

The cross-entropy loss function has shown its success in image classification tasks, but this loss function cannot distinguish the contribution from different labels. The purpose of the network is to classify whether the point belongs to DSCs, but for the coarse label generated by section 2.2.1 that belong to DSCs, misclassifying them to another category that also belongs to DSCs does not affect the result. Therefore, this research puts forward weighted coarse label cross-entropy as the loss function for coarse label learning:

$$H_w(p,q) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{c=1}^{K}p(x_i,c)\bullet w_c\bullet logq(x_i,c) \tag{1}$$

$$w_c = \begin{cases} 1 & if \quad c = 1(non-DS) \\ \dfrac{N_c}{\sum_{k=2}^{K}N_k} & else \end{cases} \tag{2}$$

where $p(x_i,c)$ refers to the ground truth probability of x_i belonging to DS generated from the K-means clustering, and $q(x_i,c)$ refers to the predicted probability of x_i belonging to class c from the network, N_c is the number of points belonging to category c and class $c = 1$ denotes the non-DSCs. By using binarization encoding, the $p(x_i,c)$ has been encoded to binarization label $y_{i,c}$, as Equation (3) shows:

$$y_{i,c} = \begin{cases} 1 & (p(x_i,c) = 1 \wedge c = 1) \vee c > 1 \\ 0 & otherwise \end{cases}, \tag{3}$$

and the predicted probability $q(x_i,c)$ can be written as the output of the network with a SoftMax operator:

$$q(x_i,c) = \hat{y}_{i,c} = SoftMax(f_w(x_i))_c \tag{4}$$

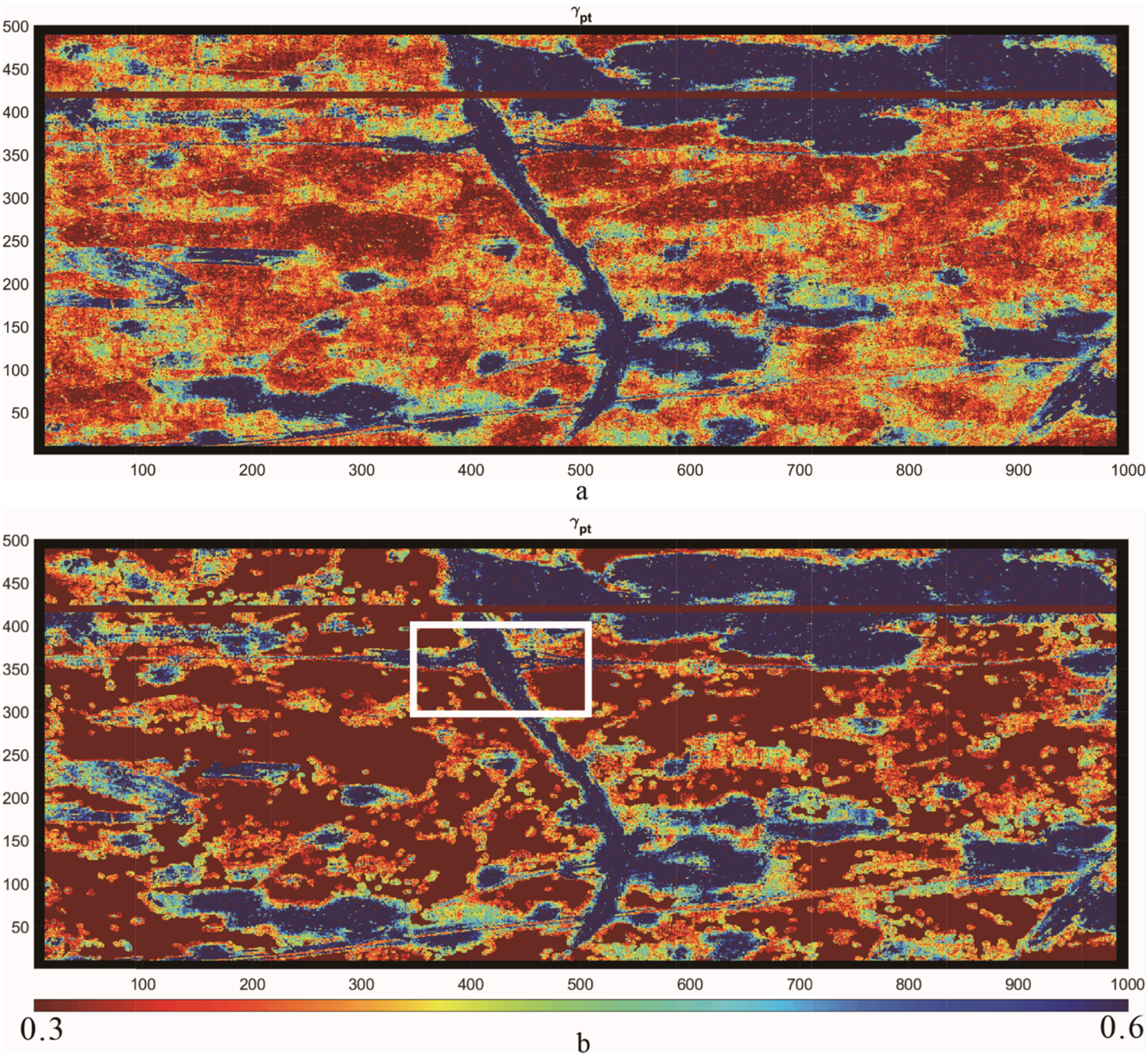


Fig. 6. γ_{pt} after pre-processing in Münster – (a) without DSPN; (b) with DSPN; the white rectangle marks the overpass between Highway 1 and Highway 43.

Since the sample contributes to the loss function only when the indicator function is 1, the weighted coarse label cross-entropy $H_w(p,q)$ can be simplified as:

$$H_w(p,q) = -\frac{1}{N}\sum\nolimits_{i=1}^{N} h_w(x_i,y_i) \tag{5}$$

$$h_w(x_i,y_i) = \begin{cases} -log\widehat{y}_{i,1} & y_i = 1 \\ -\sum\nolimits_{c=2}^{K}\frac{N_c}{\sum\nolimits_{k=2}^{K}N_k}log\widehat{y}_{i,c} & y_i > 1 \end{cases} \tag{6}$$

2.4. Coarse label correction

After training, the network can make predictions for unknown scenes. The output of DSPN is an M*N*K tensor P, where $P[m,n,k]$ is the probability for pixel $[m,n]$ to be in class k. Since we only need to filter out non-DSC points, rather than the coarse label classification map, the mask can be obtained from the tensor P with coarse label correction. The approach of coarse label correction is as follows:

$$Mask[m,n] = \begin{cases} 0 & P[m,n,1]\rangle\sum\nolimits_{i=2}^{K}P[m,n,i] \\ 1 & otherwise \end{cases} \tag{7}$$

Since the DSC is composed of a set of points sharing the scattering mechanism, the mask obtained by Equation (7) also needs to be morphologically corrected. For the real DS point, the number of neighborhoods should be 20 or higher (Ferretti et al., 2011), which means that isolated points with small neighborhoods will hardly become DS. In order to keep coverage, the threshold of the neighborhood for the DSPN mask is set as 5, and points with neighborhoods smaller than 5 will be set as non-DS. After that, a dilate operation will be used for the mask to avoid the loss of DSC near the edges.

2.5. Data and experiment environment

To validate the proposed method, two datasets built from Sentinel-1 IWS mode with different topography were used to compare the performance and generalization ability of our approach, those are North Rhine-Westphalia in Germany and Sicily in Italy. Six test regions were

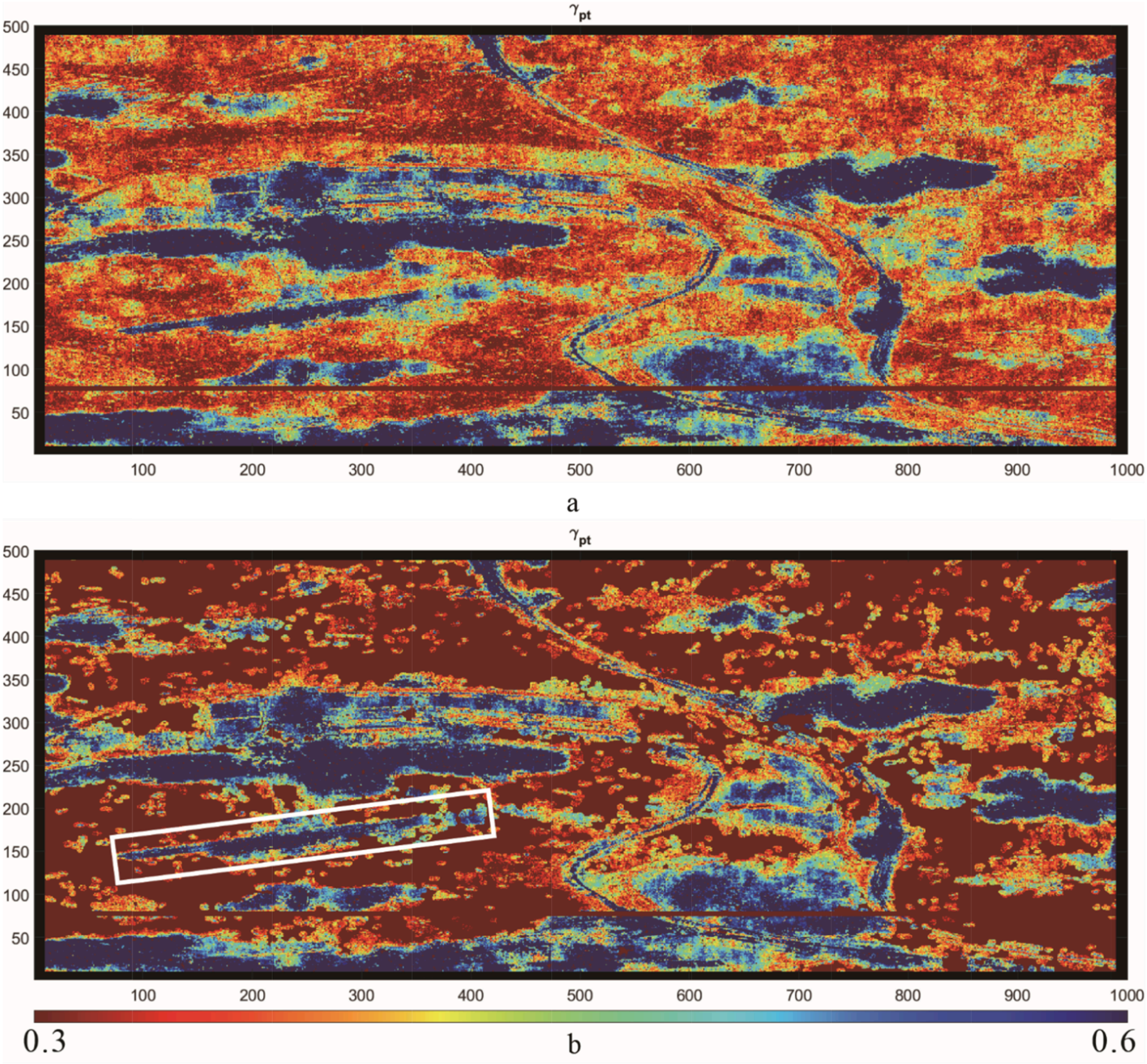


Fig. 7. γ_{pt} after pre-processing in Wickede (Ruhr) – (a) without DSPN; (b) with DSPN; the white rectangle marks the airport Arnsberg Menden.

Table 4
Evaluation results of Sicily.

Region	TP	TN	FP	FN	Accuracy(%)	Precision (%)	Recall (%)	NPV(%)	FNR (%)
Arenella	86,778	661,978	0	1244	99.83	100	98.59	99.81	1.41
South side of Etna	7,369,239	2,962,704	0	168,057	98.40	100	97.78	94.63	2.23
Rocche d'Argimusco	880,722	1,281,276	0	88,002	96.09	100	90.92	93.57	9.08

selected from those two datasets to evaluate the performance of DSPN in a variety of complex terrains. The details of the data sets and test regions are shown in Table 1 and Table 2. The patches cropped for training were taken from the whole scenes exempt the test regions.

The DS pre-processing was performed with the following choices of parameters or sub-algorithms: For grouping, a search window of size 21 pixels times 21 pixels was used. As criterion of similarity, the generalized likelihood ratio test with confidence level 99 % was applied (cp. (Jiang and Guarnieri, 2020) for some theory). The phase history was estimated with help of phase triangulation coherence maximization (Ferretti, 2011) from the coherence matrix, which was obtained from the sample covariance matrix. In order to mitigate biases, the entries of the coherence matrix were taken to the power of two (Ferretti, 2011). Furthermore, sequential estimation with a mini stack size of 20 acquisitions was employed (Ansari et al., 2017). Our choices were based on a study in two parts on how to tune DS pre-processing (Even, 2021; Even, 2022).

The proposed method above was implemented on a deep learning PC with a single NVIDIA RTX 3090 GPU and Intel(R) Core i7-10700 CPU, 8 threads. The DS pre-processing and coarse label generation was implemented by using MATLAB R2020b. The DSPN, weighted coarse label cross-entropy, and coarse label correction were implemented by using

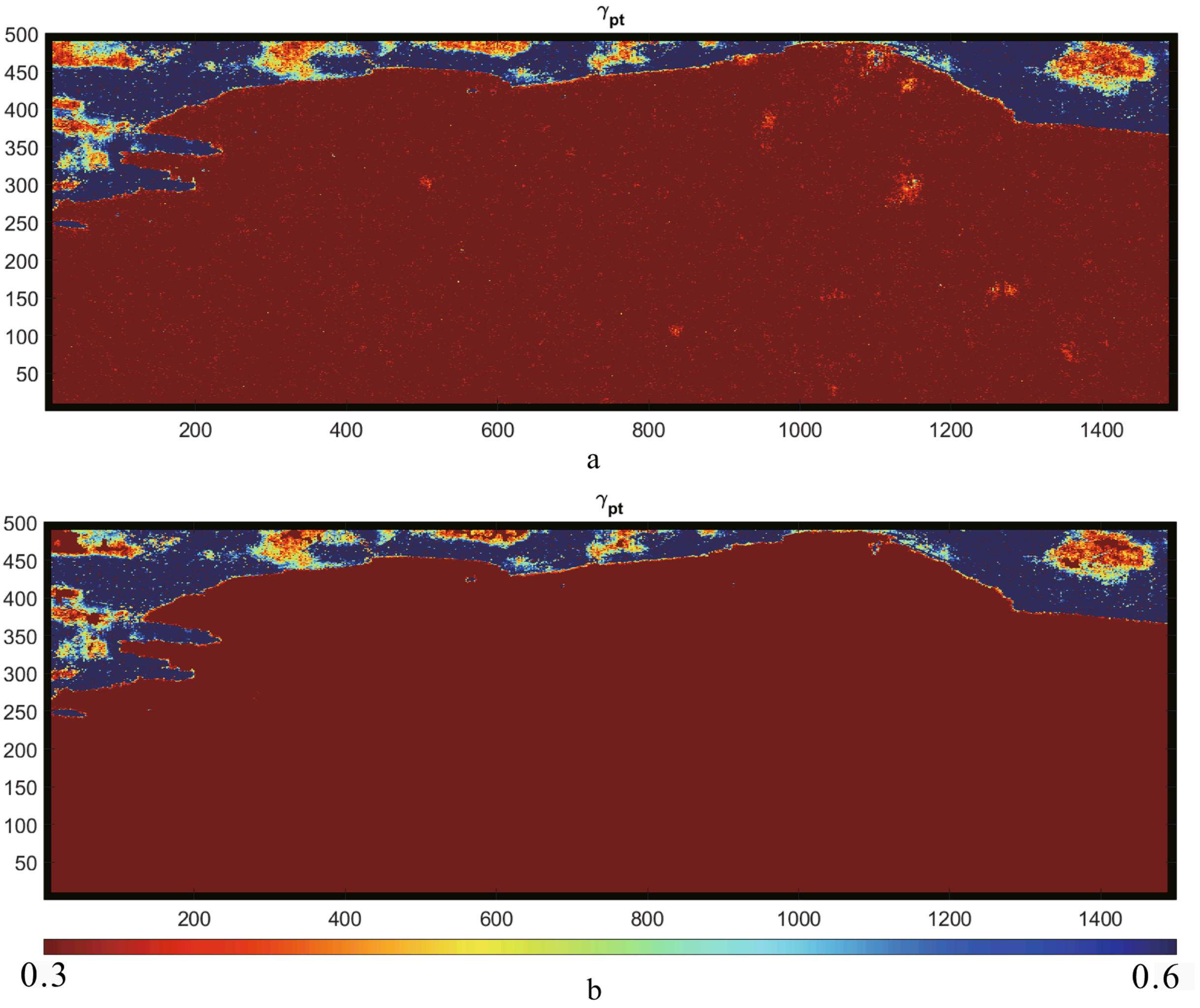


Fig. 8. γ_{pt} after pre-processing in Arenella – (a) without DSPN; (b) with DSPN.

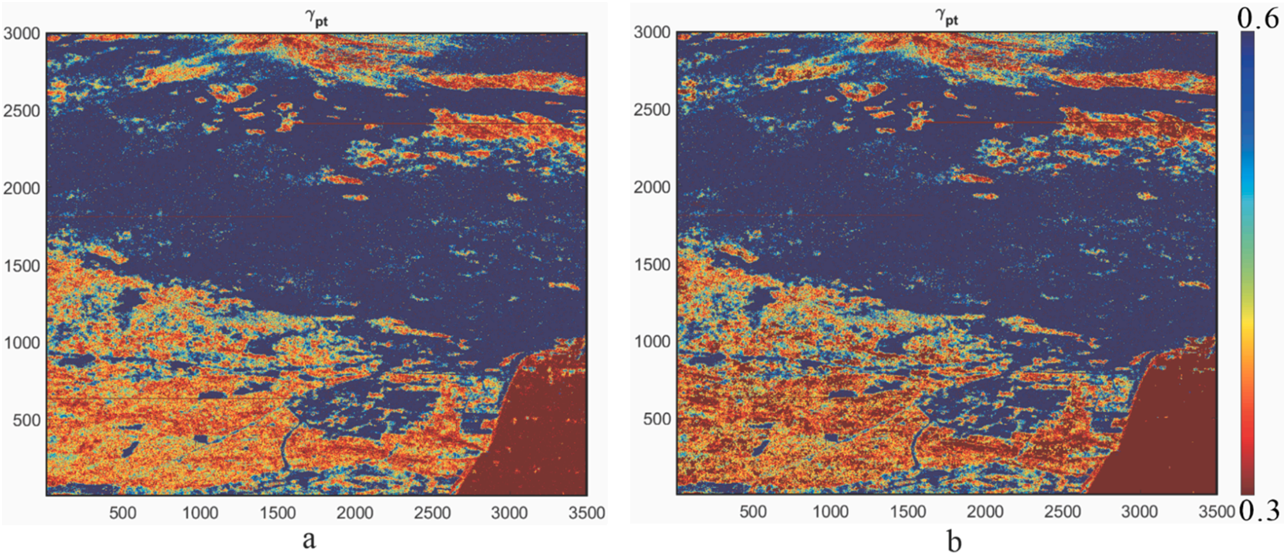


Fig. 9. γ_{pt} after pre-processing in South side of Etna – (a) without DSPN; (b) with DSPN.

Pytorch 1.7.0 build-in operators. Considering the GPU memory constraints, the input layers and corresponding ground truth labels were cropped into 500*500 pixel-sized patches subset stacks.

For North Rhine-Westphalia dataset, the model of DSPN was trained with 2000 epochs. The learning strategy was: Use of Adam optimizer with 0.001 initial learning rate, and for every 500 iterators, the learning

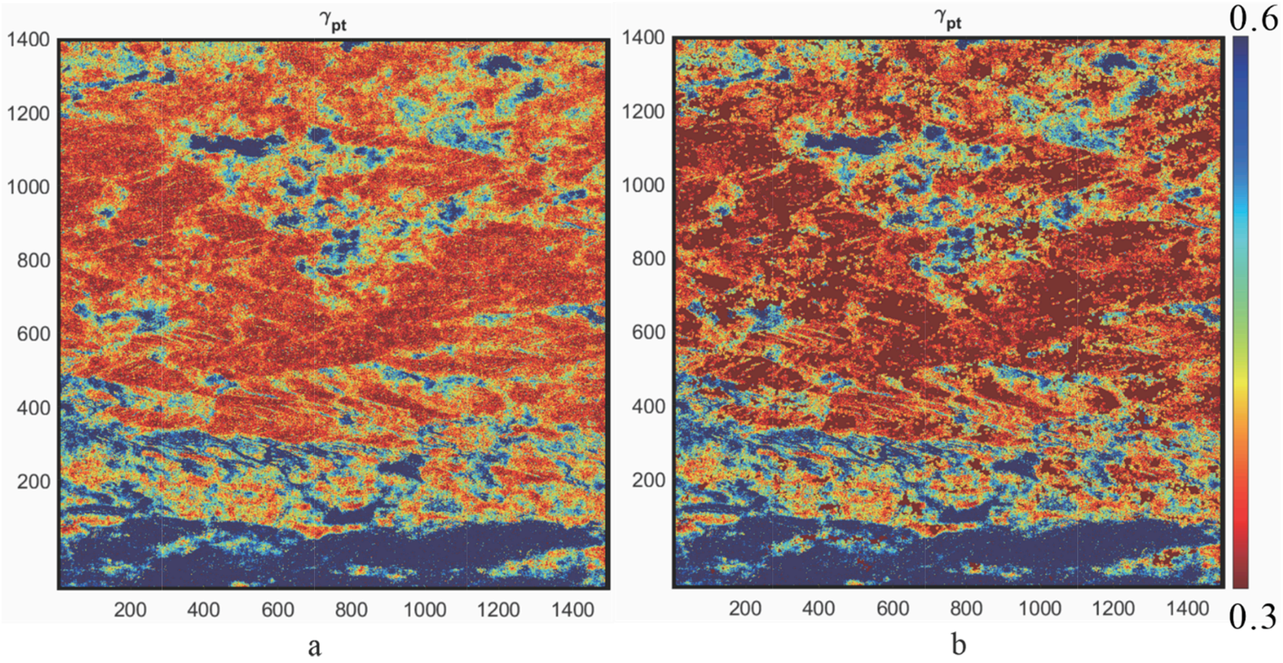


Fig. 10. γ_{pt} after pre-processing in Riserva Naturale Orientata Bosco di Malabotta. Rocche d'Argimusco – (a) without DSPN; (b) with DSPN.

Table 5
The comparison of real-runtime $\boldsymbol{T_r}$ and the total computation time $\boldsymbol{T_t}$ in test regions.

Region	$T_r(ours)/sec$	$T_r(withoutours)/sec$	$T_t(ours)/sec$	$T_t(withoutours)/sec$
Wickede (Ruhr)	9861	14,076	48,089	90,018
Münster	9633	14,222	57,783	88,043
Hamm	19,207	22,405	117,603	133,245
Arenella	7663	13,303	8358	66,271
South side of Etna	80,585	96,596	1,392,960	1,606,706
Rocche d'Argimusco	40,549	48,349	218,120	293,939

Table 6
The comparison between the percentage of Non-DSC and the speed up rate $\boldsymbol{R_{sp}}$ *in test regions.*

Region	$Non-DSC(\%)$	$R_{sp}(\%)$
Wickede (Ruhr)	*42.93*	*46.58*
Münster	*38.68*	*34.37*
Hamm	*9.12*	*11.74*
Arenella	*83.63*	*87.39*
South side of Etna	*11.28*	*13.3*
Rocche d'Argimusco	*23.86*	*25.79*

rate will decay by multiplying 0.7. Training North Rhine-Westphalia model required approximately 49 h of our environment. For each test regions, the mask generating time was no higher than one minute. Although SAR images in Sicily are susceptible to complex terrain such as slope shadows, the generalization ability of DSPN enables the network to work well with only transfer learning. To evaluate the generalization ability of DSPN, we use the fine-tuning strategy, that is: using North Rhine-Westphalia model as initial weight and fine-tune training 500 epochs with SGD, 0.0005 initial learning rate and decay to 0.0001 after 250 epochs. Although the fine-tuning training nearly took 15 h, again the mask was generated in one minute when the training finished.

3. Results

3.1. Results of coarse label generation

After searching, the best K for North Rhine-Westphalia is 3, and the best K for Sicily is 5, a higher value will cause the algorithm to fail to converge. The clustering result is shown in Fig. 3 and Fig. 4.

3.2. Evaluation of DSPN

To evaluate the performance, the numbers of true positives, false positives, true negatives, and false negatives samples are denoted as TP, FP, TN, and FN. Accuracy, precision, recall, Negative Predictive Value (NPV), and False Negative Rate (FNR) are adopted as the metrics of mask quality:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{8}$$

$$precision = \frac{TP}{TP + FP} \tag{9}$$

$$recall = \frac{TP}{TP + FN} \tag{10}$$

$$NPV = \frac{TN}{TN + FN} \tag{11}$$

$$FNR = \frac{FN}{FN + TP} \tag{12}$$

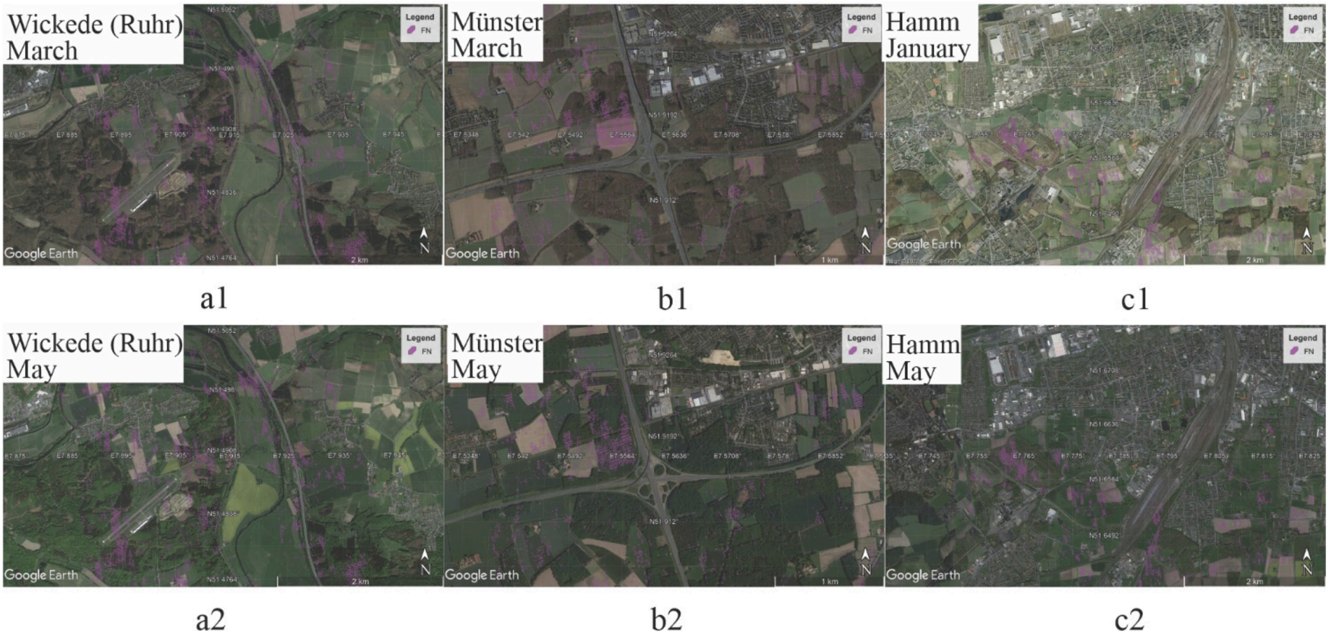


Fig. 11. False Negative points of test regions in North Rhine-Westphalia. (a1) Wickede (Ruhr) taken on March (a2) Wickede (Ruhr) taken on May (b1) Münster taken on March (b2) Münster taken on May. (c1) Hamm taken on January. (c2) Hamm taken on May.

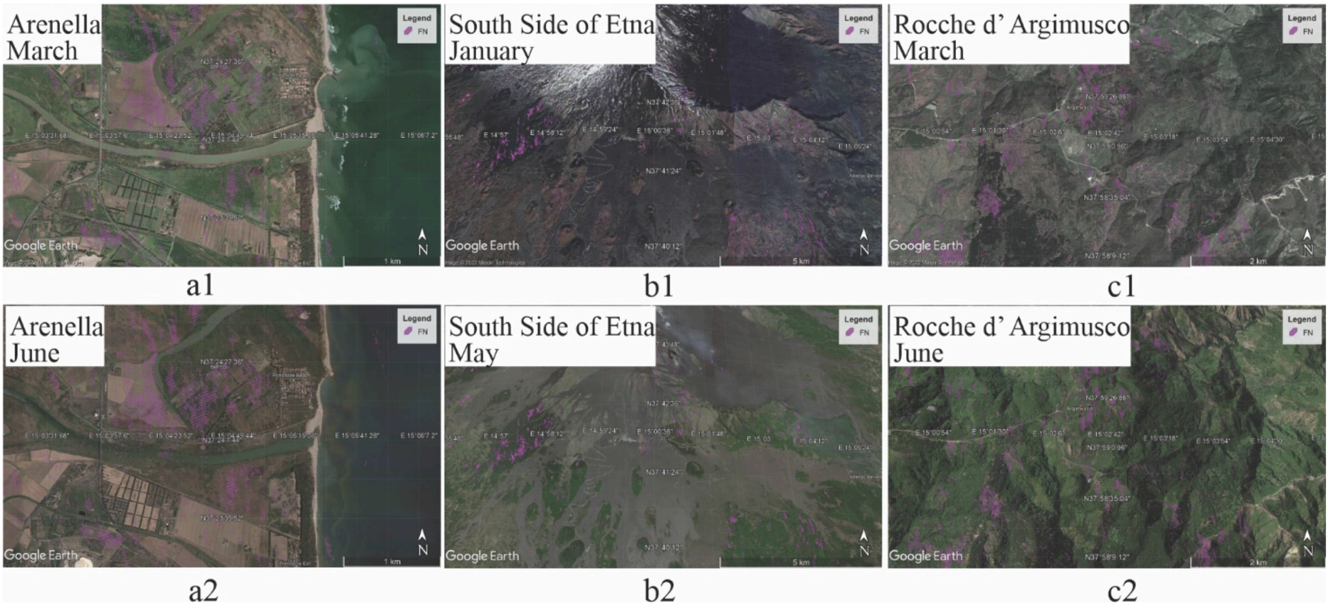


Fig. 12. False Negative points of test regions in Sicily. (a1) Arenella taken on March (a2) Arenella taken on June (b1) South side of Etna taken on January (b2) South side of Etna taken on May. (c1) Rocche d' Argimusco taken on March. (c2) Rocche d' Argimusco taken on June.

Table 7
Percentage of lost coverage caused by FN in test regions.

	Wickede (Ruhr)	Münster	Hamm	Arenella	South side of Etna	Rocche d'Argimusco
LC(%)	3.0 %	2.7 %	0.6 %	0.2 %	0.5 %	0.7 %

Table 8
Complexity comparison: DSPN vs U-Net.

	Parameters	**Training time**	**GPU memory needed**	**Prediction time**
DSPN	68,003,831	49 h	6843.00 MB/batch	56 s
U-Net	51,538,466	27 h	3382.60 MB/batch	43 s

Since our purpose is to create a mask for speeding up DS preprocessing, the type II error (Identify the DSCs as non-DSCs, representing as FN) is more harmful than the type I error (Identify non-DSCs as DSCs, representing as FP). The type I error will only add some unnecessary computation, but the type II error will discard DS points and continue to affect DS processing. The mask without coarse label correction still has some FP points, and those FP points tend to appear as independent noise points rather than a group of scatterers with similar properties. After morphological postprocessing, the remaining FP points

Table 9
Evaluation results of North Rhine-Westphalia with U-Net by using traditional cross-entropy.

Region	TP	TN	FP	FN	Accuracy (%)	Precision (%)	Recall (%)	NPV (%)	FNR (%)
Wickede (Ruhr)	110,849	257,913	31,671	99,567	73.75	77.78	52.68	72.19	47.32
Münster	129,703	244,547	39,692	86,058	74.85	76.57	60.11	73.97	39.89
Hamm	688,498	149,810	54,718	106,974	83.83	92.64	86.55	58.34	13.45

Table 10
Evaluation results of North Rhine-Westphalia with U-Net by using weighted coarse label cross-entropy.

Region	TP	TN	FP	FN	Accuracy (%)	Precision (%)	Recall (%)	NPV (%)	FNR (%)
Wickede (Ruhr)	142,779	234,415	55,169	67,637	75.44	72.13	67.85	77.61	32.14
Münster	153,780	220,077	64,162	61,981	74.77	70.56	71.27	78.03	28.73
Hamm	727,578	124,226	80,302	67,894	85.18	90.06	91.47	64.66	8.54

Table 11
Evaluation results of North Rhine-Westphalia with DSPN by using traditional cross-entropy.

Region	TP	TN	FP	FN	Accuracy (%)	Precision (%)	Recall (%)	NPV (%)	FNR (%)
Wickede (Ruhr)	142,289	295,474	27	62,210	87.55	99.98	69.58	82.61	30.42
Münster	165,169	292,287	84	42,460	91.49	99.95	79.55	87.32	20.45
Hamm	727,201	240,674	121	32,004	96.79	99.98	95.79	88.26	4.22

were corrected in subsequent DS preprocessing. In order to evaluate the impact of those errors, the metric FNR is used for evaluating the loss rate (type II error), it reflects the proportion of missing points in DSC points.

3.3. *Evaluation result of North Rhine-Westphalia*

The evaluation result of three test regions in North Rhine-Westphalia is shown in Table 3. The proposed DSPN achieved a better performance in the urban area (Hamm), which has 98.41 % accuracy and only 2.09 % FNR. The detail of γ_{pt} after DS pre-processing with or without our approach is shown in Figs. 5 to 7. Even though in the rural area or suburban highway (Wickede (Ruhr) and Münster) the accuracy of the proposed approach was lower than in the urban area, our approach still performs well in identifying important objects (airports, rivers, roads, etc.) as shown in Fig. 6b and Fig. 7b.

3.4. *Evaluation result of Sicily*

The evaluation results of the test regions in Sicily shown in Table 4 and Figs. 8-10. After 500 epochs fine tuning training, DSPN also predicted well on the sea – a surface class not appearing in North Rhine-Westphalia (achieved 99.83 % accuracy and 1.41 % FNR in Arenella). For mountain areas with complex terrain, the mask can still retain most of the DSC points (achieved 98.40 % accuracy and 2.23 % FNR in the South side of Etna).

3.5. *Time costing of DSPN*

As the acceleration performance of DSPN varies according to its application scenarios, it is necessary to investigate how faster it boosts. Considering that multi-threading technology is usually used for parallel computing in DS pre-processing, we measure the real-runtime $T_r$and the total computation time T_t(sum of all thread computation time) of DS pre-processing with or without our approach, the result is shown in Table 5. The given times include that for coarse label correction. From the comparison of the six test regions, it can be seen that the more non-DSC the scene has, the faster our approach boosts.

4. Discussion

4.1. *Acceleration efficiency*

Since the real-runtime T_r is affected by the bottleneck of data loading and by the parallel computing scheduling, our method can save about 14.27 % (Hamm, with only 9.12 % non-DSC) to 42.40 % (Arenella, with 83.63 % non-DSC) of the real processing time. Excluding the data loading factor, the main reason for the inconsistency between the actual acceleration effect and the percentage of non-DSC is that the amount of computation in the MATLAB parallel computing mechanism is not evenly distributed in each thread. The real computing time $T_r$depends on the slowest thread. In order to measure the performance improvement by DSPN, the total computation time T_t can reflect the computation amount we saved. By comparing the total computation time T_t, our method can save 8.68 % (Hamm) to 84.10 % (Arenella) of computation costs depending on the scenes Table 6. This shows a good correspondence between the percentage of non-DSC und speed up ratio R_{sp} as defined in formula 13. This also means that if only one thread is used for pre-processing, our approach will yield an efficiency gain close to the percentage of non-DSC.

$$R_{sp} = \frac{T_t(withoutours) - T_t(ours)}{T_t(withoutours)} \tag{13}$$

4.2. *Analysis of the false Negative points*

The study demonstrates that the deep learning approach DSPN can generate a high accuracy mask for predicting the DSCs after training. However, as the DS points cannot be determined only by the γ_{pt}, it is valuable to investigate what those False Negative points are like, and the potential reason that caused the misclassification. From the visual inspection of two optical images taken near the high coherence and low coherence date provided by Google Earth, it can be seen that most of the misclassification happened in the areas covered by deciduous and herbal plants such as fields, forests, street trees, etc. as shown in Figs. 11 and 12. Since the inputs of DSPN are from two acquisitions, these two acquisitions usually correspond to different seasons, spring to summer or autumn to winter. This can lead to the situation that one acquisition

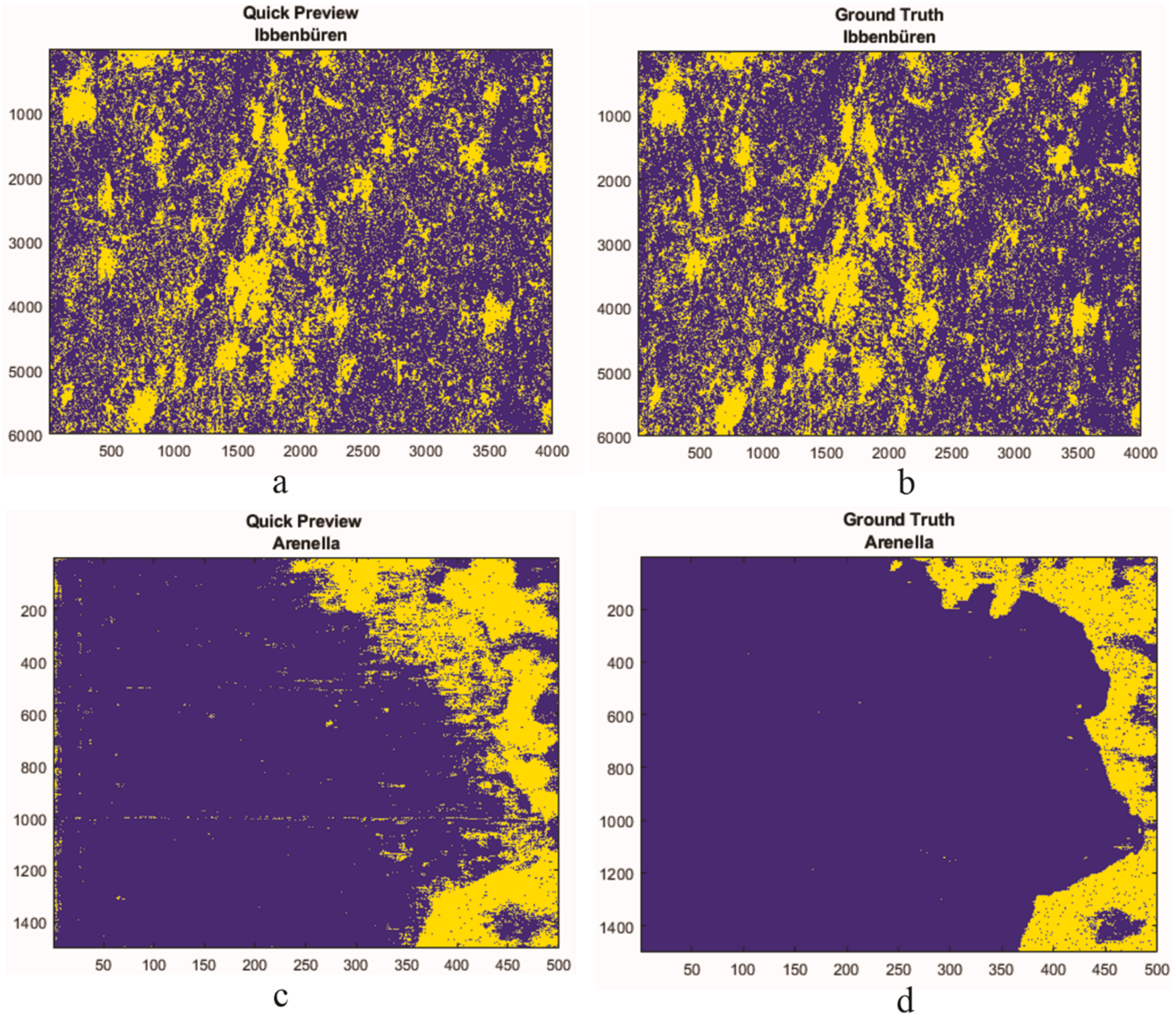


Fig. 13. Comparison between the quick previews and the ground truths. (a) Quick preview of Ibbenbüren by using North Rhine-Westphalia model. (b) Ground Truth of Ibbenbüren. (c) Quick preview of Arenella by using North Rhine-Westphalia model. (d) Ground Truth of Arenella.

yields a better and the other one a worse coherence. The variation of radar reflection patterns of deciduous and herbal plants in different seasons is one of the potential causes of misclassification. Different growth and development stages of plants will change the amplitude and coherence of the polarized radar signal, which makes it possible for DSPN to make confused judgements on the plant coverage area.

After DS pre-processing, the displacement analysis is performed with help of the Stanford Method for PS (StaMPS). Those points found in fields or forests usually are filtered out through the selection step by StaMPS. By using the modified version of StaMPS from (Even et al., 2020) with the predicted mask, the error caused by FN can be evaluated by the percentage of lost coverage LC, which is calculated in the following way: Gridding the scene into approximately 200 m by 200 m cells, and counting the number C_{DS} of cells that contain DS for the result that was obtained without considering a mask and the number C_m of these cells, where DS were missing after a mask was applied. Then the percentage of lost coverage is

$$LC = \frac{C_m}{C_{DS}} \tag{15}$$

The lost coverage for the six test regions is shown in Table 7.

4.3. Network complexity analysis

In order to investigate the complexity of DSPN, we compare for DSPN and U-Net the number of parameters, training time, GPU memory needed, and prediction time on North Rhine-Westphalia. The results are shown in Table 8. In addition, using U-Net as a benchmark, we also investigate the performance improvement brought by the DSPN network structure and weighted coarse label cross-entropy loss function, the result shown in Tables 9–11.

Since the structure of DSPN has multiple decoding paths, it has a better ability for perceiving details. Its improvement in performance mainly comes from the reduction of FP (caused by more accurate perception of details). When the traditional cross entropy is used, each label contributes to the loss with the same weight and the network will try equally hard to separate between different classes of DSC as between any class of DSC and the non-DSC class. By weighting the loss, it will be still sensitive to misclassification between DSC and non-DSC, but much less sensitive to misclassifications between different DSC labels. Presumably, this emphasis of the difference between DSC and non-DSC allows the network to learn separating them cleaner then with the traditional loss function. The gain brought by weighted coarse label cross-entropy is mainly derived from the reduction of FN, that is, the probability of occurrence of Type II errors is reduced.

Compared with U-Net, the DSPN has 24.21 % more parameters, which means it needs more time and memory for training. However, considering the gain in accuracy and that the prediction only needs minutes, we think the cost of adding complexity to the network is acceptable. In addition, the reduction of FP will save time in DS pre-processing stage. In the meanwhile, as the weighted coarse label cross-entropy can be implemented by a Pytorch function, it is supported by GPU parallel computing from CUDA. Using this loss does not significantly increase computation time.

4.4. Generalization ability and future works

This study proposes a faster way of DS pre-processing by using DSPN to generate masks. In addition, DSPN also offers the generation of DSC quick previews as a potential application. If the scene to be processed has a similar land cover as an already trained model, just doing the forward propagation of DSPN allows the user to get a preview of DSC distribution in minutes. Fig. 13 compares the predicted mask of Ibbenbüren in Germany and Arenella in Italy generated by using the North Rhine-Westphalia model and their corresponding ground truth. From the results, the general distribution of DSCs in those two areas can still be identified, although the land cover in Sicily is distinctly different.

While previews can be generated quickly using a trained model from a similar scene, transfer learning is still necessary if high-accuracy masks are needed. In future work, we will study more deeply the generalization ability of DSPN and explore the possibility of general models. To this end, the consideration of additional input layers, e.g. generated from digital elevation models or polarized data might enhance the accuracy and the generalization ability.

5. Conclusion

In this study, we propose the deep learning-based DS pre-processing network DSPN. It constitutes a new approach to accelerate DS pre-processing by telling if a pixel is a DSC or not before DS pre-processing is performed. By only applying DS pre-processing to pixels classified as DSC, DSPN will achieve a boost in speed close to the non-DS rate. Importantly, we saw that this involves no significant loss of coverage. For six different test regions, our approach saved 11.74 % to 87.39 % computation time depending on the scene with an acceptable maximum loss in DS coverage of 3 %.

Currently, the presented approach does not allow obtaining a broadly applicable model that performs well on arbitrary Sentinel-1 data, because the number and character of classes of the training data are scene-specific. Vice versa, a definition of classes that are homogeneous regarding γ_{pt} and is valid for all Sentinel-1 data or at least for those covering certain regions would hypothetically allow to train DSPN such that it is generally applicable. In the future, we intend to develop a more generally applicable version of DSPN with the help of improved training data and input layers. Finally, our approach is capable of generating quick views of DSC distribution if a trained model from an area with similar land cover is available.

Formatting of funding sources

This work was supported by funding of the first author through the China Scholarship Council (CSC).

We acknowledge support by the KIT-Publication Fund of the Karlsruhe Institute of Technology.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Appendix. Supplementary material

Supplementary data to this article can be found online at https://doi.org/10.1016/j.jag.2022.103112.

References

Anantrasirichai, N., Biggs, J., Kelevitz, K., Sadeghi, Z., Wright, T., Thompson, J., Achim, A.M., Bull, D., 2021. Detecting Ground Deformation in the Built Environment Using Sparse Satellite InSAR Data with a Convolutional Neural Network. IEEE Trans. Geosci. Remote Sens. 59 (4), 2940–2950.

Ansari, H., De Zan, F., Bamler, R., 2017. Sequential Estimator: Toward Efficient InSAR Time Series Analysis. IEEE Trans. Geosci. Remote Sens. 55 (10), pp. https://doi.org/10.1109/TGRS.2017.2711037.

Ansari, H., De Zan, F., Bamler, R., 2018. Efficient Phase Estimation for Interferogram Stacks. IEEE Trans. Geosci. Remote Sens. 56 (7), pp. https://doi.org/10.1109/TGRS.2018.2826045.

Ansari, H., De Zan, F., Parizzi, A., 2021. Study of Systematic Bias in Measuring Surface Deformation with SAR Interferometry. IEEE Trans. Geosci. Remote Sens. 59 (2), pp. https://doi.org/10.1109/TGRS.2020.3003421.

Ban, Y., Zhang, P., Nascetti, A., Bevington, A.R., Wulder, M.A., 2020. Near Real-Time Wildfire Progression Monitoring with Sentinel-1 SAR Time Series and Deep Learning. Sci. Rep. 10 (1), 1–15. https://doi.org/10.1038/s41598-019-56967-x.

Caliñski, T., Harabasz, J., 1974. A Dendrite Method Foe Cluster Analysis. Commun. Stat. 3 (1), pp. https://doi.org/10.1080/03610927408827101.

Chen, L., Zhang, P., Xing, J., Li, Z., Xing, X., Yuan, Z., 2020. A multi-scale deep neural network for water detection from SAR images in the mountainous areas. Remote Sens. 12 (19), pp. https://doi.org/10.3390/rs12193205.

Cozzolino, D., Verdoliva, L., Scarpa, G., Poggi, G., 2020. Nonlocal CNN SAR image despeckling. Remote Sens. 12 (6), pp. https://doi.org/10.3390/rs12061006.

Ho Tong Minh, D., Ndikumana, E., Baghdadi, N., Courault, D., Hossard, L., 2018. “Applying deep learning for agricultural classification using multitemporal SAR Sentinel-1 for Camargue, France,” vol. 1078911, no. October 2018, p. 39, 2018, doi: 10.1117/12.2325160.

De Zan, F., Rocca, F., 2005. “Coherent processing of long series of SAR images,” In: International Geoscience and Remote Sensing Symposium (IGARSS), 2005, vol. 3, doi: 10.1109/IGARSS.2005.1526402.

Even, M., Schulz, K., 2018. InSAR deformation analysis with distributed scatterers: A review complemented by new advances. Remote Sens. 10 (5), pp. https://doi.org/10.3390/rs10050744.

Even, M., Westerhaus, M., Simon, V., 2020. Complex Surface Displacements above the Storage Cavern Field at Epe, NW-Germany, Observed by Multi-Temporal SAR-Interferometry. Remote Sens. 12 (20), pp. https://doi.org/10.3390/rs12203348.

Even, M., 2021. “A study on algorithms and parameter settings for ds preprocessing.” In: International Geoscience and Remote Sensing Symposium (IGARSS), 2021, vol. 2021-July, doi: 10.1109/IGARSS47720.2021.9553662.

Even, M., 2022. “A study on algorithms and parameter settings for ds preprocessing part 2.” In: IGARSS 2022 - 2022 IEEE International Geoscience and Remote Sensing Symposium, 2022, pp. 8–11, doi: 10.1109/IGARSS46834.2022.9883345.

Ferretti, A., Fumagalli, A., Novali, F., Prati, C., Rocca, F., Rucci, A., 2011. A new algorithm for processing interferometric data-stacks: SqueeSAR. IEEE Trans. Geosci. Remote Sens. 49 (9), 3460–3470. https://doi.org/10.1109/TGRS.2011.2124465.

Ferretti, A., Fumagalli, A., Novali, F., De Zan, F., Rucci, A., Tebaldini, S., 2011. “Process for filtering interferograms obtained from SAR images acquired on the same area.” CA Patent App. CA 2,767,144, 13.01.2011.

Fornaro, G., Verde, S., Reale, D., Pauciullo, A., 2015. CAESAR: An approach based on covariance matrix decomposition to improve multibaseline-multitemporal interferometric SAR processing. IEEE Trans. Geosci. Remote Sens. 53 (4), pp. https://doi.org/10.1109/TGRS.2014.2352853.

Gaddes, M.E., Hooper, A., Bagnardi, M., 2019. Using Machine Learning to Automatically Detect Volcanic Unrest in a Time Series of Interferograms. J. Geophys. Res. Solid Earth 124 (11), pp. https://doi.org/10.1029/2019JB017519.

Geng, J., Wang, H., Fan, J., Ma, X., 2017. Deep Supervised and Contractive Neural Network for SAR Image Classification. IEEE Trans. Geosci. Remote Sens. 55 (4), pp. https://doi.org/10.1109/TGRS.2016.2645226.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep learning An MIT Press Book 29 (7553), pp.

Guarnieri, A.M., Tebaldini, S., 2007. Hybrid Cramér-Rao bounds for crustal displacement field estimators in SAR interferometry. IEEE Signal Process. Lett. 14 (12), pp. https://doi.org/10.1109/LSP.2007.904705.

Guo, Z., Wu, L., Huang, Y., Guo, Z., Zhao, J., Li, N., 2022. Water-Body Segmentation for SAR Images: Past, Current, and Future. Remote Sens. 14 (7), pp. https://doi.org/10.3390/rs14071752.

He, K., Zhang, X., Ren, S., Sun, J., 2015. “U-net: Convolutional networks for biomedical image segmentation,“ In: International Conference on Medical image computing and computer-assisted intervention," Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), 2015.

He, K., Zhang, X., Ren, S., Sun, J., 2016. "Deep residual learning for image recognition." Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit., vol. 2016-Decem, pp. 770–778, 2016, doi: 10.1109/CVPR.2016.90.

Henry, C., Azimi, S.M., Merkle, N., 2018. Road segmentation in SAR satellite images with deep fully convolutional neural networks. IEEE Geosci. Remote Sens. Lett. 15 (12), pp. https://doi.org/10.1109/LGRS.2018.2864342.

Ho Tong Minh, D., Ienco, D., Gaetano, R., Lalande, N., Ndikumana, E., Osman, F., Maurel, P., 2018. Deep Recurrent Neural Networks for Winter Vegetation Quality Mapping via Multitemporal SAR Sentinel-1. IEEE Geosci. Remote Sens. Lett. 15 (3), 464–468.

Jiang, M., Ding, X., Li, Z., Tian, X., Wang, C., Zhu, W., 2014. InSAR coherence estimation for small data sets and its impact on temporal decorrelation extraction. IEEE Trans. Geosci. Remote Sens. 52 (10), pp. https://doi.org/10.1109/TGRS.2014.2298408.

Jiang, M., Guarnieri, A.M., 2020. Distributed Scatterer Interferometry with the Refinement of Spatiotemporal Coherence. IEEE Trans. Geosci. Remote Sens. 58 (6), 3977–3987. https://doi.org/10.1109/TGRS.2019.2960007.

Jiang, M., Ding, X., Hanssen, R.F., Malhotra, R., Chang, L., 2015. Fast statistically homogeneous pixel selection for covariance matrix estimation for multitemporal InSAR. IEEE Trans. Geosci. Remote Sens. 53 (3), pp. https://doi.org/10.1109/TGRS.2014.2336237.

Jiang, M.i., Yong, B., Tian, X., Malhotra, R., Hu, R., Li, Z., Yu, Z., Zhang, X., 2017. The potential of more accurate InSAR covariance matrix estimation for land cover mapping. ISPRS J. Photogramm. Remote Sens. 126, 120–128.

Keskar, N.S., Socher, R., 2017 [Online].Available: "Improving Generalization Performance by Switching from Adam to SGD" no. 1 http://arxiv.org/abs/1712.07628.

Ketchen, D.J., Shook, C.L., 1996. "The application of cluster analysis in strategic management research: An analysis and critique." Strateg. Manag. J., vol. 17, no. 6, 1996, doi: 10.1002/(sici)1097-0266(199606)17:6<441::aid-smj819>3.0.co;2-g.

Kingma, D.P., Ba, J.L., 2015. "Adam: A method for stochastic optimization." In: 3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc., pp. 1–15, 2015.

Krizhevsky, A., Sutskever, I., Hinton, G.E., 2017. ImageNet classification with deep convolutional neural networks. Commun. ACM 60 (6), pp. https://doi.org/10.1145/3065386.

Lattari, F., Leon, B.G., Asaro, F., Rucci, A., Prati, C., Matteucci, M., 2019. Deep learning for SAR image despeckling. Remote Sens. 11 (13), pp. https://doi.org/10.3390/rs11131532.

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

Lin, K.F., Perissin, D., 2017. Identification of statistically homogeneous pixels based on one-sample test. Remote Sensing 9 (1), pp. https://doi.org/10.3390/rs9010037.

Ma, Y., Li, Y., Zhu, L., 2019. Land cover classification for polarimetric SAR image using convolutional neural network and superpixel. Prog. Electromagn. Res. B 83, 2019. https://doi.org/10.2528/PIERB18112104.

Mayer, T., Poortinga, A., Bhandari, B., Nicolau, A.P., Markert, K., Thwal, N.S., Markert, A., Haag, A., Kilbride, J., Chishtie, F., Wadhwa, A., Clinton, N., Saah, D., 2021. Deep learning approach for Sentinel-1 surface water mapping leveraging Google Earth Engine. ISPRS Open J. Photogramm. Remote Sens. 2, 100005.

Mazza, A., Sica, F., Rizzoli, P., Scarpa, G., 2019. TanDEM-X forest mapping using convolutional neural networks. Remote Sens. 11 (24), pp. https://doi.org/10.3390/rs11242980.

Merchant, M.A., Obadia, M., Brisco, B., Devries, B., Berg, A., 2022. Applying Machine Learning and Time-Series Analysis on Sentinel-1A SAR/InSAR for Characterizing Arctic Tundra Hydro-Ecological Conditions. Remote Sens. 14 (5), pp. https://doi.org/10.3390/rs14051123.

Mestre-Quereda, A., Lopez-Sanchez, J.M., Vicente-Guijalba, F., Jacob, A.W., Engdahl, M. E., 2020. Time-Series of Sentinel-1 Interferometric Coherence and Backscatter for Crop-Type Mapping. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 13, 4070–4084. https://doi.org/10.1109/JSTARS.2020.3008096.

Michaelides, R.J., Zebker, H.A., Zheng, Y., 2019. An Algorithm for Estimating and Correcting Decorrelation Phase from InSAR Data Using Closure Phase Triplets. IEEE Trans. Geosci. Remote Sens. 57 (12), pp. https://doi.org/10.1109/TGRS.2019.2934362.

Milletari, F., Navab, N., Ahmadi, S.A., 2016. "V-Net: Fully convolutional neural networks for volumetric medical image segmentation." In: Proc. - 2016 4th Int. Conf. 3D Vision, 3DV 2016, pp. 565–571, 2016, doi: 10.1109/3DV.2016.79.

Mukherjee, S., Zimmer, A., Sun, X., Ghuman, P., Cheng, I., 2021. An Unsupervised Generative Neural Approach for InSAR Phase Filtering and Coherence Estimation. IEEE Geosci. Remote Sens. Lett. 18 (11), pp. https://doi.org/10.1109/LGRS.2020.3010504.

Nikaein, T., Iannini, L., Molijn, R.A., Lopez-Dekker, P., 2021. On the value of sentinel-1 insar coherence time-series for vegetation classification. Remote Sens. 13 (16), pp. https://doi.org/10.3390/rs13163300.

Odena, A., Dumoulin, V., Olah, C., 2017. Deconvolution and Checkerboard Artifacts. Distill 1 (10), pp. https://doi.org/10.23915/distill.00003.

Parikh, H., Patel, S., Patel, V., 2020. Classification of SAR and PolSAR images using deep learning: a review. Int. J. Image Data Fusion 11 (1), pp. https://doi.org/10.1080/19479832.2019.1655489.

Perissin, D., Wang, T., 2012. Repeat-pass SAR interferometry with partially coherent targets. IEEE Trans. Geosci. Remote Sens. 50 (1), pp. https://doi.org/10.1109/TGRS.2011.2160644.

Pu, L., Zhang, X., Zhou, Z., Shi, J., Wei, S., Zhou, Y., 2020. A phase filtering method with scale recurrent networks for InSAR. Remote Sens. 12 (20), pp. https://doi.org/10.3390/rs12203453.

Rocca, F., Rucci, A., Ferretti, A., Bohane, A., 2013. Advanced InSAR interferometry for reservoir monitoring. First Break 31 (5), pp. https://doi.org/10.3997/1365-2397.31.5.68075.

Rousseeuw, P.J., 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. 20, 53–65.

Rumelhart, D.E., Hintont, G.E., 1986. Learning Representations by Back-Propagating Errors. Nature 323, 533–536. https://doi.org/10.7551/mitpress/1888.003.0013.

Samiei-Esfahany, S., Martins, J.E., Van Leijen, F., Hanssen, R.F., 2016. Phase Estimation for Distributed Scatterers in InSAR Stacks Using Integer Least Squares Estimation. IEEE Trans. Geosci. Remote Sens. 54 (10), pp. https://doi.org/10.1109/TGRS.2016.2566604.

Simonyan, K., Zisserman, A., 2015. "Very deep convolutional networks for large-scale image recognition".

Sun, J., Wauthier, C., Stephens, K., Gervais, M., Cervone, G., La Femina, P., Higgins, M., 2020. Automatic Detection of Volcanic Surface Deformation Using Deep Learning. J. Geophys. Res. Solid Earth 125 (9). https://doi.org/10.1029/2020JB019840.

Sun, X., Zimmer, A., Mukherjee, S., Kottayil, N.K., Ghuman, P., Cheng, I., 2020. DeepInSAR-A deep learning framework for SAR interferometric phase restoration and coherence estimation. Remote Sens. 12 (14), pp. https://doi.org/10.3390/rs12142340.

Szegedy, C., et al., 2015. "Going deeper with convolutions." In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2015, vol. 07-12-June-2015, doi: 10.1109/CVPR.2015.7298594.

Tiwari, A., Narayan, A.B., Dikshit, O., 2020. Deep learning networks for selection of measurement pixels in multi-temporal SAR interferometric processing. ISPRS J. Photogramm. Remote Sens. 166, 169–182.

Wang, Y., Zhu, X.X., 2016. Robust Estimators for Multipass SAR Interferometry. IEEE Trans. Geosci. Remote Sens. 54 (2), pp. https://doi.org/10.1109/TGRS.2015.2471303.

Xiao, F., Tong, L., Luo, S., 2019. A method for road network extraction from high-resolution SAR imagery using direction grouping and curve fitting. Remote Sens. 11 (23), pp. https://doi.org/10.3390/rs11232733.

Xie, H., Wang, S., Liu, K., Lin, S., Hou, B., 2014. "Multilayer feature learning for polarimetric synthetic radar data classification," 2014, doi: 10.1109/IGARSS.2014.6947062.

Yuan, Q. et al., 2020. "Deep learning in environmental remote sensing: Achievements and challenges," Remote Sens. Environ., vol. 241, no. January, p. 111716, 2020, doi: 10.1016/j.rse.2020.111716.

Zhang, Q., Kong, Q., Zhang, C., You, S., Wei, H., Sun, R., Li, L.i., 2019. A new road extraction method using Sentinel-1 SAR images based on the deep fully convolutional neural network. Eur. J. Remote Sens. 52 (1), 572–582.

Zhang, S., Xu, Q., Wang, H., Kang, Y., Li, X., 2022. Automatic Waterline Extraction and Topographic Mapping of Tidal Flats From SAR Images Based on Deep Learning. Geophys. Res. Lett. 49 (2), pp. https://doi.org/10.1029/2021GL096007.

Zhao, C., Li, Z., Tian, B., Zhang, P., Chen, Q., 2019. A Ground Surface Deformation Monitoring InSAR Method Using Improved Distributed Scatterers Phase Estimation. IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 12 (11), pp. https://doi.org/10.1109/JSTARS.2019.2946729.

Zhao, Z., Wu, Z., Zheng, Y.i., Ma, P., 2021. Recurrent neural networks for atmospheric noise removal from InSAR time series with missing values. ISPRS J. Photogramm. Remote Sens. 180, 227–237.

Zheng, Y., Fattahi, H., Agram, P., Simons, M., Rosen, P., 2022. On Closure Phase and Systematic Bias in Multilooked SAR Interferometry. IEEE Trans. Geosci. Remote Sens. 60, 1–11. https://doi.org/10.1109/TGRS.2022.3167648.

Zhou, Z., Rahman Siddiquee, M.M., Tajbakhsh, N., Liang, J., 2018. Unet++: A nested u-net architecture for medical image segmentation. Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics) vol. 11045 LNCS, 3–11. https://doi.org/10.1007/978-3-030-00889-5_1.

Zhu, X.X., et al., 2017. "Deep learning in remote sensing: a review" no. December https://doi.org/10.1109/MGRS.2017.2762307.

Zhu, X.X., Montazeri, S., Ali, M., Hua, Y., Wang, Y., Mou, L., Shi, Y., Xu, F., Bamler, R., 2021. Deep Learning Meets SAR: Concepts, models, pitfalls, and perspectives. IEEE Geosci. Remote Sens. Mag. 9 (4), 143–172.

Zwieback, S., et al., 2016. A Statistical Test of Phase Closure to Detect Influences on DInSAR Deformation Estimates Besides Displacements and Decorrelation Noise: Two Case Studies in High-Latitude Regions. IEEE Trans. Geosci. Remote Sens. 54 (9), pp. https://doi.org/10.1109/TGRS.2016.2569435.

Zwieback, S., Meyer, F.J., 2022. Reliable InSAR Phase History Uncertainty Estimates. IEEE Trans. Geosci. Remote Sens. 60, 1–9. https://doi.org/10.1109/TGRS.2022.3146816.