DEEP LEARNING FOR THE DETECTION OF EARLY SIGNS FOR FOREST DAMAGE BASED ON SATELLITE IMAGERY

Dennis Wittich*[,1], Franz Rottensteiner[1], Mirjana Voelsen[1], Christian Heipke[1], and Sönke Müller[2]

[1]Institute of Photogrammetry and GeoInformation, Leibniz Universiät Hannover – Germany,
(wittich,rottensteiner,voelsen,heipke)@ipi.uni-hannover.de
[2]EFTAS Fernerkundung Technologietransfer GmbH - Germany, soenke.mueller@eftas.com

Commission II, WG II/6

KEY WORDS: Deep Learning, Regression, Forest Monitoring, Label Imbalance, Sentinel-2

ABSTRACT:

We present an approach for detecting early signs for upcoming forest damages by training a Convolutional Neural Network (CNN) for the pixel-wise prediction of the remaining life-time (RLT) of trees in forests based on Sentinel-2 imagery. We focus on a scenario in which reference data are only available for a related task, namely for a bi-temporal pixel-wise classification of forest degradation. This reference is used to train a CNN for the pixel-wise prediction of forest degradation. In this context, we propose a new sub-sampling-based approach for compensating the effects of a heavy class imbalance in the training data. Using the resulting classification model, we predict semi-labels for images of a Sentinel-2 time series, from which training data for a CNN designed to regress the RLT can be derived after some label cleansing. However, due to data gaps in the time series, e.g. caused by clouds, only intervals can be derived for the target variable to be regressed, and for some training pixels one of the interval limits may even be unknown. Consequently, we propose a new loss function for training a CNN for regressing the RLT that only requires the known interval limits. The method is evaluated on a data set in Germany, covering a time-span of 5 years. We show that the proposed sub-sampling strategy for dealing with strong label imbalance when training the classifier significantly reduces the training time compared to other approaches. We further show that our model predicts the RLT with a maximum error of two months for 80% of the forest pixels that die within one year from the acquisition date of the Sentinel-2 image.

1. INTRODUCTION

Deforestation monitoring has become an important topic in the context of limiting the effects of climate change, and it is also relevant from an economic viewpoint (Holzwarth et al., 2020). Local authorities have a need for monitoring forest degradation. On the one hand, they need to detect dead trees or clear-cuts, but it may be even more important to detect early signs for upcoming forest damages to initiate countermeasures in time. One approach to monitor forests for arbitrarily large areas is to make use of satellite imagery and to develop an automated system to predict information related to the forest vitality status, e.g. in the form of a pixel-wise classification or regression of relevant parameters. Today, for both of these types of tasks, Convolutional Neural Networks (CNNs) are considered to be the best-performing methods.

In this paper, our main goal is to train a CNN for the automated detection of early signs for the upcoming forest damage on a pixel-level based on Sentinel-2 imagery. However, it is difficult to generate reference data that can be used in training for such a task. Consequently, we address a scenario in which the training data are generated automatically using existing reference data for a related task, namely for the classification of changes in the vitality state of forests between two epochs for which images are available. In this context, the state changes that are relevant are the transitions from living forest to dead forest and from living or dead forest to clear-cut areas between the epochs at which the two images used for the classification were acquired. As both, dead trees and clear-cuts lead to a very different appearance compared to a healthy forest, training data for this classification task can be determined easily using rule-based methods with minimal human intervention. The resulting reference label maps and an unlabelled time series of the test site form the basis for training a CNN for detecting early signs for upcoming forest damages. For that purpose, we first train a CNN for the bi-temporal classification task and use it to create semi-labels related to changes of the vitality state for each image in the time series. Using the semi-labelled time series, a lower and / or upper limit for the remaining life-time (RLT) can be derived for each forest pixel, i.e. for the remaining time until a forest pixel changes its state from alive to dead or clear-cut. Using these potentially half-open intervals for the RLT, we train a CNN to regress the RLT at pixel-level using a single Sentinel-2 image as input. We expect the model to learn to detect early signs for the upcoming death of trees to solve this regression task, so that a short RLT can be used by local authorities to locate endangered areas and to initiate countermeasures.

To summarize our goals, we aim at training a CNN for detecting early signs for the upcoming death of trees in forests based on reference data for which we only know whether there are still living trees or not.

In the proposed approach, we face two major methodological problems. First, when training the model for classifying vitality changes we have to cope with a strongly over-represented background class consisting of pixels that neither correspond to dead trees nor to clear-cuts in the second image of the considered pair, and pixels which correspond to damages make up a very small percentage of the data only. Such an imbalance is known to cause problems, either in terms of the training time

* Corresponding author

or in terms of the resulting performance of the model for the under-represented classes. To tackle this problem, a new approach is proposed which is based on sub-sampling pixels from the over-represented background class during training. Second, for training the regression model we are only given interval limits instead of crisp reference values for the RLT. Thus, we propose a new loss function to train a CNN for regression using only the known interval limits.

2. RELATED WORK

In this section, first, an overview of recent work addressing the classification of satellite imagery for forest monitoring is given. Afterwards, we discuss related work which exploits time series in the context of semi-supervised learning. We then discuss work that deals with training under class-imbalance. Finally, we review work related to training regression models from interval limits.

Assessing the vitality of forests based on remotely sensed data is not a new topic at all. In the 1970ies the then famous "Waldsterben" was largely addressed using aerial imagery. Later, Zink and Zimmermann (1997) used ERS 1/2 data to predict the vitality of selected forest stands at an instance level. In contrast, we aim at predictions at pixel-level, which is much more informative for local authorities. Some papers for pixel-wise vitality monitoring rely on rule-based techniques. However, such methods usually require expert knowledge, which is not always available. Furthermore, the transferability of such methods to new data may be problematic. For these reasons we focus on approaches based on Machine Learning (ML), which can be transferred easily by providing reference data to be used for training. Creating training data can be a tedious task, but it can at least partly be performed by non-experts. As an example, Tilly et al. (2020) predict forest vitality at pixel-level from WorldView-3 data using conventional ML models such as Support Vector Machines and hand-crafted features, including different vegetation indices. Today, Deep Learning (DL) techniques based on CNN outperform more traditional ML methods in image-related tasks whenever enough training samples are available. An example related to the image-based detection of deforestation is (de Bem et al., 2020). The authors train a CNN to predict deforestation in the Brazilian Amazon region based on Landsat images and show that the CNN outperforms classical ML models such as random forests. Lee et al. (2020) use a CNN to implicitly predict deforestation in South-Korea by performing a pixel-wise classification of land cover, also showing that such models are suitable for this application. It has to be noted that all of the ML approaches mentioned so far are limited to standard cases of supervised learning, and they only detect forest damages after they have occurred. In this paper, we go beyond such approaches by trying to detect early signs for forest damage without using any manually generated training data for that task. We leverage sparse annotations for a related task, the classification of vitality changes, and of time series of Sentinel-2 data to derive the reference values required for training a CNN to solve our target task automatically. To the best of our knowledge, no such approach has been presented so far.

An approach of leveraging unlabelled time series for training a classifier was proposed by Jawed et al. (2020), who apply multi-task learning to simultaneously predict class labels and to predict new data by which a time series is continued. Requiring unlabelled time series only, the latter task is expected to help the CNN to learn a representation that is also meaningful for classification. This is confirmed by experiments in (Jawed et al., 2020), but none of them is based on time series of images. In this work, rather than using the regression task to support the training of a classifier, we focus on directly regressing the RLT at pixel level, conjecturing that early signs for a loss of vitality can be learned by a CNN.

In many RS applications, the class distribution of the training samples is imbalanced. Not considering this factor can lead to long training times and possibly to a degraded classification performance of the under-represented classes. One way to compensate for this imbalance is to use a weighted cross-entropy loss in which pixels that correspond to an under-represented class are considered with a higher weight than pixels of the more frequent classes, e.g. (Bressan et al., n.d.). As such frequency-based weighting approaches are often used in RS applications, we will consider such an approach as baseline in our experiments. Alternative loss functions such as the focal loss for binary classification (Lin et al., 2017) or its variant for the multi-class case (Yang et al., 2019) define the weight of each sample according to the score predicted for the reference class. In this way, the training process should focus on difficult samples. However, in pixel-wise classification, samples that were predicted with a low confidence often correspond to pixels at object boundaries, where the label information is uncertain due to geometric inaccuracies and mixed pixels. Focusing on such pixels could be harmful for the training process. An alternative would be to use the dice loss (Sorensen, 1948; Ren et al., 2020), but this would again cause the classifier to focus on object borders, leading to the potential problems just mentioned.

In this work we propose to counteract class-imbalance by sub-sampling pixels belonging to the over-represented classes to achieve a better balance of the labels during training. This idea is frequently used in the context of assigning a single class label to a sample; see (Rendón et al., 2020) for a recent overview. To the best of our knowledge, it has not been used in the context of pixel-wise classification tasks in combination with CNNs yet.

The application of DL for the pixel-wise regression of a target variable is not new and has been used in a wide range of applications such as predicting grey values of images (Cavallari et al., 2018), depth (Liebel and Körner, 2019; Eigen et al., 2014) or disparities (Kang et al., 2020). All of these works have in common that reference values for the target variable are available, which allows for supervised training by minimizing the discrepancy between the pixel-wise predictions and the reference values. However, in applications such as the one dealt with in this paper, the exact reference values may be unknown; instead, intervals containing the correct values may be available. This could be due to data gaps in the time series, but there could also be other reasons, e.g. uncertainties of the reference labels. To the best of our knowledge, our paper is the first one to propose a solution for such a learning scenario.

Considering this review of the related work, we can summarize our scientific contributions as follows:

1. We present a method to train a CNN for detecting early signs for forest damages which only requires a reference for forest damages that have already occurred, which is easier to generate. Our method uses a classification model to automatically generate training data for the target task. For that purpose, an unlabelled time series of Sentinel-2 images is used.

2. The actual target task of our method is formulated as a regression problem. In this context we present a new loss function to train such a regression model based on potentially half-open intervals.

3. As a minor contribution, we present a new approach to deal with strong label imbalance for training a classifier which is based on sub-sampling the over-represented classes.

3. METHODOLOGY

The main goal of our method is to train a CNN R that detects early signs for forest damage at pixel-level in an input image x_r. To achieve this goal, we propose to formulate a regression problem: for every pixel $x_{r,i}$ in x_r, the CNN R should predict the remaining time $\hat{r}_{r,i}$ until a damage is expected to occur. We refer to the target variable as *remaining life-time* (RLT) and to this task as *damage forecasting*. Our method requires a reference Y for the related task of predicting forest damage from image pairs and the availability of an unlabelled time series $[x_0, ..., x_j, ..., x_J]$ of RS images of a region of interest (ROI) with pixel-wise information about cloud cover. Whereas in principle, our method can be applied to images acquired by any RS sensor with sufficient spatial resolution and sufficiently high revisit times, the experimental evaluation is restricted to Sentinel-2 imagery in this paper (cf. section 4).

The required reference Y consists of image-pairs (x_0, x_1) of a ROI acquired at different epochs in time for which it is known which pixels correspond to living trees in x_0 and to dead trees in x_1 (i.e., which pixels show trees that have died in the time interval between the acquisition of the two images) and which pixels correspond to clear-cuts (i.e. which pixels show living or dead trees in x_0 and no trees in x_1). We refer to both cases, the death of trees and clear-cuts, as *forest damage*. The reference for such a classification of forest damage can be created with minimal human effort based on image pairs (cf. section 4.1).

The available reference Y can be used to automatically generate training samples for damage forecasting. However, the required reference values for training the regression network cannot be derived directly from Y for two reasons. Firstly, the actual date when a damage occurred is unknown, because for every reference image pair (x_0, x_1) we only know whether or not a damage occurred between the corresponding acquisition dates. Secondly, even if we knew the exact date when each damage occurred, training the regression network would require many image pairs corresponding to a large variety of different RLT values.

Thus, we first train a model C for the bi-temporal damage classification (cf. section 3.2) and apply it to label each image x_j of the unlabelled time series except x_0, using every image pair $[x_0, x_j]$ with $J \geq j > 0$ as input for C. From the semi-labelled time series, an interval enclosing the date when a damage occurred can be derived for every pixel of the ROI: its lower limit corresponds to the last date when the pixel is predicted to show living forest, whereas the upper limit corresponds to the earliest date when a pixel shows either dead trees or a clear-cut. By subtracting the acquisition date of an image in the time series from these limits, we obtain a reference Y_R consisting of the lower and upper limits (b_l and b_u, respectively) of the interval enclosing the RLT of every pixel of that image. The length of these intervals varies between a few days and several months, depending on the availability of data and cloud cover. In some cases the interval limits are partially or fully unknown. For example, if no damage occurred before the end of the time series, no upper limit can be derived (cf. section 3.3). In the last step, R is trained using the known interval limits by minimizing a proposed new loss function (cf. section 3.4). Figure 1 gives an overview of the method.

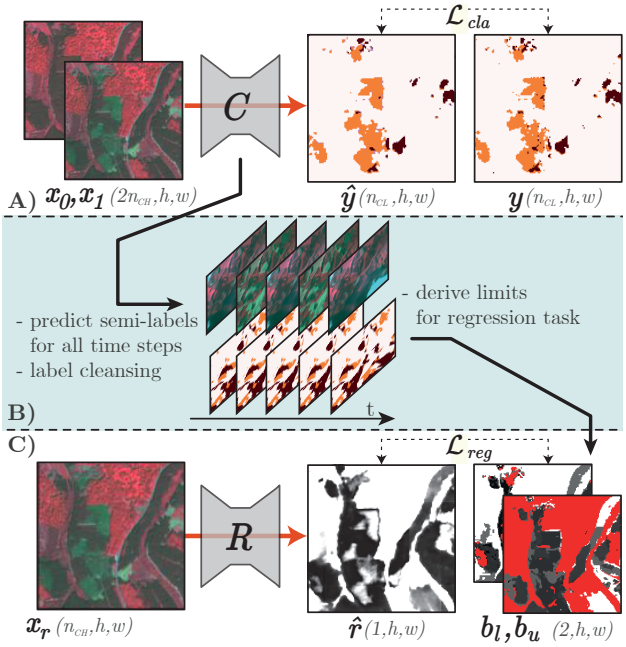

Figure 1. Method overview. A) A CNN C is trained to predict forest damage from an image pair $[x_0, x_1]$, using reference labels y. B) C is used to predict semi-labels for a time series of unlabelled images; they are used to derive the interval limits b_l, b_u for the regression task. C) A CNN R is trained to predict the RLT ($\hat{r}$) for each pixel in an input image x_r. Colour codes for $\hat{y}$ and y: orange - death of trees, brown - clear-cuts, white - background. Colour-codes for $\hat{r}$, b_l and b_u: grey-values - RLT (darker . . . shorter time), red - interval limit unknown.

3.1 Network architecture

The architectures for C and R are identical except for the number of input channels and the design of the output layers. While C takes a stack x_c of two images with n_{CH} channels each as input, the input of R consists of a single image x_r having n_{CH} channels. The architecture is based on a fully convolutional encoder-decoder network with skip connections, following the main design choices of U-Net (Ronneberger et al., 2015). Similarly to (Wittich and Rottensteiner, 2021) we use the Xception ResNet (Chollet, 2017) as encoder, but we reduce the depth of the encoder so that a smaller number of parameters is required. This design choice is motivated by preliminary experiments in which we observed this variant to result in a faster convergence in training without any noticeable reduction in the classification performance. The output layer of C is designed for pixel-wise classification and consists of a single convolutional layer with softmax activation, which predicts probabilities $\hat{y}_{c,i}$ for every pixel i to correspond to class c. The output layer of R solves a pixel-wise regression task. It consists of a convolutional layer with the output $\hat{r}$; no non-linearity is applied. The network architecture is described in Table 1. The layers marked with -C or -R are only used in the models C and R, respectively. Details concerning the structure of the Xception blocks can be found in (Chollet, 2017). Both networks have about $15.5M$ parameters.

	Layer(s)	Layer type	h, w	Depth
Encoder	1-C	Input layer for C	256	$2n_{CH}$
	1-R	Input layer for R	256	n_{CH}
	2	Conv(3) stride 2, BN, ReLU	128	32
	3	Conv(3), BN, ReLU	128	64
	4	Xception block	64	128
	5	Xception block	32	256
	6-15	Xception block	16	728
Decoder	16	Upsample, Concat(5)	32	512
	17, 18	Conv(3), ReLU	32	128
	19	Upsample, Concat(4)	64	256
	20, 21	Conv(3), ReLU	64	64
	22	Upsample, Concat(3)	128	128
	23, 24	Conv(3), ReLU	128	32
	25	Upsample	256	32
	26, 27	Conv(3), ReLU	256	16
	28-C	Conv(1), Softmax	256	n_{CL}
	28-R	Conv(1), Linear	256	1

Table 1. Layers of the architectures of C and R. Both models are based on the same architecture, but have different input and output layers. Conv(s): convolution with kernel size $s \times s$; BN: Batch-Normalization; ReLU: rectified linear unit. *Concat(X)*: depth-wise concatenation of the output of layer *X* and the current layer. Depth, h, w: output dimensions.

3.2 Training with imbalanced class distribution

A common way of dealing with class imbalance in the training data is to give each sample a weight based on its reference label (Bressan et al., n.d.), leading to the weighted categorical cross entropy loss (WCE):

$$\mathcal{L}_{cla} = -\frac{1}{\sum_i \sum_c t_{c,i}\, w_i} \sum_{i=0}^{n_P} \sum_{c=0}^{n_{CL}} t_{c,i}\, w_i\, log(\hat{y}_{c,i}). \quad (1)$$

In equation 1, $n_P = n_B \cdot h \cdot w$ is the number of pixels in a batch, where n_B, w and h are the number of images in each batch, the width and the height of each input image, respectively, n_{CL} denotes the number of classes, and $t_{c,i}$ is an indicator variable with $t_{c,i} = 1$ if the reference for pixel i is c and 0 otherwise. The predicted probability for pixel i to belong to class c is denoted by $\hat{y}_{c,i}$, while w_i corresponds to the weight of that pixel.

In this paper, we consider the case in which there is only one strongly over-represented *background class* (BG) and a set DC of n_{DC} under-represented *damage classes* $DC_f \in DC = \{DC_1, ..., D_{n_{DC}}\}$. Instead of using class-specific weights in equation 1 (Bressan et al., n.d.), we select a sub-sample of the available training pixels to compensate for the class imbalance. Formally, this is achieved by using weights $w_i = 1$ for pixels to be used for training and $w_i = 0$ for excluded pixels, so that minimizing the WCE loss in equation 1 is equivalent to minimizing the standard categorical cross-entropy (CE) loss applied to the samples to be considered in training. This subset contains all samples belonging to the damage classes and all BG samples of the current mini-batch that are erroneously assigned to a damage class; we consider the latter to contain relevant information for the training process. Furthermore, we also select some BG samples that are classified correctly in the current iteration for training. To do so, we set the weight of a pixel that was correctly classified as BG to 1 with a probability of $1 - p_{BG,b}$, while the weights of all other such BG pixels are set to 0; $p_{BG,b}$ is computed based on the label distribution in each batch b according to

$$p_{BG,b} = 1 - \frac{\sum_{f=0}^{n_{DC}} o_{DC_f,b}}{n_{DC} \cdot o_{BG,b}}, \quad (2)$$

where $o_{DC_f,b}$ and $o_{BG,b}$ denote the number of pixels in batch b belonging to the classes DC_f and BG, respectively. In this way, the number of background pixels effectively used in each batch roughly corresponds to the average number of pixels of the damage classes. Using this sub-sampling strategy, we define the binary weights $w_i \in \{0, 1\}$ and minimise the WCE loss in equation 1 using a variant of stochastic mini-batch gradient descent (SGD) to obtain the classification model C.

3.3 Exploiting semi-labelled time series

The goal of the second step is to obtain the intervals for the RLT using the trained model C and an unlabelled time series $T_0 = [x_0, ..., x_j, ..., x_J]$ with J images of a ROI. For each image x_j in T we also need the information on cloud cover in the form of a binary image cc_j in which $cc_{j,i} = 1$ if pixel i in image x_j is affected by cloud cover and $cc_{j,i} = 0$ otherwise.

We start by using C to predict a label map y_j for each image x_j in the time series $T = [x_1, ..., x_j, ..., x_J]$ (i.e., for all images in T_0 except x_0) using the image pair $[x_0, x_j]$ as input for C. As the predicted label maps are likely to contain errors, we perform label cleansing to increase the quality of the semi-labels. For each pixel $x_{j,i}$ in each image x_j in T we construct a set $S_{j,i}$ that contains the semi-label $y_{j,i}$ for $x_{j,i}$, the semi-labels of the four direct spatial neighbours of $x_{j,i}$ as well as the labels $y_{j+1,i}$ and $y_{j-1,i}$ of the temporal neighbours $x_{j+1,i}$ and $x_{j-1,i}$, respectively. Non existing neighbours are ignored at this point. After that, for each pixel, the initial semi-label $y_{j,i}$ is replaced by the most frequent label in $S_{j,i}$.

Based on the resulting cleansed semi-labels, the pixel-wise interval limits for the regression variable are derived for each image x_j in T. To that end, the label series $[y_{1,i}, ..., y_{j,i}, ..., y_{J,i}]$ for each pixel i is analysed. First, we search for the index l_i corresponding to the latest date at which pixel i belongs to the BG class. That is, l_i corresponds to the largest index r for which $(y_{r,i} = BG) \wedge (y_{\hat{r},i} \notin DC\ \forall\ \hat{r} < r)$. After that, we search for the index u_i corresponding to the first date at which the pixel is predicted to show damaged trees, considering that some epochs may have to be ignored due to cloud-cover. Thus, u_i corresponds to the smallest index s for which $(y_{s,i} \in DC) \wedge (cc_{s,i} \neq 1) \wedge (y_{\hat{s},i} \neq BG\ \forall\ \hat{s} > s)$. The lower and upper interval limits for the target variable of the regression, $b_{l,j,i}$ and $b_{u,j,i}$, respectively, for the i-th pixel $x_{j,i}$ in image x_j are obtained by subtracting the capturing date of image x_j from the capturing dates of the images x_{l_i} and x_{u_i}, respectively.

For some pixels, one or both of the limits may be invalid. In all such cases, invalid limits are marked as $unknown$. For instance, if the complete label series of a pixel belongs to the class BG, the lower limit will correspond to the last date of the time series and the upper limit will be $unknown$. If the label sequence for a pixel starts with a damage class, the lower limit will be $unknown$. Further, if the damage has occurred before the capturing date of image x_j, i.e. if any of the two limits is negative, both of them will be marked as $unknown$. We do not differentiate between forest and non-forest areas, as no reference is assumed to be available to differentiate between these classes.

In the reference for the damage classes, non-forest areas are assigned to the BG class in all images. Consequently, the lower limits will always correspond to the end of the time series and upper limits will be marked as $unknown$. Thus, the regression model should learn to predict high RTL values for non-forest areas. An example for the interval limit maps is shown in Figure 1.

3.4 Regression for damage forecasting with intervals

Having derived the lower and upper interval limits $b_{l,j,i}$ and $b_{u,j,i}$ for each pixel $x_{j,i}$ in each image x_j in T, we train a regression model to predict the RLT $\hat{r}_{j,i}$. As no reference values exist for the RLT, the loss used for training cannot be based on the differences of the predictions and the reference values. Instead, we propose to formulate the loss as the squared error for predictions that are not within the known interval limits:

$$\mathcal{L}_{reg} = \frac{1}{n_{LB}} \sum_{m=1}^{n_{LB}} e_m^2 + \frac{1}{n_{UB}} \sum_{n=1}^{n_{UB}} e_n^2, \tag{3}$$

where n_{LB} and n_{UB} are the numbers of pixels for which $b_{l,j,i}$ and $b_{u,j,i}$, respectively, are known, and

$$e_m = \begin{cases} \hat{r}_{j,m} - b_{l,j,m}, & \text{if } \hat{r}_{j,m} > b_{l,j,m} \\ 0, & \text{otherwise} \end{cases}$$
$$e_n = \begin{cases} b_{u,j,n} - \hat{r}_{j,n}, & \text{if } \hat{r}_{j,n} < b_{u,j,n} \\ 0, & \text{otherwise.} \end{cases}$$

The indices m and n refer to the sets of pixels for which the lower and upper limits are known, respectively. We chose to consider the average loss for pixels with known upper and lower limits with equal contribution to counteract the fact that in real scenarios, the number of pixels with a known lower limit is much larger than the one with a known upper limit, which could lead to a bias of the model towards predicting large RLT values. The model R is trained by minimizing $\mathcal{L}_{reg}$, again using SGD.

4. EXPERIMENTS

4.1 Dataset description

We evaluate the proposed methods in a real application, addressing the monitoring of forests in North-Rhine-Westphalia, Germany. In cooperation with the forest agency *Landesbetrieb Wald und Holz NRW* we defined the class structure {Dead trees (DT), Clear-cuts (CC), and Background (BG)}. The classes describe the vitality changes in an image pair (cf. section 3). The BG class contains both living trees and pixels not corresponding to forests. It is strongly over-represented, covering about 99% of the training data. The set of under-represented damage classes is $DC = \{DT, CC\}$, thus $n_{DC} = 2$.

The data set consists of 214 Sentinel-2 images from 2017/06 to 2021/09, spread across four Sentinel-2 tiles with an side length of $109.8\,km$ each. Table 2 lists the tile identifiers and presents the number of images available per tile and year. Figure 2 gives an overview of the data set.

The images were selected manually, picking images with an acceptable cloud coverage (usually less than 5%). We used the atmospherically corrected Level-2A products which are

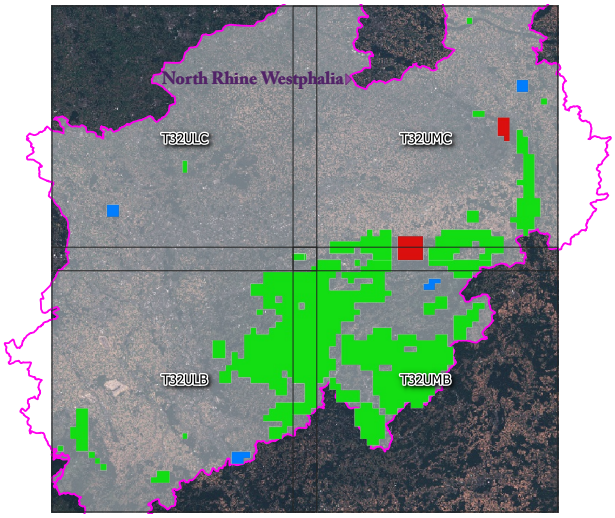

Figure 2. Overview of the test data. For each of the four tiles a time series of about 52 images is available (the exact number of images per tile varies due to varying cloud coverage). Training areas with at least 5% of the pixels belonging to a class in DC are shown in green, validation areas in blue and test areas in red.

	Sentinel-2 tile-ID			
Year	T32ULB	T32ULC	T32UMB	T32UMC
2017	5	5	6	5
2018	16	20	16	22
2019	12	10	10	12
2020	14	14	13	16
2021	5	4	4	5

Table 2. Number of images per year and Sentinel-2 tile.

provided by the platform `CODE-DE`[1]. Although more spectral bands are available, we only used the channels `Near Infrared (NIR)`,`RED`,`GREEN` with a spatial resolution of $10\,m$ and a radiometric resolution of 12 bits per pixel and channel. We further considered the cloud maps for each Sentinel-2 image which are provided with the Level-2A products. In all experiments, we ignored pixels corresponding to clouds according to these cloud maps both in training and in the evaluation.

As initial reference data for the classification task, label maps for 7 image pairs were available, provided by the company *EFTAS Fernerkundung Technologietransfer GmbH*. This reference, denoted by $Y_{RuleBased}$, was generated in a semi-automated way based on Sentinel-2 imagery: First, the decrease of the NDVI (normalized difference vegetation index) between the earlier and the later image in the image pair was compared against a threshold in order to separate the BG class from the damage classes. The threshold was set based on visual interpretation of the classification results. In a second stage, the pixels identified as belonging to one of the damage classes were further separated into DT and CC by comparing the `BLUE` band against a fixed threshold, which was tuned based on reference data for a small area that was manually annotated by experts. Lastly, a forest mask from former projects of the company was used to set the labels of non-forest pixels to the BG class. Table 3 gives an overview over the reference maps and the corresponding label distributions.

In order to reduce the overall amount of data and to increase the initial ratio of pixels in DC, all available Sentinel-2 tiles and

[1] `https://code-de.org/`

ID	Acquisition date of x_0	Acquisition date of x_1	Label distribution [%] BG	DT	CC
R1	2017/06/19	2018/09/27	99.8	0.1	0.1
R2	2017/06/19	2019/06/27	99.8	0.1	0.1
R3	2017/06/19	2019/08/23	99.7	0.1	0.2
R4	2017/06/19	2020/06/01	99.1	0.3	0.6
R5	2017/06/19	2020/09/19	98.8	0.5	0.6
R6	2020/03/23	2021/03/30	98.5	0.5	0.9
R7	2017/06/19	2021/06/13	98.4	0.3	1.3

Table 3. Dates and label distributions of the reference label maps for the bi-temporal forest degradation classification. The reference maps are available for all four Sentinel-2 tiles shown in Figure 2.

the corresponding label maps were split into regular sub-tiles having a size of $512 \times 512\,px$, of which only those containing at least $r_{min} = 5\%$ of pixels in DC were kept. Adjacent sub-tiles overlap by $256\,px$ such that the actual training patches with a size of $256 \times 256\,px$ can be sampled from every position. Compared to using the full data set, using $r_{min} = 5\,\%$ increases the effective ratio of pixels in DC from $1.1\,\%$ to $9.4\,\%$ while decreasing the overall number of BG pixels from $26.4\,M$ to $16.5\,M$. Using this strategy, we obtained a pre-sampled dataset consisting of 856 sub-tiles, each having at least one and a maximum of 7 reference maps for the classification task, depending of the corresponding fraction of DC pixels. Out of these sub-tiles, 17 were used for validation and 24 were reserved to be used as our test set.

As the initial reference label maps $Y_{RuleBased}$ were derived in a semi-automated procedure, they cannot be considered to be error free. To better asses the quality of the predictions of the classification model, we manually generated additional label maps Y_{Manual} for one image-pair of the test area, corresponding to the date of R4 in Table 3. As it is difficult to perfectly label the class transitions, we marked pixels in Y_{Manual} that have at least one neighbour with a different class as unknown and ignored these pixels during the evaluation. In the experiments this reference is solely used in section 4.4.2. For the task of regressing the RLT, we use the cleansed semi-labels for the time series as a reference for training and testing, relying on the same spatial split of the data into areas for training, validation and testing as for the classification task. Whereas we use all time-steps for training and validation, only the images from 2020/06/01 are used for testing.

4.2 Quality metrics

For the classification task we report the $F1$ score for each class, i.e. the harmonic mean of precision and recall, and the mean $F1$ score $mF1$ over all classes. We use this metric because it is not biased towards the performance of over-represented classes.

In order to assess the performance of the models for damage forecasting, we compute the average errors AE_{low} and AE_{up} for all pixels m and n with known lower and upper limits, respectively:

$$AE_{low} = \frac{1}{n_{UB}} \sum_{m=0}^{n_{UB}} e_m \quad , \quad AE_{up} = \frac{1}{n_{LB}} \sum_{n=0}^{n_{LB}} e_n, \qquad (4)$$

where the variables are the ones described in the context of equation 3. To summarize the performance in a single metric, we introduce the balanced average error $bAE = (AE_{low} + AE_{up})/2$. We further report the error rates ER_{low} and ER_{up} that indicate how often predictions exceed the lower and upper limits, respectively.

Finally, we report the interval error rate $ER_{int,p}$ that indicates how often a prediction was either inside the interval limits or less than p months off. The last metric is computed only for those pixels for which both interval limits are known.

4.3 Experimental setup

We divide our experiments into two groups. First, we focus on training C for the pixel-wise classification and evaluate the proposed strategy for dealing with strong class imbalance. In this context we compare our method to several recent approaches from the literature. These experiments are reported in section 4.4.1. Using the model C we then predict semi-labels for each image in the unlabelled time series. Instead of using only the very first image of the time series (cf. section 3.3) as first image x_0 in the bi-temporal image pair, we use all images from 2017 as first images and all images after 2017 as later images x_1. This leads to redundant predictions for each image, which are fused by using a maximum voting strategy. The reason why we use all images from 2017 as first images is that there is barely any change in the vitality during that year. Thus, the resulting redundancy should improve the quality of the semi-labels. Afterwards, label cleansing is performed and the intervals for the regression are derived as described in section 3.3. We use the manually generated reference Y_{Manual} to assess the performance of the cleansing approach, the quality of the initial label maps $Y_{RuleBased}$ and the quality of the predictions of C. These experiments are reported in section 4.4.2.

Finally, in section 4.5 we evaluate the proposed strategy for training the forecasting model R using the interval limits derived from the cleansed semi-labelled time series.

4.4 Evaluation of bi-temporal classification

4.4.1 Training with imbalanced label distribution: Using the data described in Section 4.1, we train several models for the bi-temporal classification and compare the proposed method to different variants for dealing with imbalanced label distributions. The weights of the encoders are initialized from a network that was pre-trained for pixel-wise classification of land-cover in a different area of Germany, again based on Sentinel-2 data. The weights of the decoders and the classification head are randomly initialized according to (He et al., 2015). In preliminary experiments, we found this to lead to a better classification performance compared to using all weights from the pre-trained model. For each variant, three models are trained, each time starting from a different random initialization of the layers that are not pre-trained and using a different random order for the batch generation to assess the influence of these random components.

In the proposed variant V_{sampld} we minimize $\mathcal{L}_{cla}$ using the proposed sub-sampling strategy as described in section 3.2. Based on this variant, the following hyper-parameters were tuned by optimizing the $mF1$ score on the validation set. We use SGD with a batch size of 32, a learning rate of 0.01 and a momentum of 0.9 as optimizer and apply weight decay with a factor of 10^{-5}. For data augmentation, the training images were randomly cropped from the sub-tiles (cf. section 4.1), randomly flipped and rotated in 90-degree steps. We compare the proposed method to several variants, using the same hyper-parameters.

In $V_{ce,full}$ and V_{ce} the regular CE loss is minimized. While in V_{ce} the dataset is pre-sampled as described in section 4.1, in $V_{ce,full}$ the complete dataset is used, i.e. r_{min} is set to 0%, which leads to an even stronger class imbalance, because many patches contain BG pixels only.

We also compare to V_{weight}, in which we use the class-frequency-based weighting approach by Bressan et al. (n.d.).

Furthermore, we compare to variants V_{focal} and V_{dice} in which the focal loss and the dice loss are minimized, respectively. When minimizing the dice loss, the learning rate is decreased to 0.001 as the initial learning rate of 0.01 leads to a divergence of the training process. All models are evaluated on the validation set every 2500 update steps; we refer to one such set of update steps as one epoch. Training is stopped when the $mF1$ score on the validation set no longer increases for 10 epochs and the models achieving the respectively best validation scores are kept. These models are used to predict labels for the test set. The results are summarized in table 4. In this table, *best epoch* refers to the epoch that is chosen using the best score on the validation set.

			Class $F1$ [%]		
Variant	$mF1$ [%]	best epoch	BG	DT	CC
$V_{ce,full}$	$\mathbf{94.4} \pm .0$	905 ± 45	$\mathbf{99.3}$	92.4	$\mathbf{91.5}$
V_{ce}	$94.1 \pm .3$	361 ± 14	99.2	92.8	90.4
V_{sampld}	$94.2 \pm .2$	$\mathbf{164} \pm 20$	99.2	$\mathbf{92.9}$	90.7
V_{focal}	$94.1 \pm .1$	760 ± 18	99.1	92.7	90.3
V_{dice}	$94.0 \pm .2$	483 ± 4	99.1	$\mathbf{92.9}$	89.9
V_{weight}	$92.7 \pm .4$	365 ± 77	98.8	91.0	88.4

Table 4. Performance of different variants of the classification model on the test set. V_{sampld} refers to the proposed variant. Best epochs and $mF1$ scores are reported as means and standard deviations over three runs. For the class-wise metrics the means are given. The best results per column are printed in bold font.

It can be observed that the resulting performance is quite similar for all models. Considering the $mF1$ score and a significance level of 0.05, the only statistically significant difference of the proposed method V_{sampld} to the others is the one to V_{weight}, where V_{sampld} is significantly better. It can also be observed that the proposed variant converged much faster, at least by a factor of 2 compared to the other variants. The best performance is achieved when training with standard CE loss using all data, but this comes at the cost of a drastically increased training time. Training C using variant $V_{ce,full}$ took about 100 hours on a GeForce Titan XP consumer GPU while training the CNN using V_{sampld} took only about 17 hours using the same hardware. At a first glance it might be contradictory that training without any measures to counteract the effects of a strong class imbalance in the training data leads to better results than using one of the methods specifically designed to mitigate that problem. A potential reason for this behaviour could be a very different appearance of the classes, which might lead to very well defined clusters with low overlap in feature space. In such a situation, minimizing the CE loss converges to a good solution, though at the cost of a comparably long training time.

4.4.2 Evaluation of label cleansing: Next, a model trained using V_{sampld} is used to predict semi-labels for the complete time series and label cleansing is performed. We extract the cleansed semi-labels for the test areas, which we denote by $Y_{SemLabels}$. In order to assess the performance of the generation and cleansing of the semi-labels, we compare $Y_{SemLabels}$ to the manually generated reference maps Y_{Manual}. In this context we also compare the initially available test-set reference $Y_{RuleBased,T}$ for the bi-temporal damage classification task, i.e. the one that was created semi-automatically, and the predictions of the models trained using V_{sampld} to Y_{Manual}. The results are presented in Table 5.

		Class $F1$ [%]		
Variant	$mF1$ [%] $\uparrow$	BG	DT	CC
$Y_{RuleBased,T}$	92.5	99.4	89.1	89.0
V_{sampld}	92.6 ± 0.3	99.4	89.4	88.9
$Y_{SemLabels}$	95.3	99.7	93.3	92.9

Table 5. Agreement of different label maps for the test area with the manually generated reference Y_{Manual}.

The $mF1$ score of 92.5 % of the initial labels $Y_{RuleBased,T}$ seems understandable, because they were created in a semi-automated procedure and are not error-free. Training a model on these labels using the proposed method leads to comparable results. This is to be expected, because in regular supervised training the quality of the training data can be considered as upper bound for the performance of the models trained on this data set. We conclude that the model can be trained very well on these data and the performance cannot be significantly improved without also improving the training data set.

However, the cleansed semi-labels $Y_{SemLabels}$ have a better agreement with the manual reference than the initial reference $Y_{RuleBased,T}$ (2.8 % in the $mF1$ score). This confirms our assumption that the proposed strategy for labelling the time series and the successive label cleansing step increases the quality of the reference.

4.5 Training the regression model

Using the interval limits derived from the semi-labelled time series, the CNNs for regressing the RLT were trained by minimizing $\mathcal{L}_{reg}$ using SGD with momentum. The hyper-parameters for training are the same as those in the training for the damage classification, except for the learning rate, which is reduced to 0.001. The models are initialized using the parameters of the classification models C. As the input layer of R has fewer channels, the weights related to the earlier image in C are dropped. The output layer of R is randomly initialized. We repeat training five times for each variant, in each case starting from a different random initialization of the regression head and using a different random order for the batch generation.

We propose not to use batch-normalization for training the regression models, as we assume that a modification of the features based on the respective features of other images in each batch could lead to problems in the regression task. Instead we use the running averages obtained in the training of C. For early stopping and model selection, the bAE on the validation set is considered.

Besides the proposed variant V_R, trained as described in section 3.4, we also evaluate a variant $V_{R,BN}$ in which we use standard batch-normalization to validate the above assumption. The results are presented in Table 6. Note that for the test set a lower interval limit was available for 90.2% of the pixels while the upper limit was available for 7.4% only.

Variant	bAE	AE_{low} [days]	AE_{up}	ER_{low} [%]	ER_{up}
$V_{R,BN}$	31.3 ± 0.9	13.8	48.7	11	43
V_R	25.8 ± 0.9	16.3	35.3	13	35

Table 6. Performance of the regression models on the test set. For bAE, we report means and standard deviations over five runs. For the remaining metrics the means are reported only.

We observe that not using batch-normalization for training the regression models yields significantly better results with respect to a confidence level of 0.05, which confirms our assumption that batch-normalization can be harmful for training regression models. It can be seen that both, the error rate and the average errors are higher for the upper limits, i.e. the model tends to overestimate the remaining life-time. This seems understandable, because sometimes forest areas are felled without a preceding infestation. In such a case it is not possible to make a correct prediction, because the images will not show any signs for an upcoming deforestation.

To further assess the performance of the trained models we report the $ER_{int,p}$ as a function of the maximum allowed error p, shown in Figure 3. As for this metric only pixels with known lower and upper bounds are considered, the only pixels that contribute are those that correspond to the BG class in the image to which regression is applied, but change their status to DT or CC before the end of the time series. For such pixels, it is more difficult to make correct predictions than for BG pixels. We note that the average width of these intervals is 31.7 days. Setting $p = 0$ leads to an interval error rate of 67.8%, which means that only about one third of the predictions are within the reference intervals. About two thirds of the predictions are less than 1 month off and 80% of the predictions are less then 2 months off. Allowing a maximum error of 6 months increases the accuracy to 95%.

We believe that these findings indicate that the resulting models can potentially be used to forecast upcoming forest damages, e.g. by identifying areas with a predicted RLT of a few months.

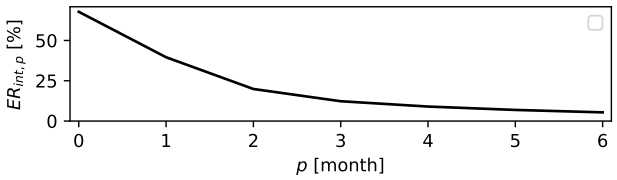

Figure 3. $ER_{int,p}$ for different allowed maximum errors p.

Figure 4 shows the image, the corresponding interval limits and the predictions for a sub-region of the test area. It can be seen that the predictions for non-forest areas are much larger than potentially affected areas, which is why a further masking of forest areas is not required.

5. CONCLUSION AND OUTLOOK

In this work we have presented a strategy for the automatic detection of early signs of forest damage in remote sensing imagery that does not require hand-labelled training data for that task. A bi-temporal classification model was trained and used to predict semi-labels for an unlabelled time series. Based on

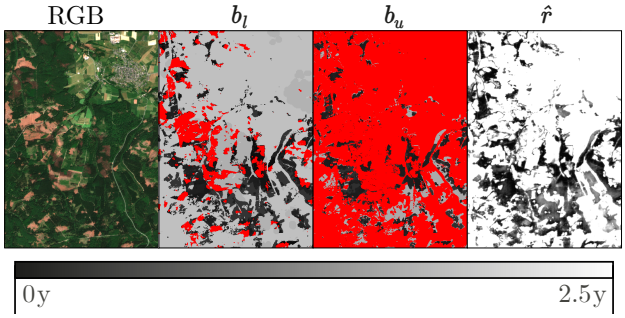

Figure 4. Example for an image in the test region (RGB), corresponding interval limits b_l,b_u and predictions $\hat{r}$ of the regression model. The grey values indicate the RLT. Red areas correspond to unknown limits.

the derived interval limits, a regression model for predicting the remaining life time was trained using a new loss function.

The proposed strategy for dealing with strong class imbalance in the data to be used for training a classifier was shown to drastically decrease the training time while achieving comparable performance compared to recent methods which address the same problem. Whereas training on the entire imbalanced data set using the standard CE loss leads to a slightly better performance, this comes at the cost of a much longer training time.

Furthermore, a strategy to create cleansed semi-labels for a time series was proposed. Using manually generated reference data it was shown that the cleansed semi-labels have a higher quality compared to the initial reference. We also have shown how to use the time series with the cleansed semi-labels and a cloud map to derive (potentially half-open) intervals for the RLT of each pixel and that a regression model can successfully be trained based on these limits. The RLT predicted by our method had a maximum error of 2 months in 80% of the investigated cases. The metrics presented for the regression model indicate that the model can potentially be used for early detection of signs for upcoming infestations, e.g. by focusing on areas with a short predicted RLT.

In order to improve the quality of damage forecasting, we see a high potential in designing forecasting models that do not only operate on a single image, but on a complete time series in order to grasp information about the whole phenology of forests. A related approach to partially consider the phenology would be to provide the model with explicit information about the capturing date of the input images. Additional data like information about the respective past weather conditions could also provide useful hints to assess the forest vitality and, thus to predict upcoming damages. Another approach to improve the forecasting is to investigate multi-task learning, i.e. to combine the RLT regression with a classification task. However, this would require additional label maps, for example of forest types.

ACKNOWLEDGEMENTS

This work was partially funded by the Federal Ministry for Economic Affairs and Climate Action, Germany (Bundesministerium für Wirtschaft und Klimaschutz, Funding codes 50EE2017A and 50EE2017B).

REFERENCES

Bressan, P. O., Junior, J. M., Martins, J. A. C., Gonçalves, D. N., Freitas, D. M., Osco, L. P., de Andrade Silva, J., Luo, Z., Li, J., Garcia, R. C., Gonçalves, W. N., n.d. Semantic segmentation with labeling uncertainty and class imbalance applied to vegetation mapping. *International Journal of Applied Earth Observation and Geoinformation*, 108, 102690–102699.

Cavallari, G. B., Ribeiro, L., Ponti, M., 2018. Unsupervised representation learning using convolutional and stacked autoencoders: A domain and cross-domain feature space analysis. *31st SIBGRAPI Conference on Graphics, Patterns and Images*, 440–446.

Chollet, F., 2017. Xception: Deep learning with depthwise separable convolutions. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1251–1258.

de Bem, P. P., de Carvalho Júnior, O. A., Guimarães, R. F., Gomes, R. A. T., 2020. Change detection of deforestation in the Brazilian Amazon using Landsat data and convolutional neural networks. *Remote Sensing*, 12(6), 901–919.

Eigen, D., Puhrsch, C., Fergus, R., 2014. Depth map prediction from a single image using a multi-scale deep network. *Advances in Neural Information Processing Systems 27*, 2366–2374.

He, K., Zhang, X., Ren, S., Sun, J., 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *IEEE International Conference on Computer Vision (ICCV)*, 1026–1034.

Holzwarth, S., Thonfeld, F., Abdullahi, S., Asam, S., Canova, E. D. P., Gessner, U., Huth, J., Kraus, T., Leutner, B. F., Kuenzer, C., 2020. Earth observation based monitoring of forests in Germany: a review. *Remote Sensing*, 12(21), 3570–3613.

Jawed, S., Grabocka, J., Schmidt-Thieme, L., 2020. Self-supervised learning for semi-supervised time series classification. *Advances in Knowledge Discovery and Data Mining - 24th Pacific-Asia Conference, Proceedings, Part I*, 12084, Springer, 499–511.

Kang, J., Chen, L., Deng, F., Heipke, C., 2020. Improving disparity estimation based on residual cost volume and reconstruction error volume. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Science*, XLIII-B2-2020, 135–142.

Lee, S.-H., Han, K.-J., Lee, K., Lee, K.-J., Oh, K.-Y., Lee, M.-J., 2020. Classification of landscape affected by deforestation using high-resolution remote sensing data and deep-learning techniques. *Remote Sensing*, 12(20), 3372–3388.

Liebel, L., Körner, M., 2019. Multidepth: single-image depth estimation via multi-task regression and classification. *2019 IEEE Intelligent Transportation Systems Conference*, 1440–1447.

Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P., 2017. Focal loss for dense object detection. *IEEE International Conference on Computer Vision (ICCV)*, 2980–2988.

Ren, Y., Zhang, X., Ma, Y., Yang, Q., Wang, C., Liu, H., Qi, Q., 2020. Full convolutional neural network based on multi-scale feature fusion for the class imbalance remote sensing image classification. *Remote Sensing*, 12(21), 3547–3567.

Rendón, E., Alejo, R., Castorena, C., Isidro-Ortega, F. J., Granda-Gutiérrez, E. E., 2020. Data sampling methods to deal with the big data multi-class imbalance problem. *Applied Sciences*, 10(4). 1276.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 9351, 234–241.

Sorensen, T., 1948. A method of establishing groups of equal amplitude in plant sociology based on similarity of species content and its application to analyses of the vegetation on Danish commons. *Biologiske Skrifter*, 5(4), 1–34.

Tilly, N., Reddig, F., Lussem, U., Bareth, G., 2020. First investigation of mediterranean oak tree vitality with high-resolution WorldView-3 satellite data: comparing ten vegetation indices and three machine learning classifiers. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Science*, XLIII-B3-2020, 1069–1076.

Wittich, D., Rottensteiner, F., 2021. Appearance based deep domain adaptation for the classification of aerial images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 180, 82-102.

Yang, C., Rottensteiner, F., Heipke, C., 2019. Towards better classification of land cover and land use based on convolutional neural networks. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Science*, XLII-2/W13, 139–146.

Zink, M., Zimmermann, R., 1997. Microwave remote sensing for monitoring forest vitality. *Third ERS Symposium on Space at the service of our Environment*, 1891–1897.