Remote Sensing of Environment 281 (2022) 113217

Contents lists available at ScienceDirect

Remote Sensing of Environment

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

Mapping forest in the Swiss Alps treeline ecotone with explainable deep learning

Thiên-Anh Nguyen [a,*], Benjamin Kellenberger [a,b], Devis Tuia [a]

[a] Environmental Computational Science and Earth Observation Laboratory, Ecole Polytechnique Fédérale de Lausanne, Switzerland
[b] Ecology and Evolutionary Biology, Yale University, New Haven, CT, USA

ARTICLE INFO

Edited by Marie Weiss

Keywords:
Forest mapping
Treeline
Explainable deep learning
Convolutional neural network

ABSTRACT

Forest maps are essential to understand forest dynamics. Due to the increasing availability of remote sensing data and machine learning models like convolutional neural networks, forest maps can these days be created on large scales with high accuracy. Common methods usually predict a map from remote sensing images without deliberately considering intermediate semantic concepts that are relevant to the final map. This makes the mapping process difficult to interpret, especially when using opaque deep learning models. Moreover, such procedure is entirely agnostic to the definitions of the mapping targets (*e.g.*, forest types depending on variables such as tree height and tree density). Common models can at best learn these rules implicitly from data, which greatly hinders trust in the produced maps. In this work, we aim at building an explainable deep learning model for forest mapping that leverages prior knowledge about forest definitions to provide explanations to its decisions. We propose a model that explicitly quantifies intermediate variables like tree height and tree canopy density involved in the forest definitions, corresponding to those used to create the forest maps for training the model in the first place, and combines them accordingly. We apply our model to mapping forest types using very high resolution aerial imagery and lay particular focus on the treeline ecotone at high altitudes, where forest boundaries are complex and highly dependent on the chosen forest definition. Results show that our rule-informed model is able to quantify intermediate key variables and predict forest maps that reflect forest definitions. Through its interpretable design, it is further able to reveal implicit patterns in the manually-annotated forest labels, which facilitates the analysis of the produced maps and their comparison with other datasets.

1. Introduction

Forests play an essential role for biodiversity conservation, recreation, carbon sequestration and climate. Monitoring forests closely is crucial in regard to the current climate and biodiversity crisis, for example to understand how forest habitats are influenced by changing climate and project their evolution in the future. To this end, forest maps are produced all over the world for official inventories, land planning and research in domains like forest ecology. Historically, forest mapping has first been conducted through field surveys, which only allowed measurements with slow updates and limited spatial coverage, with some areas remaining inaccessible. However, through the use of airborne and satellite sensors, remote sensing enables to monitor the earth's surface with imagery with dense coverage and at frequent intervals. Yet, the time cost of manual annotation involved still limited the ability to exploit this vast amount of remote sensing data. In recent years, advances in computer vision and machine learning have enabled to develop forest mapping methods that are highly automatized (Waser et al., 2017). The use of traditional machine learning methods marked a huge step in decreasing the need for manual annotation. Deep learning, which has revolutionized the domain of computer vision with Convolutional Neural Networks (CNNs; LeCun et al., 2015), has then been adapted for remote sensing applications with great success (Zhu et al., 2017; Ma et al., 2019), attaining remarkably high accuracies for large enough annotated datasets.

One challenging aspect of forest mapping is that different definitions of forest and forest types exist. They commonly depend on the entity producing the map (institution, country, individual) and on the management objective (Chazdon et al., 2016). Common criteria involve variables such as tree height and tree canopy density, structural form (*e.g.*, shrubs), land use history, as well as spatial extent (area, width/length). In remote sensing imagery, groups of trees oftentimes exhibit considerable variations in appearance, even within the same area of interest. This can greatly affect the mapping continuity across patches of forest within the context of photo-interpretation. As a result,

* Corresponding author.
E-mail address: thien-anh.nguyen@epfl.ch (T.-A. Nguyen).

https://doi.org/10.1016/j.rse.2022.113217
Received 8 April 2022; Received in revised form 3 August 2022; Accepted 5 August 2022
Available online 1 September 2022

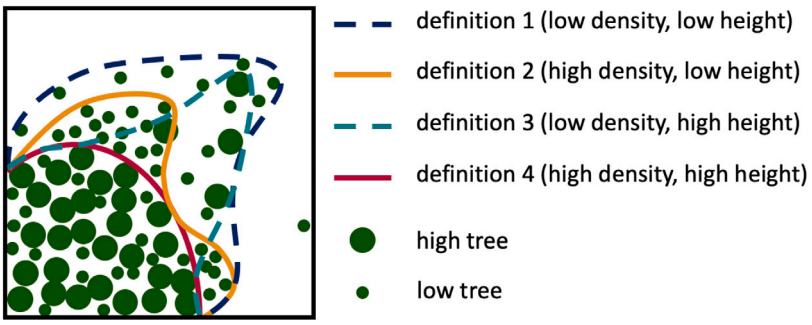


Fig. 1. Illustration of forest boundaries obtained from different forest definitions, over an area with a diffuse transition from forested to non-forested land. The four forest definitions have differing minimum tree canopy density and minimum tree height criteria.

forest maps from different sources, even when created by different annotators in the same institution, are likely to exhibit significant differences. Forest boundaries in particular can vary greatly depending on the forest definition. This becomes especially relevant at the alpine treeline, *i.e.*, the upper limit of forest in high-altitude regions, where the boundaries between forested and non-forested areas are diffuse and fragmented (Fig. 1).

Deep learning methods used for forest mapping are generally agnostic to such definitions and variables. They learn a complex function that directly maps pixel values to a forest category by learning on training labels, which themselves might or might not follow a strict forest definition. While deep learning-based forest maps can be validated using external datasets, they fail to provide a transparent and intelligible explanation for a given mapped area, *i.e.*, they lack interpretability. Ignoring domain knowledge also hinders model explainability, which is attained by providing domain-relevant explanations. Interpretability and explainability are crucial for trust in and usability of the maps by those they are intended for (Tuia et al., 2021). They also facilitate improvement of the model itself, as well as verification by domain experts (Samek et al., 2017).

Recent efforts in interpretable and explainable deep learning attempt to overcome the *black-box* behavior of deep learning methods, by making the models' decision process more accessible to humans and relevant to the domain. This is often achieved by identifying factors contributing to a given result and incorporating prior knowledge and semantic concepts into the models (Marcos et al., 2021; Roscher et al., 2020). In this direction, we attempt to incorporate prior knowledge from forestry into deep learning models in an explicit way to solve an automated forest mapping task. More specifically, we propose an explainable forest mapping method that explicitly predicts intermediate variables involved in the targeted forest definition, such as tree height and tree canopy density, and applies a set of rules that mirrors the forest definition. This effectively constrains the model in such a way that it is forced to predict forest types *by the rules*, increasing trust. As a second aspect, we address the inherent heterogeneity of forest maps due to varying definitions and subjectivity, as described above, and furnish the model with a correction pathway that is able to compensate for errors in the strict rule-based predictions. This way, the model allows users to contrast the predictions obtained through the constrained decision process with available, manually-annotated targets that might deviate from the definitions, and even quantify the degree to which the model outcome disagrees with the originally imposed forest definition rules.

All aspects combined, we seek to obtain a means of automated forest cover and type prediction that *(i.)* adheres to the forest definitions respected by annotators to create the reference maps; *(ii.)* explicitly predicts intermediate variables used for these definitions; and *(iii.)* can override the mechanistic combinatorial of the intermediate variables if required to compensate for potential biases, such as errors made by the model when predicting the intermediate variables or the heterogeneity of the target maps due to annotator biases. In a nutshell, our model can automatically infer forest type maps according to pre-defined rules and provide evidence for the explanation of the predictions.

We train and test our method over the Vaud and Valais Alps in Switzerland and focus on the treeline ecotone where forest mapping is particularly challenging, due to diffuse and fragmented forest boundaries. The model yields performance metrics with respect to the available labels that are close to that of a *black-box* version of the model, while providing intuitive spatially-explicit and task-relevant explanations for its decisions. Through a comparison with Swiss National Forest Inventory measurements, we demonstrate the relevance of the rule application in our explainable model, as well as the relevance of the provided explanations. We believe this work to be one of the first to study the explicit incorporation and understanding of expert-provided forest definitions into a deep learning model.

The rest of the paper is organized as follows: Section 2 describes a selection of related works. We describe the datasets we used in Section 3, and the methods in Section 4. We report our experiments and results in Section 5, before concluding in Section 6.

2. Related works

2.1. Forest mapping

Forest monitoring can be divided into three levels of information (Boyd and Danson, 2005): *(i)* forest extent and change dynamics (referred to as forest mapping hereafter), *(ii)* forest type, and *(iii)* forest biophysical and biochemical properties. We will consider *(i)* and *(ii)* as part of our forest mapping task.

Optical remote sensing data has become available at a wide range of spatial, spectral and temporal resolutions and constitutes a crucial source of information for forest mapping. While low to medium resolution satellite data with high revisit times such as Landsat enable the production of global, regularly updated forest maps (Song et al., 2018), authors in White et al. (2016) emphasize the potential of high and very high spatial resolution data (*i.e.*, with a ground sample resolution lower than 10 m) for forest inventories. Forest mapping can be viewed as a regression problem (Leboeuf et al., 2012), but it is more often tackled as a classification problem. Methods such as k-nearest neighbors, fuzzy algorithms, region growing or merging and maximum likelihood have been used to map forests by classifying pixels into a few forest categories (Waser et al., 2017).

A few automated forest mapping methods developed specifically for treeline ecotones are available in the literature. Several works have explicitly measured vegetation height or structure using Airborne Laser Scanning (ALS) data in order to characterize forest at the treeline (Ørka et al., 2012; Coops et al., 2013; Bolton et al., 2018). However, acquisition of high density ALS data is particularly costly in mountainous areas with complex topography. Alternatively, one could use Synthetic Aperture Radar (SAR) data, but the processing of SAR data remains challenging in these same areas due to geometrical distortion effects like layover and foreshortening (Morley et al., 2019). Most methods

thus rely on passive optical data such as aerial and multispectral satellite imagery. A pixel-wise Maximum Likelihood algorithm is applied to high-resolution satellite imagery in Hill et al. (2007). Authors explore categorized and continuous representations of the ecotone, derived from the obtained class probabilities rather than established definitions, and assess their relevance to various potential uses of the maps. Morley et al. (2019) apply pixel-wise logistic regression on multispectral satellite imagery for the application of four forest structure characterizations: the forest/non-forest definition of the Food and Agriculture Organization of the United Nations (Food and Agriculture Organization of the United Nations, 2020), a set of structural classes proposed in Harsch and Bader (2011) as well as a simplified version of them, and above-ground woody biomass as a continuous variable. Authors in Luo and Dai (2013) opt for an object-based approach, consisting of an unsupervised segmentation step followed by a classification of the obtained segments with k-nearest neighbors algorithm. The targeted forest classes (two forest classes and one *other* class) are defined by the authors specifically for the chosen study area. The mentioned methods have been successful at characterizing distinct treeline ecotones. Moreover, they share the advantage of using clear assumptions on the data distribution, and algorithms derived from domain knowledge. However, they are usually not intended for larger and more ecologically diverse study areas. In such case, they might suffer from a need for substantial manual parameter crafting, a lack of expressivity, or a lack of scalability to larger datasets.

Most recent remote sensing applications employ models from deep learning, such as convolutional neural networks (CNNs; LeCun et al., 2015). They have been shown to be highly versatile for land use/land cover mapping (Zhu et al., 2017) and vegetation remote sensing (Kattenborn et al., 2021), yielding high accuracy results even with simple fine-tuning on annotated remote sensing datasets of off-the-shelf methods. Methods developed for semantic segmentation, *i.e.*, pixel-level classification of images, are particularly suited for land cover mapping, including forest mapping. For example, Wagner et al. (2019) and Waser et al. (2021) show promising mapping results on land cover mapping with a U-net (Ronneberger et al., 2015). The first study maps forest type and disturbance in the Atlantic rain forest, whereas the second maps dominant leaf type in Switzerland.

The expressivity of deep learning models allows them to capture complex patterns over large and diverse areas. As a result, they generally exceed in performance over more traditional methods and have become the *de facto* standard for mapping applications. However, their success comes at the cost of explainability: oftentimes, measured concepts are large in numbers, exhibit complex interactions, and are of rather abstract nature. Crucially, deep learning models are generally *black boxes*, completely obfuscating their internal workings and processes that led to respective predictions. This greatly limits the credibility and trustworthiness of such models, no matter their performance. In this paper, we thus propose to improve on current semantic segmentation methods for forest mapping by making them inherently explainable following explicit forest definitions.

2.2. Interpretable and explainable deep learning

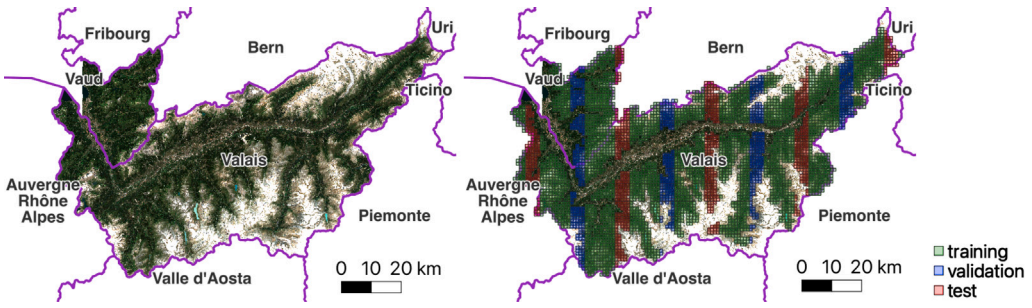
As deep learning techniques are maturing and being incorporated in real life decision-making or scientific systems, research in interpretability and explainability of deep learning models has become crucial, to ensure reliability and fairness (Samek et al., 2017), but also to foster scientific discovery (Roscher et al., 2020; Reichstein et al., 2019). *Interpretability* of a machine learning model can be defined as its ability to represent its decision process in a domain that is understandable by humans (Montavon et al., 2018). We define *explainability* as one way to attain interpretability, by obtaining insights based on domain knowledge that is relevant to a particular application (Stomberg et al., 2021). The main reason why common deep learning architectures lack interpretability and explainability is their high number of parameters. Indeed, deep learning architectures consist of a succession of tens to hundreds processing layers, such as convolutional filters in CNNs, that transform an input into features that help discriminate between classes or between samples.

Post-hoc methods are one way of make the interpretation of deep learning models possible, by attaching to a trained model a method which helps interpret each of the model's decisions. This includes approaches that identify which parts of the input contributed to a model's decision. For a remote sensing-based approach, this could correspond to locations in the image, *e.g.*, trees or buildings to detect forest and urban areas respectively. Such relationships can be revealed through occlusion sensitivity measurements (Zeiler and Fergus, 2014), which compare the response of the model with and without occluding small parts of the input. Instead of altering the input, saliency maps (Simonyan et al., 2014) and Grad-CAM measure the gradient of the output with respect to the input and the activations respectively. All these methods output a score for each location of the image that indicates the its relevance for the model's decision. Instead of looking at the input, LIME (Ribeiro et al., 2016) learns a simple interpretable model that approximates a complex *black-box* model, hoping that inspection of the parameters in the smaller model provides insights into the decision process of the larger *black-box* model.

Rather than using post-hoc interpretation tools, interpretability *by design* aims at making a model understandable through factoring explainability into the architecture of the model itself. A common strategy is to generate a feature map of reduced dimensions (often called *semantic bottleneck*) where key concepts (or *attributes*) of the final prediction outcome are represented. For instance, authors in Al-Shedivat et al. (2020) predict variables such as "Has electricity" or "Nightlight intensity" as intermediate concepts and combine them to derive the final outcome of predicting poverty. In Marcos et al. (2021), concepts including "man-made", "asphalt" or "ocean" are quantified to predict the scenicness (perceived beauty) of landscape photographs. The predicted concepts can be combined by a simple interpretable classifier to obtain a final decision. Such classifiers include probabilistic graphical models (Al-Shedivat et al., 2020) or linear mappings (Marcos et al., 2021; Levering et al., 2021). Desired properties of interpretable features are sparsity of the activations (Marcos et al., 2021) (*i.e.*, only few concepts contribute to the final outcome) and disentanglement of concepts in order to ease feature inspection. Sparsity and disentanglement of concepts can be sought for, whether along one of the concept feature map's dimensions (Ye et al., 2018; Losch et al., 2019), or numerically, for example in distinct clusters in the space of possible feature values (Stomberg et al., 2021). Intermediate concepts used as the basis elements for explanations can be derived from auxiliary datasets (Losch et al., 2019; Levering et al., 2021; Marcos et al., 2021) or domain knowledge about the underlying process (Ye et al., 2018). They can also be discovered during the learning process, for example by exploring the structure of the generated features (Stomberg et al., 2021) or through the learning procedure (Alvarez-Melis and Jaakkola, 2018).

A few studies tackle the incorporation of logical rules into deep learning models to enhance interpretability. In Hu et al. (2016), logic rules are incorporated in a deep neural network using knowledge distillation, with a student network which tries to match training labels while imitating a rule-constrained teacher network. A different approach is to train networks to choose a rule in a set of possible rules instead of directly assigning a class, with a generative (*Bayesian Rule List* (Letham et al., 2015)) or discriminative (*Rule-constrained Network* (Okajima and Sadamasa, 2019)) model.

Our work combines the semantic bottleneck approach with the use of logic rules in order to attain explainability. Our proposed model generates an intermediate feature map quantifying two key concepts, which are defined using domain knowledge consisting of the official definition of the targeted forest classes. To combine these intermediate predictions, we use logic rules with fixed parameters that are



(a) Sentinel-2 imagery over the study area (b) Image tiles repartition

Fig. 2. Study area, displaying the general appearance with Sentinel-2 imagery (a) and the partition into training, validation, and test sets (b).

also derived from the forest definitions. Context-specific behavior is accounted for using a parallel feature extractor, which learns to correct the predictions obtained by applying the logic rules. Through this correction pathway, the model is allowed to learn patterns found in the manually-annotated targets that are not explicitly expressed in the forest definitions.

3. Data

Study area. We explore the task of forest mapping over a 5897 km^2 area, comprised of the Valais canton and a subregion of the Vaud canton in Southern Switzerland (Fig. 2(a)). The study area is divided into square tiles of size 1 km × 1 km, corresponding to the tiling system of the SwissImage aerial images described in the following paragraph. In order to focus on forest boundaries between the subalpine and alpine zones, only the 2660 tiles intersecting the altitude range 1500–2500 m a.s.l are included in the dataset. The footprint of these 2660 tiles, as well as the distribution among training, validation and test sets, are displayed in Fig. 2(b) as green, blue and red squares, respectively.

SwissImage. We use SwissImage-10 aerial images from 2017 as the main input source. They are produced by the Swiss Federal Office of Topography Swisstopo and openly available online (Swisstopo, 2021a). The images have three spectral bands, red, blue and green (RGB), and a ground resolution of 10 cm over lowland regions and the main Alpine valleys and 25 cm over mountains, which includes our area of interest. The planimetric precision (1σ) is ±25 cm, except for vegetation and buildings for which the position of the top of the object can deviate further from its true location, depending on the acquisition angle of the aerial image. Images are acquired with a minimal sun elevation angle of 35° and pointing nadir.

While the near-infrared band and oblique images are also acquired during the SwissImage campaigns, we only use RGB, single-view nadir images to have a sense of the potential of this more widely available modality for forest mapping.

SwissALTI3D. SwissALTI3D is a digital elevation model (DEM) produced by Swisstopo and openly available online (Swisstopo, 2021d). We use the 50 cm resolution version as a second input source, in addition to the aerial images. The altimetric precision (1σ) is ±50 cm below 2000 m a.s.l and ±1 to 3 m above 2000 m a.s.l.

SwissTLM3D. To obtain training, validation and testing targets, we extracted forest labels from the topographic landscape model SwissTLM3D, produced by Swisstopo in vector format and openly available online (Swisstopo, 2021b). We extracted polygons corresponding to three types of forest: Open Forest (OF), Closed Forest (CF), and Shrub Forest (SF), as well as Woodland (WL).

The polygons were created manually by Swisstopo operators using stereoscopic SwissImage aerial images from 2015 to 2019, and a set of class definitions as annotation instructions. We identified criteria involving two key variables, tree height and tree canopy density (TCD), which together summarize the full set of criteria accurately (Table 1). Additional criteria not reported in the table influence the annotation in a less systematical way. They involve area and width of the forest polygons, as well as land use. For example, forest destroyed by fire and windfall remains labeled as forest after the event. Species composition is also a recurrent criterion, in particular for the SF class.

The dominant tree height and TCD variables are estimated by annotators to guide their annotation decisions. However, they are not explicitly recorded on a specific spatial grid. Dominant tree height generally corresponds to the mean height of the n highest trees in a given area. For the SwissTLM3D product, it is estimated by annotators by measuring the height of the highest trees in a moving disk of about 2500 m^2. There are no strict rules about the number of trees considered and the height averaging method. TCD is defined as the proportion of the ground covered by canopy. It is estimated visually by SwissTLM3D annotators by considering the aerial image content inside a moving disk of about 2500 m^2.

Due to the time span of the SwissTLM3D annotation process, and use of landuse history criteria for the annotation, mismatches between the SwissTLM3D annotations and the state of the forest cover in the 2017 SwissImage images might occur. While the extent of natural forest expansion is negligible in a two years span before or after the acquisition date of the 2017 aerial images, events such as clear cut or fires can cause abrupt forest loss of larger extent. However, such events are scarce in the area of interest. We thus consider such mismatches as annotation errors that are part of the label noise in our data.

NFI Vegetation Height Model. We used the Vegetation Height Model (VHM) produced for the Swiss National Forest Inventory (Ginzler and Hobi, 2015) as an additional target data source for our explainable model (Section 4.3). It is generated by first computing a digital surface model (DSM) using stereo-matching methods on stereo images from the same SwissImage campaigns as the aerial images we use as input. Then, the VHM is obtained by subtracting the DEM SwissALTI3D from the DSM. The vegetation height measurements are specified on a spatial grid of 1 m × 1 m.

NFI plots. We use plot-level data from the 4th Swiss National Forest Inventory (NFI4) (Brändli et al., 2020) as an external comparison dataset. Measurements were collected between 2009 and 2017 on a 1.4 km × 1.4 km grid whose vertices match the corners of the SwissImage tiles. Therein, plots are sampled as 50 m × 50 m square areas centered around points on the grid. For each plot, a decision on the presence and type (shrub, or forest without shrub) of forest is taken by visual inspection of stereo-images from SwissImage. Field measurements are then collected for all the plots attributed to forest, with priority given to the field measurements over the aerial image inspection decision in case of disagreement.

In the present study, we use the NFI plots located in the test set and exclude those that touch the boundaries of the training set. This results in a total of 321 plots, including 100 plots in the forest. Some of these plots are located at the boundaries of the dataset, but not touching the

Table 1
Main criteria for the SwissTLM3D definitions for the open forest, closed forest, shrub forest and woodland classes. TCD: tree canopy density.

Category	Description (Swisstopo, 2021c)	Min. dominant tree height (m)	TCD (%)
Open forest (OF)	Sparsely wooded area with vegetated soil	3	20 to 60
Closed forest (CF)	Wooded area, relatively dense, composed of one or more stands	3	over 60
Shrub forest (SF)	Area covered with shrubs (woody plants branched from the base)	3 for at least 1/3 of the area, 1 for the rest	over 60
Woodland (WL)	Small areas with trees and shrubs often along roads and waterways. Isolated forest areas are also recorded as wooded areas if they do not meet the defined minimum sizes of forest or shrub forest	–	–

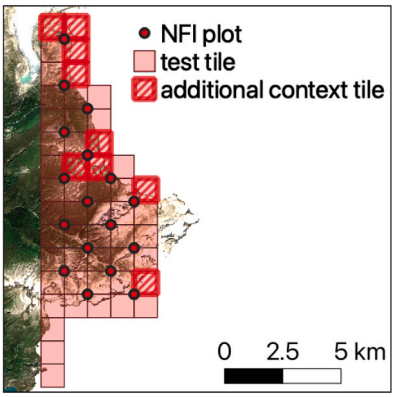


Fig. 3. Extract of the NFI plots and SwissImage tiles spatial arrangement (easternmost part of the test set).

training set. To compare our deep learning models predictions with the NFI measurements at these plots, we need provide some context to the deep learning models around the plot. We thus use tiles that were initially excluded from the dataset to provide some context to the segmentation model around the plot (Fig. 3).

The NFI forest definitions differ from the SwissTLM3D definitions, while depending on similar variables: dominant tree height, TCD and width. The TCD thresholds in the SwissTLM3D definitions are independent from the width value, whereas in NFI definitions the minimum TCD of a forested area to be classified as forest depends on its width (Fig. 4). Exceptions to the main criteria are mostly related to land use and species composition. The variables themselves are measured and defined differently. TCD is measured by determining the landcover type of 25 points on a regular grid in the plot area. Dominant tree height is defined as the mean height of the 100 biggest trees per hectare.

4. Methods

4.1. Segmentation task

We approach forest mapping as a semantic segmentation task, using SwissImage aerial imagery and the SwissALTI3D DEM as inputs to a CNN and SwissTLM3D forest annotations as targets (Fig. 6). We included the DEM as an input to our model, as we believe altitude and topography-related features are important to detect the presence of forest. An ablation study numerically motivating the use of the DEM as an additional input is also provided in Appendix A.1.

We map every pixel to one forest type (OF, CF, and SF), or to Non-Forest (NF), at 1 m resolution. Examples of each forest type are displayed in Fig. 5(a). We divide the four-classes task into two sub-tasks: the *forest type* sub-task with target classes OF, CF, and SF, and the binary *forest presence/absence* sub-task with target classes NF and Forest (F). The F class of the *forest presence/absence* task corresponds to a union over the OF, CF, SF, and Woodland (WL) polygons of the SwissTLM3D labels. Predictions for both tasks are produced by the model at every pixel. The choice of using these two sub-tasks is motivated by several elements. First, OF, CF and SF share similar textures. The task of classifying F against NF is both semantically meaningful and relatively balanced (Table 4). Since most model weights are shared between the two tasks, we hope that training the *forest presence/absence* task benefits the *forest type* classification. Second, the use of the *forest presence/absence* task allows including the WL class in the F category to obtain more fine-grained forest boundaries, even though the forest type is unknown for the WL pixels. This way, the model can learn from the WL class even though it is not in the final set of classes. The benefit of this 2-task structure is assessed by comparing our proposition against a model predicting a single multi-class output in Appendix A.2.

We generate 1 m resolution raster format targets from the SwissTLM3D polygons using the *forest type* and *forest presence/absence* tasks classes described in the previous paragraph (Fig. 5). For the *forest type* task, the pixels that do not correspond to OF, CF, and SF polygons are set as unknown forest type to be ignored by the *forest type* loss function at training. We will refer to the generated targets as "TLM targets".

4.2. Baseline architecture

Our baseline model consists of a *black-box* CNN, which directly outputs a forest segmentation map (Fig. 6(a)). In practice, this map consists of a tensor of size $C \times W \times H$, for C classes (F for the *forest presence/absence* task, and OF, CF and SF for the *forest type* task, respectively) and images of width W and height H. For each pixel location, a prediction score in the interval $[0, 1]$ is obtained by using a sigmoid activation (Eq. (5)) for the *forest presence/absence* task and a softmax activation (Eq. (4)) for the *forest type* task. We will refer to this baseline model as *black-box* (BB) model.

4.3. Explainable deep learning model architecture

We use the semantic bottleneck approach (Marcos et al., 2021) to build our explainable model (Fig. 6(b)). More specifically, we replace the *black-box* forest class predictor part with a branch that consists of two sequential steps: first, a *concepts extractor* predicts intermediate concepts that are relevant to the task of forest mapping. These have also been used to create the TLM targets in the first place during manual annotation. Second, a *rule module* translates these intermediate predictions into class probabilities using fixed, simple heuristics, which also apply to the manual annotation process for the TLM labels. This pathway essentially constitutes the basis for the prediction of the eventual forest classes via intermediate products and combinatory rules. In addition, we allow the model to modify these rule-based class probabilities with a *correction module*. This module is informed primarily by the rule module, but receives additional features from the base feature extractor, referred to as "correction activations". These additional features allow the model to learn when and where to best adjust the rule-based outputs. All together, we refer to this model as *semantic bottleneck* (SB) model and describe each of its components in more detail in the following paragraphs.

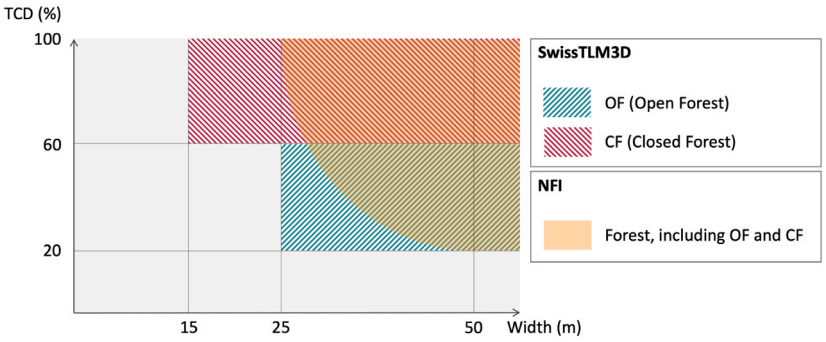
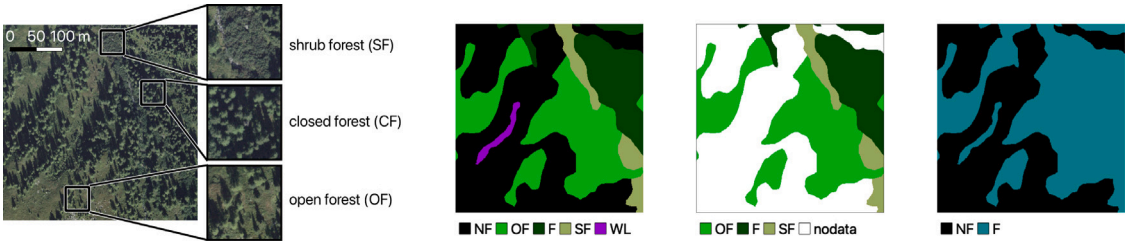
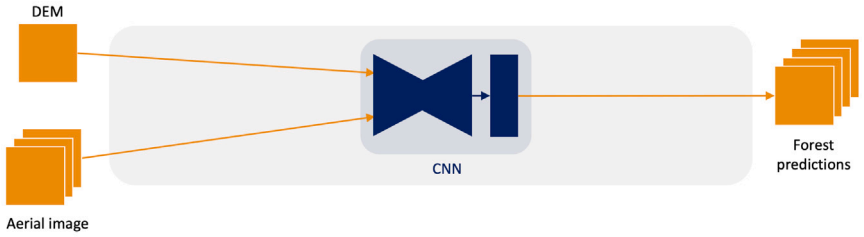


Fig. 4. Comparison of the TCD and width criteria of NFI and our interpretation of the SwissTLM3D definitions. *Source:* Adapted from Brändli et al. (2020).

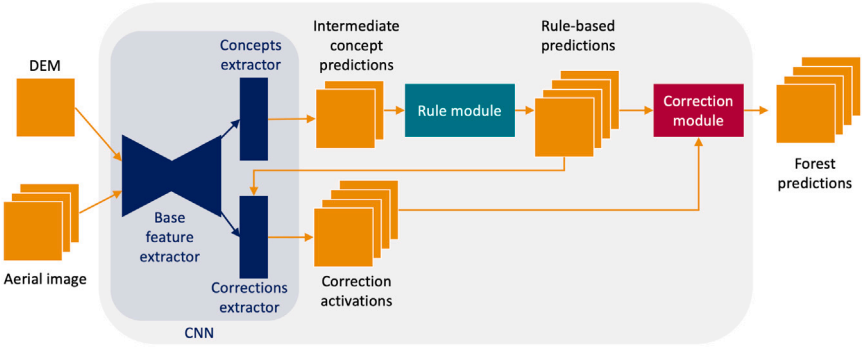


(a) Aerial image (b) SwissTLM3D annotations (c) *Forest type* target (d) *Forest p/a* target

Fig. 5. Aerial image and associated targets extracted from SwissTLM3D annotations. p/a: presence/absence.



(a) *Black-box* (BB) model



(b) Semantic bottleneck (SB) model

Fig. 6. Flowchart of the forest mapping methods.

Intermediate concepts prediction. Before predicting the forest classes, the model explicitly quantifies intermediate concepts that are relevant to the task by means of the *concepts extractor*. We leverage prior knowledge about the targets and choose tree height (TH) and TCD as intermediate concepts. Indeed, these are key variables of the definitions of the target forest classes. Our semantic bottleneck thus consists of a two-channel feature map, of same width and height as the final forest map, with each channel corresponding to regression values of one of the two intermediate concepts, respectively. The semantic bottleneck is generated by using the features extracted by the base feature extractor, and further processing them with a *concepts extractor* block composed of additional convolutional layers (Fig. 6(b)).

We obtain targets to train the two intermediate regression tasks using pre-processing functions applied to the VHM. Given the absence

Table 2
Rules enforced by the rule module. Cell colors indicate unique categories, listed in Table 3.

TH (m)	TCD (%)		
	[0, 20)	[20, 60)	≥ 60
[0, 1)	NF	NF	NF
[1, 3)	NF	NF	NF/SF
≥ 3	NF	OF	CF/SF

of precise definition of the chosen variables, we define simple pre-processing functions that aim to imitate how an annotator would estimate the two concepts:

- To obtain TCD targets, we first threshold the VHM at 1 m height and then average the obtained binary map with a moving circular window of 28 m radius.
- To obtain TH targets, we assume an average tree diameter of 3 m and use a *maxpool* function, which replaces each pixel value by the local maximum in a square 3 × 3 pixels neighborhood, to better reflect tree height.

Both targets are produced at 1 m resolution, even though the pre-processing functions use a larger neighborhood. Note that we do not attempt to precisely compute the *dominant* tree height, which would have involved more complex pre-processing functions, for example by using the *variable window filter* algorithm (Popescu and Wynne, 2004) to localize treetops. Instead, we keep the modifications of the VHM product to a minimum, to avoid introducing additional bias.

A discussion about the quality of the obtained training targets is developed in Section 5.2.3.

Rule module. The role of the rule module is to combine the intermediate concepts estimated in the semantic bottleneck to obtain forest class probabilities. We simplify the SwissTLM3D class definitions into a small set of rules involving the concepts predicted in the semantic bottleneck (Table 1). The rule module assigns forest probabilities using this set of rules accordingly.

Some of the categories obtained by applying the chosen set of rules are impure, *i.e.*, they apply to more than one forest class. This is due to set of rules not reflecting the class definitions completely. For example, variables differentiating shrubs and non-shrub trees are missing. This prevents isolating the SF class.

We enforce the rules by assigning hard-coded class probabilities. More specifically, for each possible category output by the rules, *i.e.*, for each cell of Table 2, we set probabilities as defined in Table 3. For each category, we assign a strong probability to the class indicated by the rules. Equal probabilities are assigned to classes that cannot be distinguished using only the rules. We also assign equal probabilities to the *forest type* classes for the NF category, since it does not indicate preference for any of the *forest type* classes.

In practice, we assign log-probabilities computed from the probability values of Table 3. We compute these log-probabilities using inverse softmax (Eq. (1)) and inverse sigmoid (Eq. (2)) functions for the *forest type* and *forest presence/absence* class probabilities respectively.

$$a_i = \log p_i + c,\ \forall i \in \{\mathtt{OF}, \mathtt{CF}, \mathtt{SF}\} \tag{1}$$

$$a_{\mathrm{F}} = \log\left(\frac{p_{\mathrm{F}}}{1 - p_{\mathrm{F}}}\right) \tag{2}$$

The inverse softmax (Eq. (1)) is bound to a constant c. Any choice of c leads to the same probabilities after applying the softmax function to the obtained log-probabilities. In practice we choose c so as to obtain log-probabilities that are approximately centered around zero, which facilitates the learning process. ϵ_{rule} in Table 3 should be in the interval $(0, \frac{1}{3})$ to keep the probability values in Table 3 between 0 and 1 and to keep higher probabilities for the classes indicated by the rules. The choice of ϵ_{rule} within this interval controls how strictly the rules are enforced: the smaller ϵ_{rule}, the stronger the rules enforcement.

Analogous to the BB model, the output of the rule module is a tensor of log-probabilities for each class. They can be used on their own to obtain a rule-based forest segmentation map.

Correction module. The role of the correction module is to modify the output of the rule module to better match the segmentation target, for example in cases of incorrect predictions and/or errors emerging from the training data annotation process. To do so, we dedicate a number of convolutional layers to the generation of correction activations, represented as the *corrections extractor* box in Fig. 6(b). These layers use the feature maps provided by the base feature extractor, as well as the rule-based log-probabilities, as input. As such, the correction activations are conditioned by both the content of the aerial image and the output of the rule module. These generated correction activations are used in the correction module, where they are added element-wise to the rule-based log-probabilities. The motivation behind the design of the correction module is to overcome three challenges:

1. the limits of the semantic predictions, *i.e.*, to compensate for errors in the intermediate TH and TCD estimations;
2. the limits of the rule module caused by the simplification of the class definitions:
   (a) by solving the impure categories in Table 2 through identification of shrubs;
   (b) by increasing the spatial smoothness of the segmentation.
3. The noise in the targets, by compensating potential systematic biases in the targets with respect to the rules.

For the *forest type* task, instead of generating correction activations for each class, the *corrections extractor* produces correction activations $\hat{y}_{\mathrm{corr,\ OF}}$ and $\hat{y}_{\mathrm{corr,\ CF}}$ for classes OF and CF respectively, and $\hat{y}_{\mathrm{corr,\ SF}}$ for the SF class is computed using Eq. (3):

$$\hat{y}_{\mathrm{corr,SF}} = -\hat{y}_{\mathrm{corr,OF}} - \hat{y}_{\mathrm{corr,CF}}. \tag{3}$$

This forces the correction activations to increase the log-probabilities of a subset of the *forest type* classes and decrease the log-probabilities of the other(s), and thus prevents the correction process from shifting all the rule-based log-probabilities without modifying the relative scores between classes. This is not necessary for the binary *forest presence/absence* task since the model only predicts *forest presence* probabilities. We also enforce sparsity of the correction activations of both sub-tasks using an L_1 penalty (see Section 4.4). This prevents the correction activations to radically change the class log-probabilities through addition with the rule-based log-probabilities. It thus helps control the amount of deviation from the rules we allow through the correction module.

The output of the correction module corresponds to corrected log-probabilities for the segmentation task.

4.4. Training and inference procedure

Loss functions. To train models for the base segmentation task, we use a cross-entropy loss for the *forest type* task and a binary cross-entropy loss for the *forest presence/absence* task. The losses are computed on predicted probabilities obtained from the output activations using a softmax function (Eq. (4)) and a sigmoid function (Eq. (5)) respectively.

$$p_i = \mathrm{softmax}(\hat{y}_i) = \frac{e^{\hat{y}_i}}{\sum_{j \in \{\mathtt{OF}, \mathtt{CF}, \mathtt{SF}\}} e^{\hat{y}_j}},\ \forall i \in \{\mathtt{OF}, \mathtt{CF}, \mathtt{SF}\} \tag{4}$$

$$p_{\mathrm{F}} = \mathrm{sigmoid}(\hat{y}_{\mathrm{F}}) = \frac{1}{1 + e^{-\hat{y}_{\mathrm{F}}}} \tag{5}$$

We sum the two loss terms to obtain $\mathcal{L}_{\mathrm{seg}}$, as shown in Eq. (6), where p_X and y_X refer to a predicted probability and its target for a particular class X. To partially compensate for class imbalance, the *forest type* cross-entropy loss, $\mathcal{L}_{\mathrm{forest\ type}}$, is weighted using class-specific

Table 3
Hard-coded class probabilities enforced by the rule module, for each unique category of Table 2. ϵ_{rule} is a stabilizing factor used to keep the log-probabilities small in absolute terms.

Category from Table 2	*Forest type* task			*Forest presence/absence* task
	$p_{\texttt{OF}}$	$p_{\texttt{CF}}$	$p_{\texttt{SF}}$	$p_{\texttt{F}}$
NF	1/3	1/3	1/3	ϵ_{rule}
OF	$1-2\epsilon_{\text{rule}}$	ϵ_{rule}	ϵ_{rule}	$1-\epsilon_{\text{rule}}$
NF/SF	ϵ_{rule}	ϵ_{rule}	$1-2\epsilon_{\text{rule}}$	0.5
CF/SF	ϵ_{rule}	$0.5-\epsilon_{\text{rule}}/2$	$0.5-\epsilon_{\text{rule}}/2$	$1-\epsilon_{\text{rule}}$

weights w_c that are inversely proportional to the class frequencies in the training set. The class frequencies are shown in Table 4. The BB model is trained using $\mathcal{L}_{\text{seg}}$ as the only loss term. For the SB model, $\mathcal{L}_{\text{seg}}$ is computed using the corrected log-probabilities and the segmentation targets for both the *forest type* and *forest presence/absence* tasks.

$$\mathcal{L}_{\text{seg}} = \underbrace{-\left[\sum_{c\in\{\texttt{OF},\texttt{CF},\texttt{SF}\}} w_c y_c \log(p_c)\right]}_{\mathcal{L}_{\text{forest type}}} \underbrace{-\left[y_{\texttt{F}}\log(p_{\texttt{F}}) + (1-y_{\texttt{F}})\log(1-p_{\texttt{F}})\right]}_{\mathcal{L}_{\text{forest p/a}}} \quad (6)$$

The loss function for the SB model contains additional terms (Eq. (7)). Scalars λ_{corr} and λ_{sem} are hyperparameters to control the influence of the correction L_1 sparsity penalty and the semantic concept regression loss respectively.

$$\mathcal{L} = \mathcal{L}_{\text{seg}} + \lambda_{\text{corr}}\|\hat{y}_{\text{corr}}\|_{L_1} + \lambda_{\text{sem}}\mathcal{L}_{\text{sem}} \quad (7)$$

$\mathcal{L}_{\text{sem}}$ is the sum of the regression losses used for the semantic bottleneck concepts TH and TCD (Eq. (8)).

$$\mathcal{L}_{\text{sem}} = \mathcal{L}_{\text{TH}} + \mathcal{L}_{\text{TCD}} \quad (8)$$

We use a mean squared error loss to train the TCD regression. For TH, we use a mean squared error loss in log space. The error in log space for a pixel i is defined as $\log(y_i+\alpha) - \log(\hat{y}_i+\alpha)$, where y_i and $\hat{y}_i$ are the TH target and prediction values at pixel i, and α is a stabilizing factor. Since the rules involve thresholding the TH prediction at low values, we set $\alpha = 0.1$ to encourage the model to focus on small height values.

Decision procedure at inference. Hard decisions are obtained from the class probabilities using Eqs. (9) and (10):

$$z_{\text{forest type}} = \underset{i}{\text{argmax}}\ p_i,\ i \in \{\texttt{OF},\texttt{CF},\texttt{SF}\} \quad (9)$$

$$z_{\text{forest p/a}} = \texttt{F} \text{ if } p_{\texttt{F}} > 0.5, \text{ else } \texttt{NF} \quad (10)$$

To obtain a four-classes segmentation map with classes NF, OF, CF, and SF, we simply assign the chosen forest type based on the highest softmax-probability for the pixels classified as F, and assign the other pixels to NF. We will refer to this map as *combined four-classes predictions*.

5. Experiments and results

5.1. Experiments

Model hyperparameterization. The combination of a U-net model structure (Ronneberger et al., 2015) with a Res-Net encoder (He et al., 2016) has been shown to be highly adapted to semantic segmentation tasks for both remote sensing (Chu et al., 2019; Cao and Zhang, 2020; Malkin et al.; Gazzea et al., 2021) and non-remote sensing (Xiao et al., 2018; Siddique et al., 2021) applications. We thus use a modified U-net with a ResNet-18 encoder as the base semantic segmentation model (dark blue components in Fig. 6). This directly corresponds to the BB model. For the SB model, we choose to use all but the two last CNN layers as the base feature extractor, allowing features to be shared between the intermediate concepts prediction task and the correction task. In the concepts extractor and the corrections extractor, we allow each branch's two final convolutional layers of the CNN decoder to specialize to intermediate concept prediction and correction activations generation, respectively. For both BB and SB models, since the DEM used as input has a resolution twice inferior to the resolution of the aerial images (Table 6), the aerial images are first fed to the model and the DEM is then concatenated to the features of same height and width obtained after the first convolutional layer.

Table 4
Class distribution (in %). *train+* corresponds to the training samples where at least one pixel is labeled as one of the forest classes.

		train	train+	val	test
Forest type	OF	6.1	6.1	5.9	6.1
	CF	87.1	87.0	88.6	88.2
	SF	6.8	6.9	5.4	5.7
Presence of forest	NF	73.3	60.4	68.5	74.8
	F	26.7	39.6	31.5	25.2

We train and evaluate the following models:

- BB: *black-box* model (Fig. 6(a))
- SB: semantic bottleneck model (Fig. 6(b)), using as forest predictions
  - the *rule-based* predictions, *i.e.* the output of the rule module, and
  - the *corrected* predictions, *i.e.* the output of the correction module.

For the SB model, we set ϵ_{rule} to 0.001 and λ_{corr} to 1.

Training setup. We train the models for 20 epochs with the AdamW optimization algorithm (Loshchilov and Hutter, 2019), with batches containing 32 128 × 128 pixels patches randomly extracted from the tiles.

We use a curriculum learning (Bengio et al., 2009) approach to alleviate the effect of class imbalance. We identify a subset of the training set, called *train+*, in which each tile has at least one pixel labeled as forest; *i.e.*, we exclude all tiles that are entirely labeled as NF. These samples usually correspond to alpine grasslands, rocky areas or snow-covered areas above the treeline. The *train+* set is slightly more balanced that the full training set (Table 4) for the *forest presence/absence* task. We start the learning procedure by training the model for two epochs solely on the *train+* set, to maximize the occurrence of forest. We then inject tiles that do not belong to *train+* into the set of training samples in increasing number along the training epochs. More specifically, we include five of them for two epochs, and then double the number of such tiles every two epochs. We limit the maximum number of tiles without forest to 320 out of 874.

Dataset composition. The sizes of the training, validation and test sets are detailed in Table 5; their spatial distribution is displayed in Fig. 2(b). We train the models on the training set and evaluate them on the validation set to guide the model design and hyperparameter choices. The results reported in this section are obtained by evaluating the models on the previously unseen test set, and using TLM targets as a reference, unless specified otherwise.

Model inputs and training targets used for the BB and SB models are summarized in Table 6.

Table 5
Dataset composition.

	Training	Validation	Test	Total
Number of tiles	2660	596	673	3929
Proportion	68%	15%	17%	100%

Table 6
Input and target datasets used for the BB and SB models.

		Resolution (m)	BB	SB
Inputs	Aerial image (SwissImage)	0.25	✓	✓
	DEM (SwissALTI3D)	0.50	✓	✓
Targets	TH	1		✓
	TCD	1		✓
	TLM	1	✓	✓

Table 7
Overall accuracy scores across training, validation and test sets (combined four-classes predictions), using TLM targets as reference.

	Training	Validation	Test
BB	**0.93**	**0.93**	**0.94**
SB	**0.93**	0.92	0.93

Table 8
Overall accuracy scores on the test set, using TLM targets as reference.

	Forest type	Forest presence/absence
BB	**0.89**	**0.93**
SB	0.80	**0.93**
SBrules^{-}	0.83	**0.93**
SBcorr^{+}	0.81	**0.93**

Data preprocessing. Prior to being fed to the model, all the input data (aerial images and DEM) are normalized using mean and standard deviation values computed on all the training and validation samples. The TH and TCD targets are divided by the standard deviation of target values over the training and validation sets. This allows us to reduce the regression predictions values to a small range and thus keep the model parameters reasonably small. It also naturally balances the two regression loss terms.

Implementation. We implement our models in Python using the Pytorch library (Paszke et al., 2019). Our implementation is available at: https://github.com/thienanhng/ExplainableForestMapping.

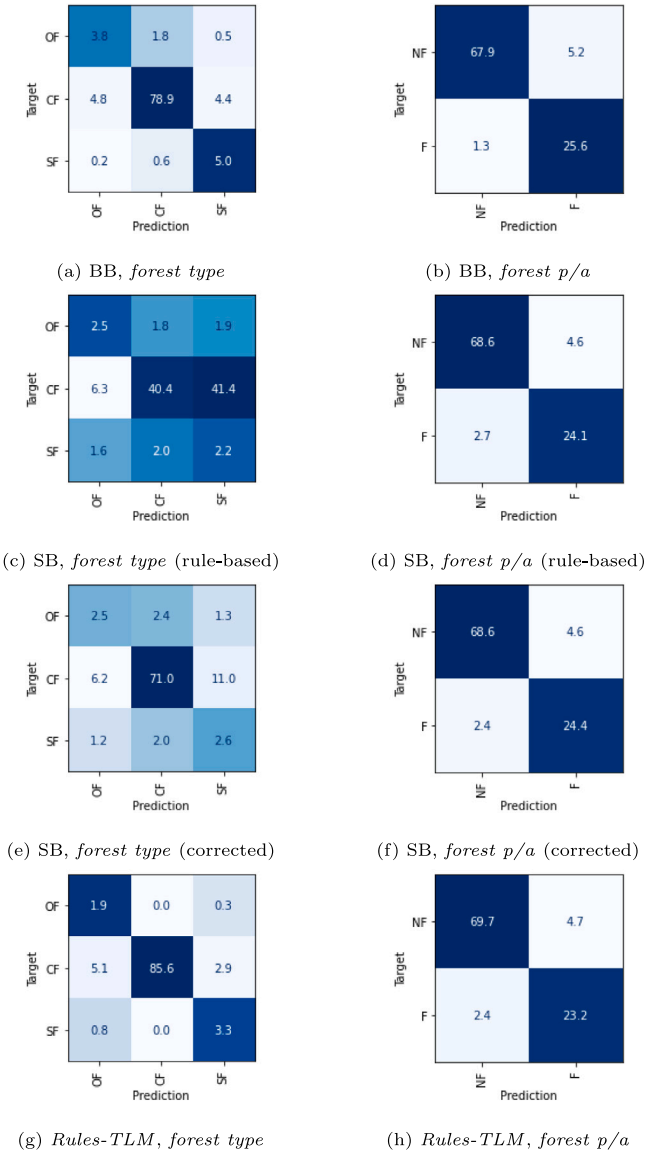
5.2. Results

5.2.1. Forest mapping results

Table 7 displays the overall accuracy scores over the training, validation and test sets for models BB and SB. The scores are consistent for both models across training, validation and test sets, which indicates good generalization, resp. low degrees of overfitting to the training set.

Overall accuracy scores for each task on the test set are displayed in Table 8. They reveal that compared to the BB model, the SB model performs similarly for the *forest presence/absence* task with respect to the TLM targets. However, for the *forest type*, there is a noticeable performance degradation.

This is further confirmed in Table 9, which contains class-specific metrics on the test set, as well as *average* scores, where the contribution of each class is equal, and thus the performance of minority classes has more influence than in the metrics reported in Table 7. We notice a significant performance degradation for the *forest type* task between the BB and SB models, but a similar performance on the *forest presence/absence* task. *Forest type* prediction is a challenging task in the first place. Indeed, the class imbalance is more pronounced than for the *forest presence/absence* task and the targets are particularly noisy



(a) BB, *forest type* (b) BB, *forest p/a*

(c) SB, *forest type* (rule-based) (d) SB, *forest p/a* (rule-based)

(e) SB, *forest type* (corrected) (f) SB, *forest p/a* (corrected)

(g) *Rules-TLM*, *forest type* (h) *Rules-TLM*, *forest p/a*

Fig. 7. Confusion matrices obtained on the test set on both *forest type* and *forest presence/absence* (*p/a*) tasks, using TLM targets as reference. Results are obtained from: (a–b) the predictions the BB model, (c–d) the rule-based and (e–f) the corrected predictions of the SB model, as well as (g–h) the application of the rules on the intermediate concept targets (see Section 5.2.3).

for the least frequent classes (OF and SF). While the BB model heavily overpredicts the most frequent class CF (Fig. 7(a)), the rule module of the SB model predicts the other classes (OF and SF) more often, even though this makes the prediction match the TLM targets less (Fig. 7(c)). The results after correction represent a compromise between the BB predictions and the rule-based predictions (Fig. 7(e)).

Fig. 8 shows visual results on selected example areas with more or less diffuse forest boundaries. Extract A contains sharp boundaries shaped by land use whereas extracts B and C are examples of diffuse forest expansion. Both models demonstrate good performance where the forest boundaries are sharp. The predictions differ mostly by the presence of an OF belt around the forest boundaries predicted by the SB model (column "Prediction SB"), and more generally by the prediction of forest types OF and SF. The prediction of OF between CF and NF areas is expected due to the chosen TCD definition: the large spatial extent of the averaging kernel used to compute the TCD targets leads to gradually decreasing TCD values at the forest boundaries. In

Table 9
Per-class and averaged IoU and F-1 scores on the test set, using TLM targets as reference.

	Forest type								Forest presence/absence					
	OF		CF		SF		Average		NF		F		Average	
	IoU	f-1	IoU	f-1	IoU	f-1	IoU	f-1	IoU	f-1	IoU	f-1	IoU	f-1
BB	**0.34**	**0.51**	**0.87**	**0.93**	**0.47**	**0.64**	**0.56**	**0.69**	**0.91**	0.95	**0.80**	**0.89**	**0.85**	**0.92**
SB	0.18	0.31	0.77	0.87	0.14	0.25	0.37	0.48	**0.91**	0.95	0.78	0.87	0.84	0.91
SBrules$^-$	0.20	0.32	0.82	0.90	0.22	0.26	0.41	0.49	0.89	0.95	0.76	0.87	0.83	0.91
SBcorr$^+$	0.19	0.32	0.78	0.88	0.21	0.34	0.39	0.51	**0.91**	**0.96**	0.78	0.88	**0.85**	**0.92**

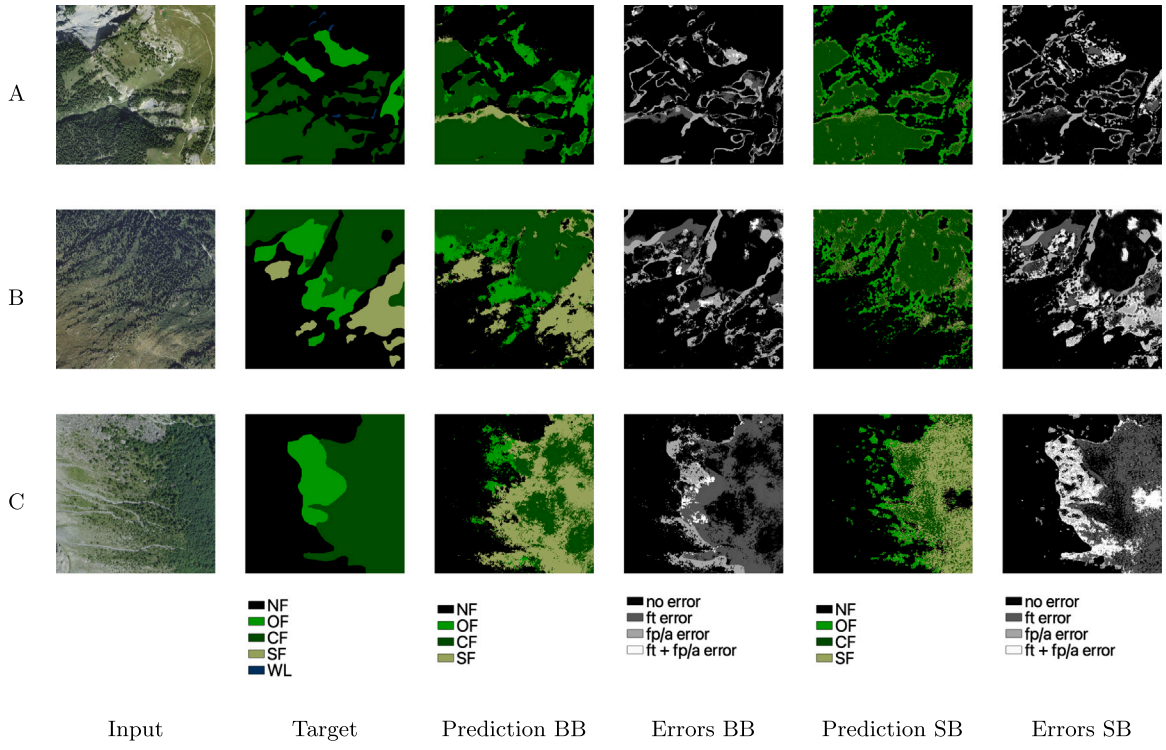


Fig. 8. Visual extracts of the forest mapping results. ft: *forest type*, fp/a: *forest presence/absence*. Note that *forest type* and forest presence/absence errors can co-occur, for example when the model predicts an incorrect *forest type* while predicting NF for a F pixel..

extracts B and C, the BB and SB predictions contain large differences in the predicted forest type, with SF being less frequently predicted for the latter. For both models, forest boundaries are particularly spatially detailed around patches of low density forest, which contrasts with the smooth shapes of the target annotations, but is truthful to the input image. Regarding the BB model, extracts A and B show satisfying results in terms of forest type, while extract C shows large areas classified as CF in the target but as SF by the model. When looking at the input image in the corresponding area, we can argue that the absence of large shadows in the corresponding area makes the forest type identification difficult.

Overall, the results for the BB model demonstrate a good ability to generate spatially detailed forest maps. The results of the SB model are less similar to the targets. However, they better reflect the explicit definitions underlying the target classes, for example through the gradual transition from CF to OF to NF (Fig. 8). Additionally, they provide an estimation of intermediate concepts, as shown in the next section, allowing thorough interpretation of the model decisions and errors.

5.2.2. Intermediate concepts estimations

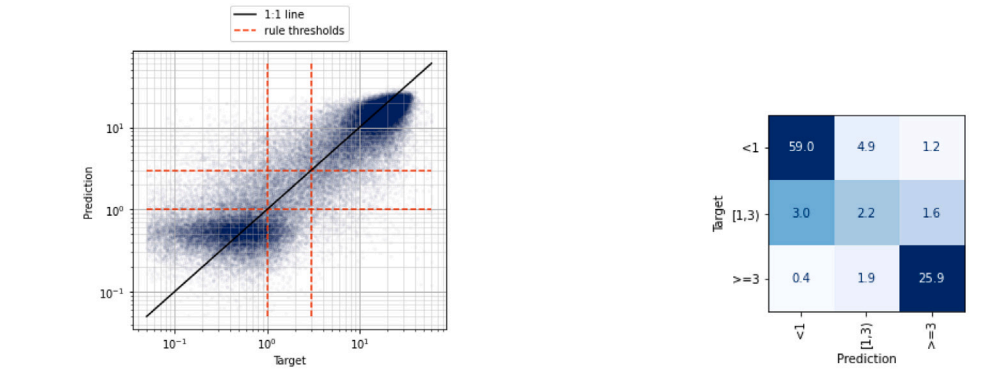
In this paragraph we evaluate the intermediate predictions of the SB model.

Tree height prediction. Fig. 9 and Table 10 show quantitative results of the TH prediction task; a visual example is provided in Fig. 10. We expect the TH prediction task in the semantic bottleneck to be challenging due to the wide range of heights, the scarcity of intermediate height values, as well as the use of monocular input images rather than stereoscopic images. The prediction scores in Table 10 indicate that textural cues are sufficient to roughly approximate TH. The model can differentiate between low and high vegetation but precise height estimation remains difficult, in particular for heights below 2 m (Fig. 9(a)). Fig. 10 shows an example where the shrub-like texture of the forest confuses the model, resulting in an underestimation of height. In term of categories involved in the rules of our SB model, the most frequent height categories, TH < 1 and TH ≥ 3, are mostly classified accurately, while more than half of the predictions in category $[1, 3)$ actually belong to the lower category (Fig. 9(b)).

Table 10
R^2 and Root Mean Squared Error ($RMSE$) scores for the TH and TCD predictions on the test set ($RMSE$ scores in m and % respectively).

	R^2	$RMSE$
TH	0.75	4.2
TCD	0.89	12.1

The performances could be improved by using stereo-matching of pairs of images (Ginzler and Hobi, 2015) or by linking sensor viewing



(a) Log-scale scatter plot (200 random pixels per tile) (b) Confusion matrix

Fig. 9. Tree height prediction results on the test set (SB).

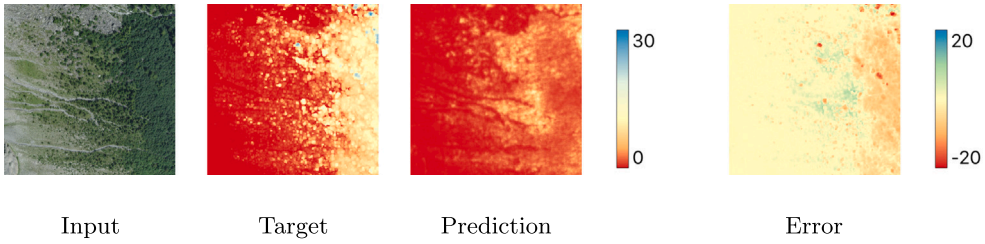
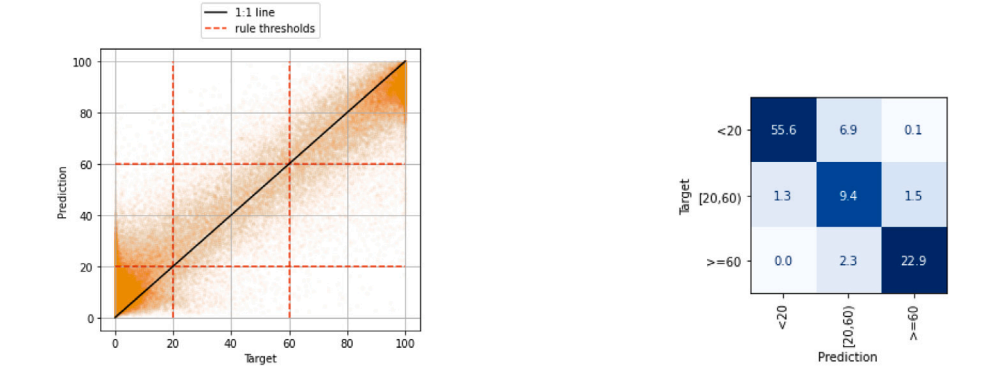


Fig. 10. Visual extracts of the tree height predictions (in m) on the test set (SB).



(a) Scatter plot (200 random pixels per tile) (b) Confusion matrix

Fig. 11. Tree canopy density prediction results on the test set.

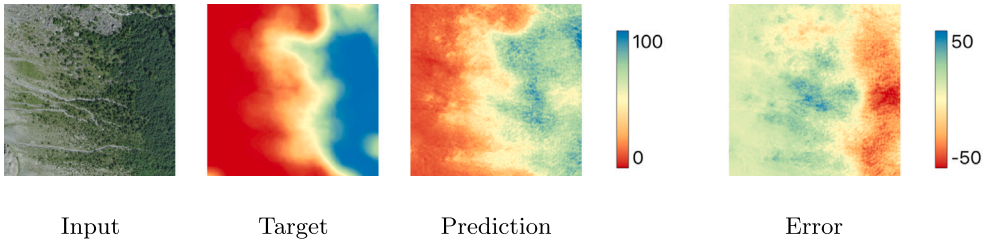
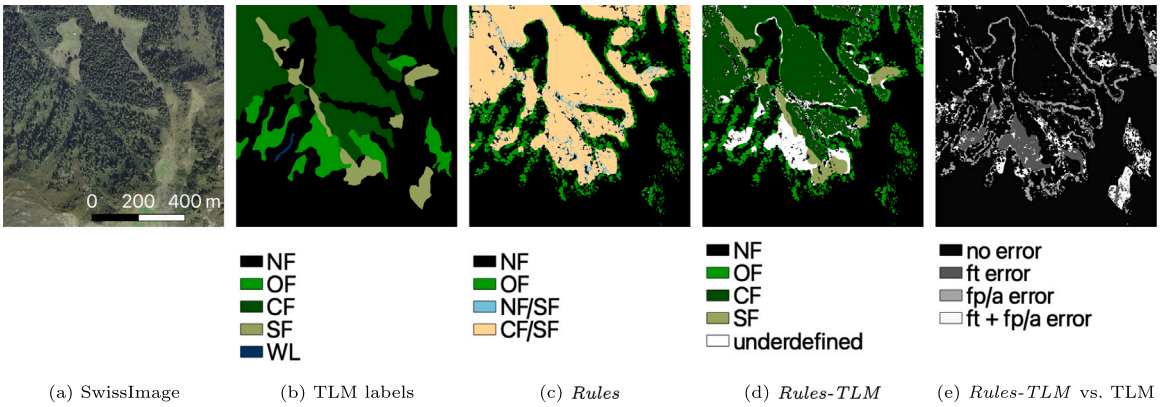


Fig. 12. Visual extracts of the tree canopy density predictions (in %) on the test set (SB).

information with tree shadows (Leboeuf et al., 2012). However, we consider the present performances sufficient for the purpose of exploring intermediate concept estimation for explainability.

Tree canopy density prediction. The TCD distribution shows a similar trend in the distribution of values, with frequent low and high values inside and outside of the forest, respectively, and less frequent intermediate values corresponding to forest borders (Fig. 11). The predictions scores of Table 10 indicate a good estimation of TCD, even though low values tend to be overestimated and high values tend to be underestimated (Figs. 12 and 11(a)). The predictions fall mostly into the right rule categories (Fig. 11(b)).



(a) SwissImage (b) TLM labels (c) *Rules* (d) *Rules-TLM* (e) *Rules-TLM* vs. TLM

Fig. 13. Segmentation results by applying rules on intermediate concept targets. *Rules*: map obtained by applying the rules to the intermediate targets. *Rules-TLM*: *Rules* map with TLM disambiguation. ft: *forest type*, fp/a: *forest presence/absence*.

Using the above results, we can assess the fidelity of the rule enforcement on the test set with respect to the chosen set of rules, which is conditioned by the quality of the intermediate concept predictions. Indeed, the low accuracy for the intermediate categories for both concepts ($[1, 3)$ m for TH and $[20, 60)$ % for TCD) introduces considerable noise into the rule enforcement, especially for classes SF and OF that involve those intermediate categories (Table 2). For the other categories of both concepts, the higher accuracies indicate a more accurate application of the rules.

These observations suggest ways to improve both BB and SB models. The difficulty of the SB model to estimate intermediate TH values shows that the BB model necessarily relies on concepts other than tree height. It also suggests that additional input data giving more indications about tree height would benefit both models.

5.2.3. Quality of the intermediate concept targets

In this section, we analyze to what extent the targets we generated for the intermediate TH and TCD estimation (Section 4.3) match the implicit TH and TCD estimations of the TLM annotators. While we do not have access to the TLM annotator estimations of these variables, we can have a sense of whether the TH and TCD rule thresholds are aligned. We do so by applying the set of rules (Table 2) on the TH and TCD targets and compare the obtained categories with the TLM labels.

An example is shown in Fig. 13. The image in Fig. 13(c) shows the categories obtained after applying the rules of Table 2 on the TH and TCD targets. Some areas fall into impure categories (NF/SF and CF/SF). The image in Fig. 13(d) shows the segmentation map after disambiguation of these impure categories. To obtain this map, we choose the TLM label if it is included in the pair of classes proposed by the rules. If not, no choice is made and the pixels are displayed as underdefined. We refer to this map as *Rules-TLM* and use it to compute confusion matrices over the entire test set (Figs. 7(g) and 7(h)).

The visual extract suggests that the SB model trained on our TH and TCD targets can theoretically reproduce the TLM CF and SF classes quite well. However, the confusion matrix (Fig. 7(g)) reveals that the agreement for the SF class leaves room for improvement. Both the visual extract and the confusion matrix show significant disagreement for the SF class. It consists of frequent confusion with OF and CF classes, the presence of an OF belt at the forest boundaries, and more fine-grained predictions matching the location of trees.

The observed discrepancies suggest that the TH and TCD threshold values are not perfectly aligned between our targets and the estimations of the TLM annotators, especially in the intermediate range involved in the OF and SF rules. The discrepancies might also originate from differences between the simplified rules and the complete set of instructions, and the compliance of the annotations with the set of instructions itself.

Table 11
Hyperparameters of the semantic bottleneck models.

	ϵ_{rule}	λ_{corr}
SB	0.001	1
SBrules$^-$	0.2	1
SBcorr$^+$	0.001	0.2

However, most discrepancies are limited to the close vicinity of the forest boundaries and the least frequent, most challenging forest types. This justifies the sparsity constraint applied to the correction activations, since correction of the rule-based predictions should be necessary only in a few locations, typically diffuse forest boundaries at the treeline with shrub expansion. This spatial sparsity of the correction activations is an essential quality contributing to the explainability aspect of the model.

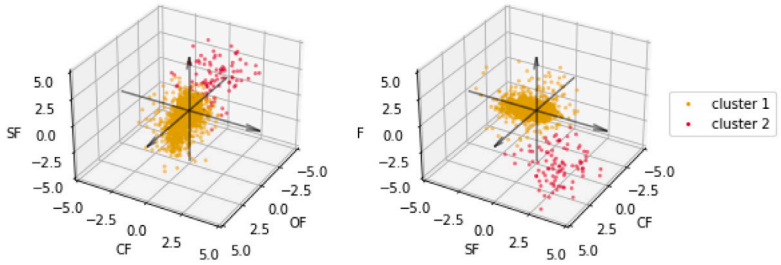
The present experiment can also be interpreted as simulating a SB model with perfect TH and TCD estimation w.r.t our targets, with a correction module that is rule-compliant and able to disambiguate classes perfectly w.r.t. the TLM targets. With this interpretation in mind, we can compare the confusion matrices in Fig. 7g–h with those of the actual SB model in Fig. 7e–f. While the SB model yields nearly identical results for the *forest presence/absence* class, the *forest type* results differ significantly. This highlights the gap between the predictions and the targets, for both the semantic bottleneck concepts and the correction module. Note that the *forest type* confusion matrices have different row-wise sums because in the present experiment the *forest type* label is not always available.

5.2.4. Analysis of the rule implementation and correction process

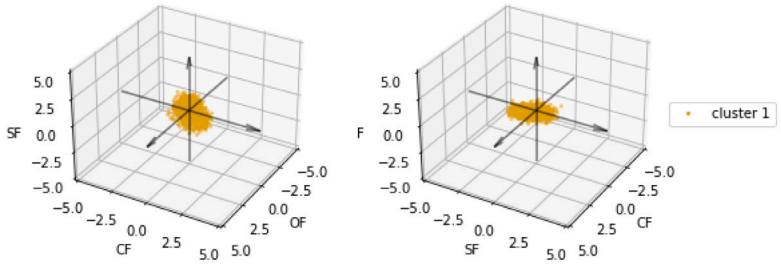
In this section, we further analyze the rule implementation and correction process of our explainable architecture and their parameterization. We train and evaluate the two following models, additionally to the SB model (Table 11):

- SBrules$^-$: a semantic bottleneck model with a softer rules enforcement than the SB model, *i.e.*, a higher ϵ_{rule} value
- SBcorr$^+$: a semantic bottleneck model for which we allow stronger corrections than for the SB model, *i.e.*, with a lower λ_{corr} value.

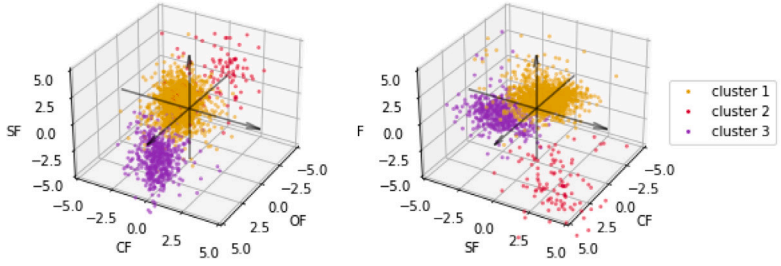
The specific hyperparameters of each semantic bottleneck model are summarized in Table 11. The three SB models are fine-tuned from a common pre-trained SB model obtained by training only the intermediate concept estimation task, backpropagating the $\mathcal{L}_{\text{sem}}$ loss only, for five epochs. This renders the three SB models more comparable by making their intermediate predictions more similar, and thus the effect of the correction module parameterization more visible.



(a) SB



(b) SBrule^{-}



(c) SBcorr^{+}

	forest p/a	*forest type*		
	F	OF	CF	SF
cluster 1	n	n	n	n
cluster 2	--	-	n	+
cluster 3	-	+	n	-

(d) Summary of the three identified clusters shared across SB, SBrule^{-} and SBcorr^{+} models.

Fig. 14. Clustered correction activations on the test set (2000 points picked randomly). n: neutral, +: in favor, ++: strongly in favor, -: against, --: strongly against. p/a: presence/absence.

By inspecting the correction activations (Fig. 14), we observe that the amount of large-valued correction activations is higher for model SBcorr^{+} than for model SB. This is expected, because model SBcorr^{+} was trained with a softer sparsity penalty on the correction activations. The distributions are split into several modes of correction, except for the SBrule^{-} model, where the values are concentrated in one narrow mode around zero.

We use the K-means algorithm (Lloyd, 1982) to cluster the correction activations of each model independently. Interestingly, the obtained clusters can be matched across models, yielding three possible correction modes in total (Fig. 14). Cluster 1, centered around 0, is found across the 3 models and corresponds to no or subtle corrections. Cluster 2, found in models SB and SBcorr^{+}, tends to disadvantage the F class (*forest presence/absence*) and favor the SF *forest type* over OF. For model SBcorr^{+}, the remaining values can be gathered in a 3rd cluster that favors OF over SF.

The effect of the correction activation clusters on the forest predictions is visible in Fig. 15. Depending of the model's hyperparameters, non-zero correction activations might or might not cause a change in the predicted class.

- For the choice between equiprobable classes in the rule-based prediction, *i.e.*, for corrections that are compliant with the rules, small non-zero activations are sufficient to favor one of the classes. This is the case for areas assigned to the CF/SF category (speckled areas in Fig. 15's *Rule-based predictions* column,

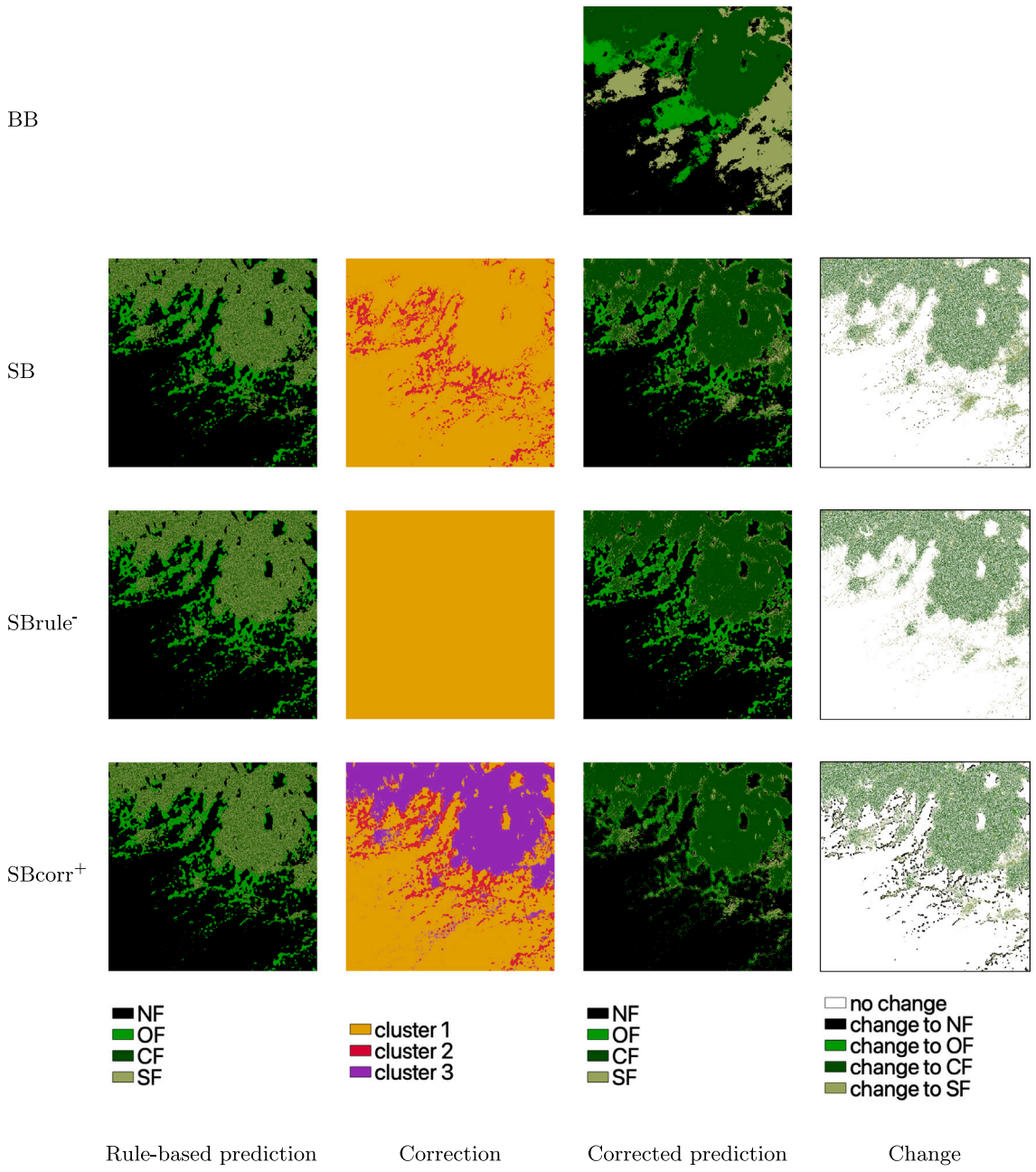


Fig. 15. Correction process for extract B of Fig. 8. *Rule-based prediction* column: rule-based class predictions obtained from probability maps output by the rule module. *Correction* column: clustering of the correction activation maps obtained with the K-means algorithm (see Fig. 14). *Corrected prediction* column: class predictions obtained after applying the correction. *Change* column: change maps comparing the *Prediction* and *Rule* maps, showing the new class in case of change.

corresponding to the orange cell in Table 2). The small correction activations of cluster 1 suffice to assign most of the pixels to CF instead of SF (models SB and SBrule$^-$). With model SBcorr$^+$, trained with a looser correction penalty, the correction activations are not only used to choose the CF class, but also to soften the high-confidence rule-based class probabilities (cluster 3, *Correction* column in Fig. 15).

- For corrections that override the rules, their effect depends on the correction hyperparameters, in particular on hyperparameter λ_{corr}, which controls the strength of the correction activations. This is visible at the boundaries between forested and non-forested areas where the rule-based model systematically predicts OF (Fig. 15). This pattern has been identified by the correction module as uncharacteristic of the TLM targets. Indeed, the correction activations of cluster 2 attempt to replace OF by either NF or SF at the OF belt (Figs. 14 and 15, models SB and SBcorr$^+$). In the case of the SB model, the correction activations are not sufficient to provoke a class change, whereas with the SBcorr$^+$ model, which generates larger correction values, the correction can result in a change to NF or SF.

In terms of segmentation accuracy, models SB, SBrule$^-$ and SBcorr$^+$ yield similar results (Tables 8 and 9). Since the difference between the three models take place mostly at the forest boundaries (Fig. 15), they only generate small differences in the segmentation scores.

Overall, these results illustrate how the λ_{corr} hyperparameter can be modulated to obtain corrected predictions that strictly follow the set of rules, or are more similar to the BB model predictions. Such modulation can for example be used to compensate for differences in annotation quality between regions and/or annotators. In this study we left λ_{corr} constant across the entire dataset.

Table 12
Comparison labels derived from NFI4. KOMBWALDENT: forest category, DGRAD25KL20: TCD in steps of 20%.

KOMBWALDENT	DGRAD25KL20 (%)				
	[0, 20]	(20, 40]	(40, 60]	(60, 80]	> 80
Non-forest	NF	NF	NF	NF	NF
Forest	NF	OF	OF	CF	CF
Shrub forest	NF	NF	NF	SF	SF

One striking result is the "laziness" of the correction for the SBrule$^-$ model. Despite keeping the λ_{corr} hyperparameter unchanged between the SB and SBrule$^-$ models, we observe a collapse of the correction activations. Such low correction activation are not caused by near perfect rule-based predictions, but rather by a re-balancing of the loss terms. With the current setting, the relaxation of the prior knowledge-based constraints thus results in a loss of explainability.

An important takeaway of the present analysis is that with proper parameterization, the correction activations represent a valuable source of explanations, separating explicit prior knowledge-based reasoning from implicit patterns learned from training data. Moreover, the clear separation of the correction activations into a few correction modes helps provide global explanations that complement pixel-wise explanations.

5.2.5. Comparison with National Forest Inventory data

In previous sections, we trained and evaluated our models using TLM targets. However, we would like to assess the compliance of the BB and SB models to the class definitions using data coming from a different source than the actual TLM targets. Indeed, the latter do not perfectly reflect the class definitions, due to noise, as well as biases caused by implicit hand-annotation principles. As a consequence, the BB model naturally yielded better metrics. In this section, we thus leverage NFI plot data as a comparison dataset obtained through a different annotation process. The comparison also allows us to illustrate how the explainability of the SB model helps understanding the models' outputs.

We derive forest labels from NFI4 plot data over the test set. Table 12 details how we derived labels from NFI4 that best match SwissTLM3D definitions. From this process, we obtain 100 plots labeled as forest out of 321, with 22, 76, and 2 plots labeled as OF, CF and SF, respectively. We then compare our deep learning models' predictions with these labels at plot location. We post-process our model predictions with morphological operations to ensure the robustness of the predictions at the exact plot locations.

Confusion matrices of the TLM labels and the BB and SB model predictions against the NFI-derived labels at the selected NFI plots are shown in Fig. 16. Overall, we obtain good agreement with the NFI-derived labels when considering only the forest presence/absence. The *forest type* confusion matrices include the plots labeled as forest in both the NFI-derived labels and the compared source. Given the low number of NFI plots assigned to the OF and SF classes (22 and 2 respectively), analysis of the results is more delicate. Overall, the BB model reproduces the statistical distribution of the TLM targets, with almost all plots assigned to CF and a few plots to OF or SF. The SB model shows a different distribution with several CF plots assigned to SF.

Most plots showing disagreement are located close to forest boundaries. For example, in Fig. 17a, the TLM labels consist of a CF area with a sharp boundary with NF. However, the NFI TCD estimation method indicates a TCD between 20 and 40%, which we attribute to OF. In the segmentation maps obtained with BB and SB models, the plot is at the boundary of an OF area. The position of the forest boundary and the choice of the forest type remain unexplained for the BB model. For the SB model, inspection of the intermediate predictions indicate that the TH value is just above the 3 m criterion, and the TCD prediction is still largely above the 20% criterion. The correction activations indicate low confidence in the position of the boundary.

Table 13
Computational complexity of the BB and SB models.

	BB	SB
Number of trainable parameters	18.9×10^6	19.0×10^6
Training time per epoch (2660 tiles)	9 min	11 min
Inference time per tile	0.53 s	0.61 s

Disagreement among the set of predictions and labels can also be caused by land use history. Fig. 17b corresponds to an area where forest is starting to regrow after a fire event in 2003. The TLM label reflects the past state of the forest before the event, while the NFI information reflects the undergoing regrowth. Both models' predictions indicate NF. For the SB model it is due to a low estimated TH value, and no correction of the rule-based predictions. This example illustrates how the intermediate concept estimations of the SB model inform the user about the causes of the decision of classifying the plot into NF. The BB outputs the same decision, but for unknown reasons.

The comparison conducted here suggests that the SB model indeed outputs predictions that reflect the set of rules we extracted from the class definitions better than its BB counterpart. This validates the intermediate concept predictions and the correction activation as a reliable source of explanations.

5.2.6. Computational complexity

The SB model contains additional model parameters due to the specialization of the last layers of the CNN to intermediate concept prediction and correction activations generation. Moreover, during training, the SB model has additional loss terms for the correction activations sparsity constraint and the intermediate concepts regression. Training and inference times for the BB and SB models are displayed in Table 13. They were obtained by running our implementation on a Linux workstation with an AMD Ryzen 9 3950X CPU and an NVIDIA GeForce RTX 3090 graphics card. They demonstrate that the use of the SB model instead of the BB model introduces a proportionally small increase of the number of trainable parameters. Training and inference times are only slightly increased and remain suitable for large-scale forest mapping.

6. Conclusion

In this paper, we explored the use of prior knowledge of forest definitions to make a deep learning model for forest mapping more explainable. More specifically, prior knowledge consisted in the identification of a few key concepts, tree height and tree canopy density, along with a set rules combining these variables, that approximate the class definitions. Crucially, these definitions directly correspond to procedures employed during manual forest annotations. We designed the explainable model based on this knowledge, with a semantic bottleneck where key concepts are explicitly quantified, and a rule application module that assigns class-probabilities based on the estimated concept values. We enabled the model to learn additional patterns contained in the targets through correction activations, which modify the rule-based predictions to better match the targets. We applied our model on a challenging forest mapping task over Switzerland, using high resolution aerial imagery, and focused on the treeline ecotone where different forest definitions can lead to radically different maps.

The constraints of the explainable model lead to predicted forest maps that are less similar to the targets than a *black-box* counterpart, but nicely reveal patterns in the targets that are not covered by the rules. Analysis of the correction activations clearly emphasized these patterns, especially systematic open forest predictions at forest boundaries yielded by the explainable model. One could want to explicitly incorporate such pattern into the rules or into the correction process

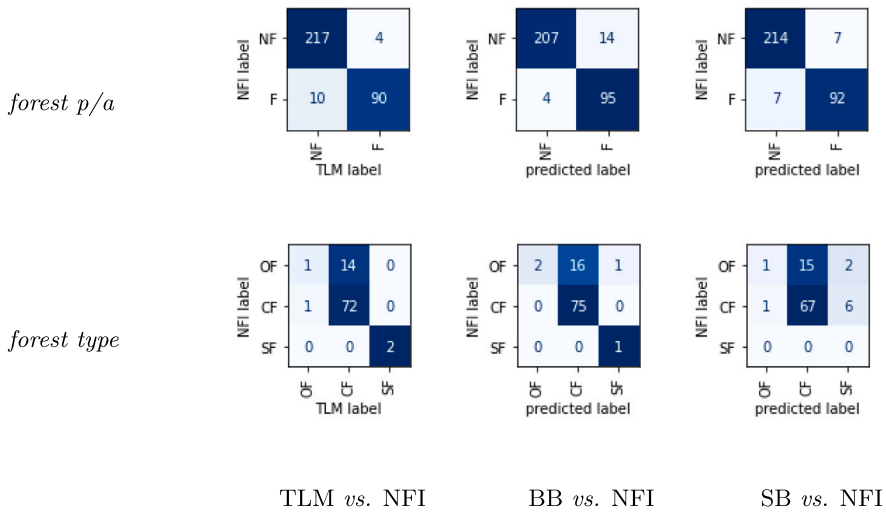


Fig. 16. Confusion matrices of the TLM labels and BB and SB predictions using the NFI-derived labels as a reference.

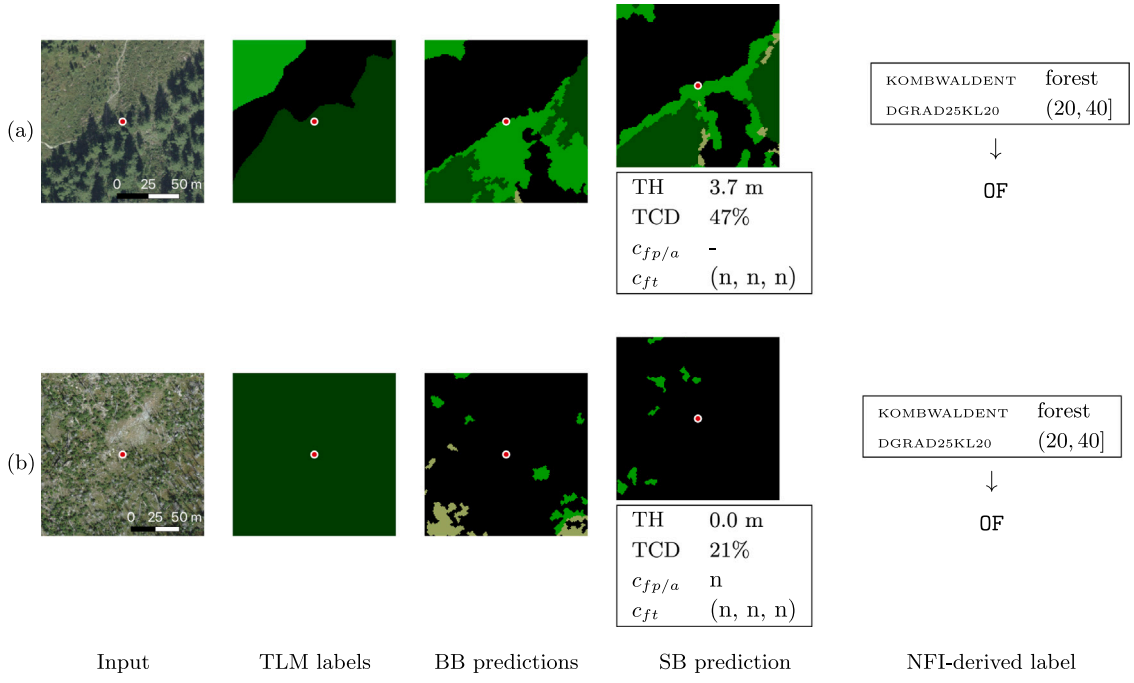


Fig. 17. Extracts of TLM labels and model predictions at two NFI plots. Center red dot: location of the plot. $c_{fp/a}$ and c_{ft} are the *forest presence/absence* and *forest type* (OF, CF, SF) correction activations (n: neutral, +: in favor, -: against, --: strongly against). The color legend is similar to Fig. 8.

as new prior knowledge about the forest mapping process. Moreover, evaluation of the intermediate concept estimations indicates ways to make the rule enforcement more accurate and as a result, improve the overall model. Our results suggest that the model is missing some cues to quantify tree height. This cannot be detected by using only a *black-box* model.

Besides guiding the model's design, constraining the model's decision process also enables extracting simple explanations for any model decision. Indeed, along with each of its decisions, our explainable model provides additional predictions, namely the two intermediate concepts (tree canopy density and tree height) and the correction value, for each class and for each pixel. Paired with the set of rules encoded in the model, they form concise and logical explanations that mirror the decision process of a human annotator. Comparison with Swiss National Forest Inventory plot data demonstrated how these explanations can facilitate interpretation and links between datasets. Another possible use of these explanations is editing of the model predictions to match one's specific needs—for example, one can inspect the explanations to make local manual modifications. One can also alter the rules application or correction process to modify the predictions in a systematic way.

More generally, the use of a semantic bottleneck forces the model to partially or completely rely on some concepts that we know are relevant to the task. This limits the use of spurious cues by the model. Additionally, if the intermediate concept predictions are accurate, it prevents the model from producing illogical, or physically implausible predictions. If the intermediate predictions are wrong, inspection of the intermediate concept estimations enables easily attributing the final prediction error to one or several of these intermediate predictions. All these elements allow for more trust in the model.

However, it should be noted that while we made the final layers of decision more explainable, the intermediate concepts predictions and

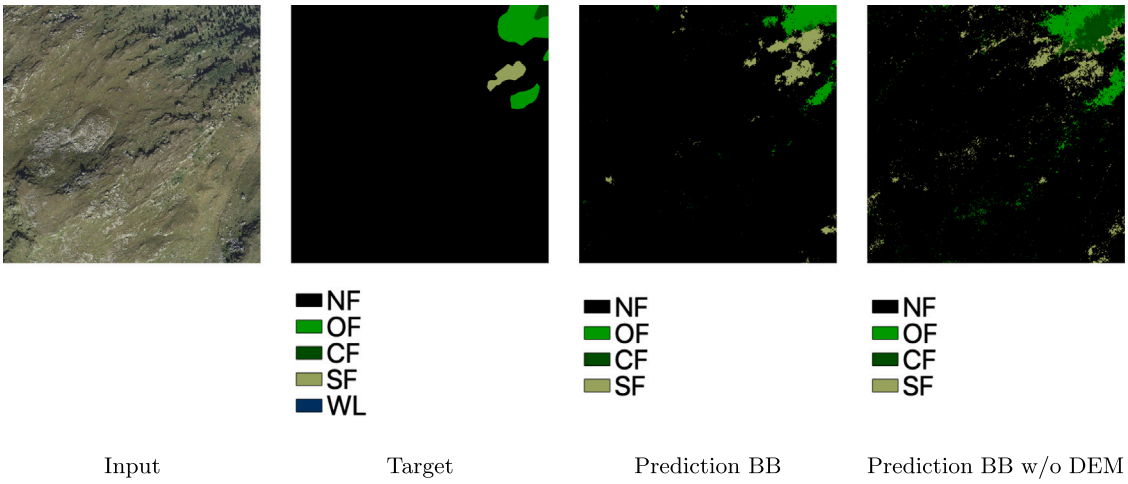
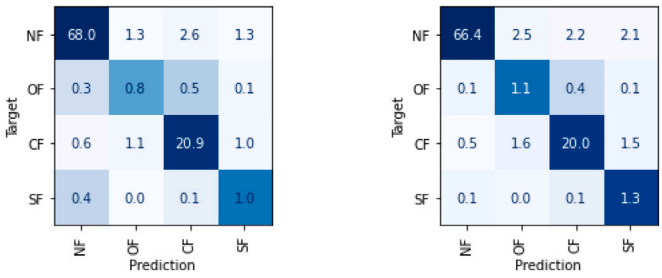


Fig. A.18. Visual extracts of the forest mapping results with the BB model trained with and without a DEM as input.



(a) BB (b) BB (single task)

Fig. A.19. Confusion matrices of the BB model with and without using a task hierarchy, using TLM targets as reference.

the correction activations generation still rely on a *black-box* CNN. We chose to leverage the expressive power of CNNs to measure some intermediate concepts, and combine these concepts with simple functions, to make the overall model more explainable.

While we focused on forest mapping at the treeline ecotone with aerial imagery, the methodology proposed here can be adapted to forest mapping in other ecotones, or to other types of remote sensing imagery. More importantly, it can be transferred to any other land cover mapping task involving class definitions or annotation instructions relying on concepts that can be quantified, allowing to leverage this prior knowledge to make the model more explainable.

CRediT authorship contribution statement

Thiên-Anh Nguyen: Conceptualization, Methodology, Software, Investigation, Writing – original draft. **Benjamin Kellenberger:** Conceptualization, Writing – review & editing. **Devis Tuia:** Conceptualization, Supervision, Writing – review & editing.

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

The authors do not have permission to share data.

Acknowledgments

The authors would like to thank Swisstopo, and more specifically Tobias Kellenberger and Theo Sautebin, for advice and support regarding the SwissImage and SwissTLM3D datasets. We would also like to thank Christian Ginzler and Lars Waser from the Swiss Federal Institute for Forest, Snow and Landscape Research (WSL), for advice and provision of the NFI VHM and the NFI plot data. Lastly, we would like to thank Jayshri Mizeret-Lad for writing advice and proof-reading of some of the sections of this paper, and the anonymous reviewers for their insightful comments.

Appendix. Ablation experiments

A.1. Use of the digital elevation model as input

In order to assess the benefit of using a DEM as additional input to the forest mapping model, we train a model similar to the BB model, which uses aerial images as the only input. This model yields lower segmentation scores than the BB model using both aerial images and the DEM (Table A.14), in particular for the SF class. The visual extracts displayed in Fig. A.18 suggest that one of the reasons for this performance loss is a higher number of false positive forest predictions above the treeline. We hypothesize that the model uses high altitude values as an indicator of low probability of presence of forest, and topography in general as a useful covariate for shrub forest mapping.

A.2. Hierarchy of the segmentation tasks

We train a model by framing the segmentation task as a 4-classes task (NF, OF, CF, SF). We use a single-term cross-entropy loss with class-specific weights that are inversely proportional to the class frequencies, instead of the 2-term loss function of Eq. (6).

We observe a slight decrease of the performance metrics compared to the model trained in the 2-task configuration (Table A.15). The confusion matrices displayed in Fig. A.19 reveal that by using the 2-task configuration, the amount of OF and SF pixels wrongly predicted as NF decreases noticeably, even though it remains highdecreases. This is clearly visible in Fig. A.20, where an example of predictions of the BB model with a single-task and a 2-task configuration is shown. Since the textures of OF, CF and SF are similar, we hypothesize that merging these 3 classes into a F class is a useful hint for the model that helps predicting better forest/non-forest boundaries.

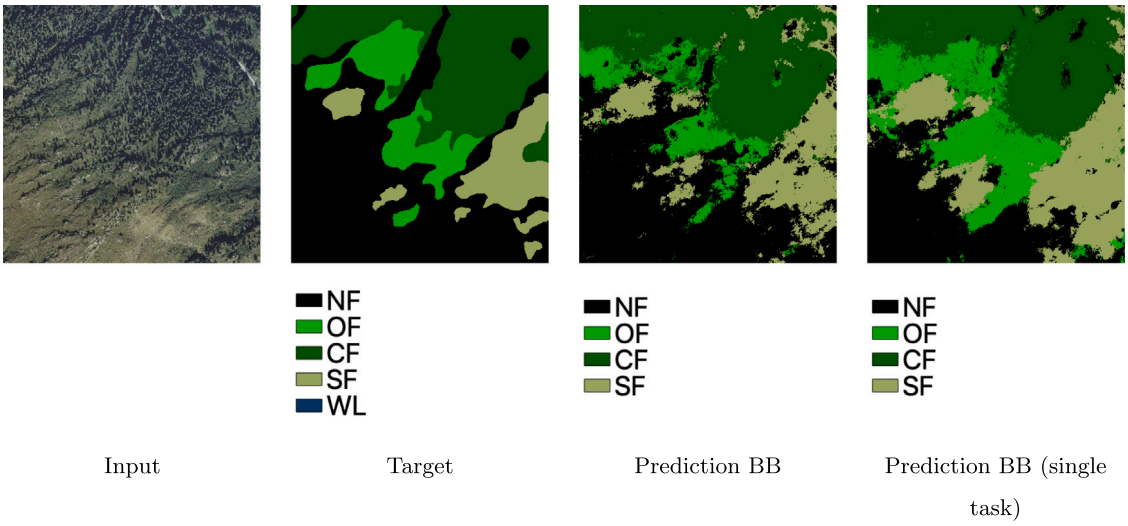


Fig. A.20. Visual extracts of the forest mapping results with the BB model with and without using a task hierarchy.

Table A.14
Per-class and averaged IoU and F-1 scores on the test set for the BB model with and without a DEM as input.

	Forest type								Forest presence/absence					
	OF		CF		SF		Average		NF		F		Average	
	IoU	f-1	IoU	f-1	IoU	f-1	IoU	f-1	IoU	f-1	IoU	f-1	IoU	f-1
BB	**0.34**	**0.51**	**0.87**	**0.93**	**0.47**	**0.64**	**0.56**	**0.69**	**0.91**	**0.95**	**0.80**	**0.89**	**0.85**	**0.92**
BB without DEM	0.33	0.49	0.86	0.92	0.40	0.57	0.53	0.66	0.90	**0.95**	0.77	0.87	0.83	0.91

Table A.15
Per-class and averaged IoU and F-1 scores on the test set for the BB model with and without using a task hierarchy.

	NF		OF		CF		SF		Average	
	IoU	f-1	IoU	f-1	IoU	f-1	IoU	f-1	mIoU	f-1
BB	**0.91**	**0.95**	**0.19**	**0.33**	**0.78**	**0.89**	**0.25**	**0.40**	**0.54**	**0.64**
BB (single task)	0.90	**0.95**	**0.19**	0.31	0.76	0.86	0.24	0.39	0.52	0.63

References

Al-Shedivat, M., Dubey, A., Xing, E., 2020. Contextual explanation networks. J. Mach. Learn. Res. 21, 1–44, https://github.com/alshedivat/cen. arXiv:1705.10301.

Alvarez-Melis, D., Jaakkola, T.S., 2018. Towards robust interpretability with self-explaining neural networks. In: Advances in Neural Information Processing Systems. pp. 7775–7784, http://arxiv.org/abs/1806.07538[arXiv:1806.07538].

Bengio, Y., Louradour, J., Collobert, R., Weston, J., 2009. Curriculum learning. In: Proceedings of the 26th Annual International Conference on Machine Learning. pp. 41–48.

Bolton, D.K., Coops, N.C., Hermosilla, T., Wulder, M.A., White, J.C., 2018. Evidence of vegetation greening at alpine treeline ecotones: three decades of landsat spectral trends informed by lidar-derived vertical structure. Environ. Res. Lett. 13, 084022. http://dx.doi.org/10.1088/1748-9326/AAD5D2, https://iopscience.iop.org/article/10.1088/1748-9326/aad5d2.

Boyd, D.S., Danson, F.M., 2005. Satellite remote sensing of forest resources: Three decades of research development. https://journals.sagepub.com/doi/abs/10.1191/0309133305pp432ra. http://dx.doi.org/10.1191/0309133305pp432ra.

Brändli, U.B., Abegg, M., Leuch, Barbara Allgaier, 2020. Inventaire Forestier National Suisse. Réultats Du Quatrième Inventaire 2009-2017. Technical Report, Swiss Federal Institute for Forest, Snow and Landscape Research (WSL), https://www.lfi.ch/publikationen/publ/LFI4_Ergebnisbericht-fr.pdf. https://www.lfi.ch/publikationen/publ/ergebnisberichte/lfi4-fr.php.

Cao, K., Zhang, X., 2020. An improved res-unet model for tree species classification using airborne high-resolution images. Remote Sens. 12 (2020), 1128. http://dx.doi.org/10.3390/RS12071128, https://www.mdpi.com/2072-4292/12/7/1128.

Chazdon, R.L., Brancalion, P.H., Laestadius, L., Bennett-Curry, A., Buckingham, K., Kumar, C., Moll-Rocek, J., Vieira, I.C.G., Wilson, S.J., 2016. When is a forest a forest? Forest concepts and definitions in the era of forest and landscape restoration. Ambio 45, 538–550. http://dx.doi.org/10.1007/s13280-016-0772-y.

Chu, Z., Tian, T., Feng, R., Wang, L., 2019. Sea-land segmentation with res-unet and fully connected CRF. In: International Geoscience and Remote Sensing Symposium (IGARSS) 2019-January. pp. 3840–3843. http://dx.doi.org/10.1109/IGARSS.2019.8900625.

Coops, N.C., Morsdorf, F., Schaepman, M.E., Zimmermann, N.E., 2013. Characterization of an alpine tree line using airborne LiDAR data and physiological modeling. Global Change Biol. 19, 3808–3821. http://dx.doi.org/10.1111/GCB.12319, https://onlinelibrary.wiley.com/doi/full/10.1111/gcb.12319.

Food and Agriculture Organization of the United Nations, 2020. Terms and Definitions, Forest Resources Assessment Working Paper 188. Technical Report, https://www.fao.org/3/I8661EN/i8661en.pdf.

Gazzea, M., Aalhus, S., Kristensen, L.M., Ozguven, E.E., Arghandeh, R., 2021. Automated 3D Vegetation Detection Along Power Lines using Monocular Satellite Imagery and Deep Learning. Institute of Electrical and Electronics Engineers (IEEE), pp. 3721–3724. http://dx.doi.org/10.1109/igarss47720.2021.9554938.

Ginzler, C., Hobi, M.L., 2015. Countrywide stereo-image matching for updating digital surface models in the framework of the swiss national forest inventory. Remote Sens. 7, 4343–4370. http://dx.doi.org/10.3390/rs70404343, http://www.mdpi.com/2072-4292/7/4/4343.

Harsch, M.A., Bader, M.Y., 2011. Treeline form - a potential key to understanding treeline dynamics. Global Ecol. Biogeogr. 20, 582–596. http://dx.doi.org/10.1111/j.1466-8238.2010.00622.x, https://onlinelibrary.wiley.com/doi/full/10.1111/j.1466-8238.2010.00622.x.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition. pp. 770–778. http://dx.doi.org/10.1109/CVPR.2016.90, arXiv:1512.03385.

Hill, R.A., Granica, K., Smith, G.M., Schardt, M., 2007. Representation of an alpine treeline ecotone in SPOT 5 HRG data. Remote Sens. Environ. 110, 458–467. http://dx.doi.org/10.1016/j.rse.2006.11.031.

Hu, Z., Ma, X., Liu, Z., Hovy, E., Xing, E.P., 2016. Harnessing deep neural networks with logic rules. In: 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016 - Long Papers, Association for Computational Linguistics. ACL, pp. 2410–2420, https://arxiv.org/abs/1603.06318v6. http://dx.doi.org/10.18653/v1/p16-1228. arXiv:1603.06318.

Kattenborn, T., Leitloff, J., Schiefer, F., Hinz, S., 2021. Review on convolutional neural networks (CNN) in vegetation remote sensing. http://dx.doi.org/10.1016/j.isprsjprs.2020.12.010.

Leboeuf, A., Fournier, R.A., Luther, J.E., Beaudoin, A., Guindon, L., 2012. Forest attribute estimation of northeastern Canadian forests using QuickBird imagery and a shadow fraction method. Forest Ecol. Manag. 266, 66–74. http://dx.doi.org/10.1016/j.foreco.2011.11.008.

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444.

Letham, B., Rudin, C., McCormick, T.H., Madigan, D., 2015. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. Ann. Appl. Stat. 9, 1350–1371. http://dx.doi.org/10.1214/15-AOAS848, arXiv:1511.01644.

Levering, A., Marcos, D., Tuia, D., 2021. On the relation between landscape beauty and land cover: A case study in the U.K. at sentinel-2 resolution with interpretable AI. ISPRS J. Photogramm. Remote Sens. 177, 194–203. http://dx.doi.org/10.1016/j.isprsjprs.2021.04.020.

Lloyd, S.P., 1982. Least squares quantization in PCM. IEEE Trans. Inform. Theory 28, 129–137. http://dx.doi.org/10.1109/TIT.1982.1056489.

Losch, M., Fritz, M., Schiele, B., 2019. Interpretability beyond classification output: Semantic bottleneck networks. http://arxiv.org/abs/1907.10882, arXiv:1907.10882.

Loshchilov, I., Hutter, F., 2019. Decoupled weight decay regularization. https://arxiv.org/abs/1711.05101v3. arXiv:1711.05101.

Luo, G., Dai, L., 2013. Detection of alpine tree line change with high spatial resolution remotely sensed data. J. Appl. Remote Sens. 7, 073520. http://dx.doi.org/10.1117/1.jrs.7.073520, https://www.spiedigitallibrary.org/terms-of-use.

Ma, L., Liu, Y., Zhang, X., Ye, Y., Yin, G., Johnson, B.A., 2019. Deep learning in remote sensing applications: A meta-analysis and review. ISPRS J. Photogramm. Remote Sens. 152, 166–177. http://dx.doi.org/10.1016/j.isprsjprs.2019.04.015.

Malkin, K., Robinson, C., Jojic, N., High-Resolution Land Cover Change from Low-Resolution Labels: Simple Baselines for the 2021 IEEE GRSS Data Fusion Contest. Technical Report, https://github.com/calebrob6/, arXiv:2101.01154v1.

Marcos, D., Fong, R., Lobry, S., Flamary, R., Courty, N., Tuia, D., 2021. Contextual semantic interpretability. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer Science and Business Media Deutschland GmbH, pp. 351–368. http://dx.doi.org/10.1007/978-3-030-69538-5_22, https://arxiv.org/abs/2009.08720v1. arXiv:2009.08720.

Montavon, G., Samek, W., Müller, K.R., 2018. Methods for interpreting and understanding deep neural networks. http://dx.doi.org/10.1016/j.dsp.2017.10.011. arXiv:1706.07979.

Morley, P.J., Donoghue, D.N., Chen, J.C., Jump, A.S., 2019. Quantifying structural diversity to better estimate change at mountain forest margins. Remote Sens. Environ. 223, 291–306. http://dx.doi.org/10.1016/j.rse.2019.01.027.

Okajima, Y., Sadamasa, K., 2019. Deep neural networks constrained by decision rules. In: 33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence. EAAI 2019, AAAI Press, pp. 2496–2505. http://dx.doi.org/10.1609/aaai.v33i01.33012496.

Ørka, H.O., Wulder, M.A., Gobakken, T., Næsset, E., 2012. Subalpine zone delineation using LiDAR and landsat imagery. Remote Sens. Environ. 119, 11–20. http://dx.doi.org/10.1016/J.RSE.2011.11.023.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. Pytorch: An imperative style, high-performance deep learning library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, Vol. 32. Curran Associates, Inc. pp. 8024–8035, http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Popescu, S.C., Wynne, R.H., 2004. Seeing the trees in the forest: Using lidar and multispectral data fusion with local filtering and variable window size for estimating tree height. http://dx.doi.org/10.14358/PERS.70.5.589.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., Prabhat, 2019. Deep learning and process understanding for data-driven earth system science. Nature 566, 195–204. http://dx.doi.org/10.1038/s41586-019-0912-1, https://www.nature.com/articles/s41586-019-0912-1.

Ribeiro, M.T., Singh, S., Guestrin, C., 2016. Why should i trust you? explaining the predictions of any classifier. In: NAACL-HLT 2016-2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Demonstrations Session. Association for Computing Machinery, pp. 97–101, https://arxiv.org/abs/1602.04938v3. http://dx.doi.org/10.18653/v1/n16-3020, arXiv:1602.04938.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-net: Convolutional networks for biomedical image segmentation. In: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer Verlag, pp. 234–241. http://dx.doi.org/10.1007/978-3-319-24574-4_28, http://lmb.informatik.uni-freiburg.de/http://lmb.informatik.uni-freiburg.de/people/ronneber/u-net. arXiv:1505.04597.

Roscher, R., Bohn, B., Duarte, M.F., Garcke, J., 2020. Explainable machine learning for scientific insights and discoveries. IEEE Access 8, 42200–42216. http://dx.doi.org/10.1109/ACCESS.2020.2976199, arXiv:1905.08883.

Samek, W., Wiegand, T., Müller, K.R., 2017. Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. http://arxiv.org/abs/1708.08296. arXiv:1708.08296.

Siddique, N., Paheding, S., Elkin, C.P., Devabhaktuni, V., 2021. U-Net and Its Variants for Medical Image Segmentation: A Review of Theory and Applications. IEEE Access, http://dx.doi.org/10.1109/ACCESS.2021.3086020.

Simonyan, K., Vedaldi, A., Zisserman, A., 2014. Deep inside convolutional networks: Visualising image classification models and saliency maps. https://arxiv.org/abs/1312.6034v2. arXiv:1312.6034.

Song, X.P., Hansen, M.C., Stehman, S.V., Potapov, P.V., Tyukavina, A., Vermote, E.F., Townshend, J.R., 2018. Global land change from 1982 to 2016. Nature 560, 639–643. http://dx.doi.org/10.1038/s41586-018-0411-9.

Stomberg, T., Weber, I., Schmitt, M., Roscher, R., 2021. Jungle-net: Using explainable machine learning to gain new insights into the appearance of wilderness in satellite imagery. In: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences. pp. 317–324. http://dx.doi.org/10.5194/isprs-annals-V-3-2021-317-2021.

Swisstopo, 2021a. Orthoimages. Online https://www.swisstopo.admin.ch/en/geodata/images/ortho/swissimage10.html. (Accessed 21 December 2021).

Swisstopo, 2021b. SwissTLM3D. Online https://www.swisstopo.admin.ch/en/geodata/landscape/tlm3d.html. (Accessed 21 December 2021).

Swisstopo, 2021c. Catalogue des objets swisstlm3d 1.9. https://www.swisstopo.admin.ch/fr/geodata/landscape/tlm3d.html#dokumente.

Swisstopo, 2021d. swissALTI3D. Online https://www.swisstopo.admin.ch/fr/geodata/height/alti3d.html. (Accessed 21 December 2021).

Tuia, D., Roscher, R., Wegner, J.D., Jacobs, N., Zhu, X.X., Camps-Valls, G., 2021. Towards a collective agenda on AI for earth science data analysis. IEEE Geosci. Remote Sens. Mag. 9, 88–104.

Wagner, F.H., Sanchez, A., Tarabalka, Y., Lotte, R.G., Ferreira, M.P., Aidar, M.P., Gloor, E., Phillips, O.L., Aragão, L.E., 2019. Using the U-net convolutional network to map forest types and disturbance in the atlantic rainforest with very high resolution images. Remote Sens. Ecol. Conserv. 5, 360–375. http://dx.doi.org/10.1002/rse2.111, https://zslpublications.onlinelibrary.wiley.com/doi/full/10.1002/rse2.111.

Waser, L.T., Boesch, R., Wang, Z., Ginzler, C., 2017. Towards automated forest mapping. In: Mapping Forest Landscape Patterns. Springer, New York, pp. 263–304. http://dx.doi.org/10.1007/978-1-4939-7331-6_7, https://link.springer.com/chapter/10.1007/978-1-4939-7331-6_7.

Waser, L.T., Rüetschi, M., Psomas, A., Small, D., Rehush, N., 2021. Mapping dominant leaf type based on combined sentinel-1/-2 data – challenges for mountainous countries. ISPRS J. Photogramm. Remote Sens. 180, 209–226. http://dx.doi.org/10.1016/j.isprsjprs.2021.08.017, https://linkinghub.elsevier.com/retrieve/pii/S0924271621002239.

White, J.C., Coops, N.C., Wulder, M.A., Vastaranta, M., Hilker, T., Tompalski, P., 2016. Remote sensing technologies for enhancing forest inventories: A review. http://dx.doi.org/10.1080/07038992.2016.1207484, https://www.tandfonline.com/doi/full/10.1080/07038992.2016.1207484.

Xiao, X., Lian, S., Luo, Z., Li, S., 2018. Weighted res-unet for high-quality retina vessel segmentation. In: Proceedings - 9th International Conference on Information Technology in Medicine and Education. ITME 2018, pp. 327–331. http://dx.doi.org/10.1109/ITME.2018.00080.

Ye, T., Wang, X., Davidson, J., Gupta, A., 2018. Interpretable intuitive physics model. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), pp. 89–105. http://dx.doi.org/10.1007/978-3-030-01258-8_6, arXiv:1808.10002.

Zeiler, M.D., Fergus, R., 2014. Visualizing and understanding convolutional networks. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer, Cham, pp. 818–833. http://dx.doi.org/10.1007/978-3-319-10590-1_53, https://link.springer.com/chapter/10.1007/978-3-319-10590-1_53. arXiv:1311.2901.

Zhu, X.X., Tuia, D., Mou, L., Xia, G.S., Zhang, L., Xu, F., Fraundorfer, F., 2017. Deep learning in remote sensing: a comprehensive review and list of resources. http://dx.doi.org/10.1109/MGRS.2017.2762307.