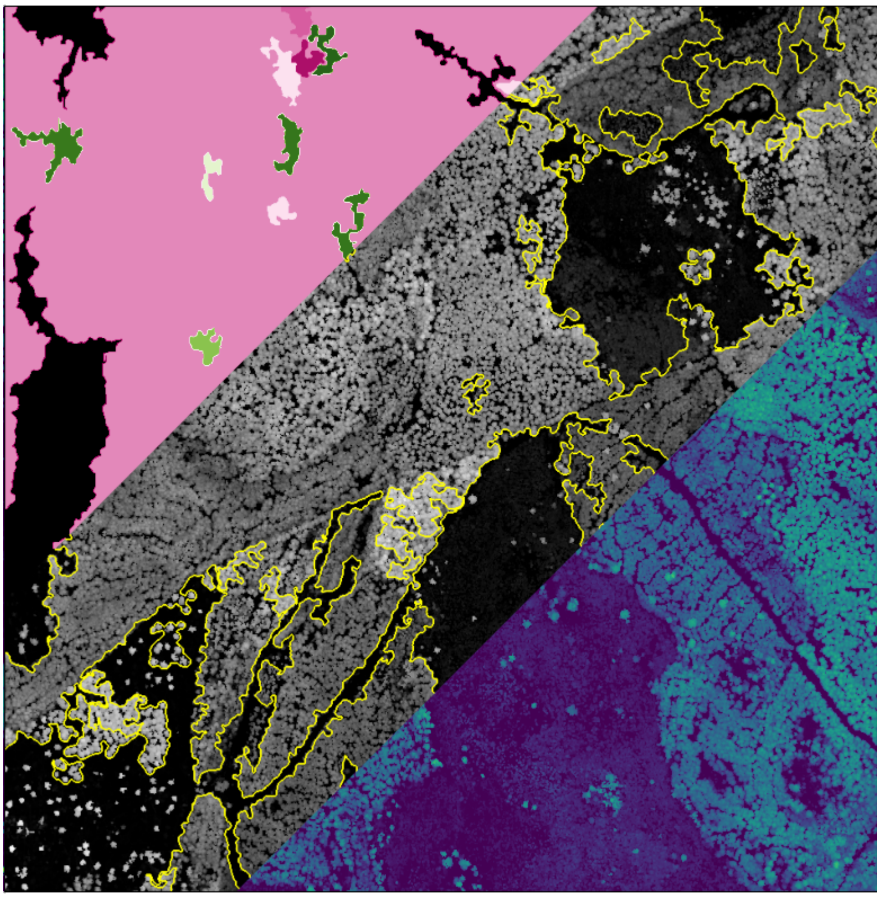


Interpretable segmentation and naturalness classification of forests using the Canopy Height Model

Master's thesis in Complex Adaptive Systems

OSCAR STÅLNACKE

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024
www.chalmers.se

MASTER'S THESIS 2024

Interpretable segmentation and naturalness classification of forests using the Canopy Height Model

OSCAR STÅLNACKE

Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
Applied Artificial Intelligence
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2024

Interpretable segmentation and naturalness classification of forests using the Canopy Height Model
Oscar Stålnacke

Examiner and Supervisor:
Marco L. Della Vedova, Department of Mechanics and Maritime Sciences

Master's Thesis 2024
Department of Mechanics and Maritime Sciences
Division of Vehicle Engineering and Autonomous Systems
Applied Artificial Intelligence
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover image — in order from top-left to bottom-right: predicted naturalness color-coded from saturated red to saturated green to represent degree of naturalness, boundaries of segments as produced by SLIC, the Canopy Height Model.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2024

Interpretable segmentation and naturalness classification of forests using the Canopy Height Model
OSCAR STÅLNACKE
Department of Mechanics and Maritime Sciences
Chalmers University of Technology

Abstract

As increasing amounts of ecological data from remote sensing is available, so does the opportunities to utilize this data also increase. One such opportunity is to classify the naturalness of forests automatically from data instead of using costly field-surveys, in the purpose of more efficiently managing natural ecosystems. Doing this using AI-based methods in an **interpretable** way is the topic of this thesis. Interpretability is of utmost importance in order to improve testing and verifying of models as well as to provide transparency and understanding to affected parties, such as forestry companies.

Using the Canopy Height Model (CHM), i.e. laser-scanned height data, as single source of data, three segmentation algorithms are evaluated on the task of creating segments of forest from CHM-rasters. A lack of ground-truth data in the evaluation is overcome by using human feedback as preference-based pairwise comparisons to evaluate and optimize. A framework to collect such data has been developed in the shape of a web application to assure convenience and efficiency in collection of data. Based on this data, one superior algorithm is selected and further optimized, to then be applied in tandem with a previously developed naturalness classifier in order to map the naturalness of forests.

The result is a CHM–into–naturalness pipeline with a 75% naturalness classification accuracy, evaluated on a 50×50 km area in Southern Sweden, only performing segmentation and classification on 640×640 m tiles of the CHM at a time. This method is shown to have both big potential but also some vulnerabilities, among them being the tile sizes, and remedies are suggested and discussed. A certain mismatch in how the model was trained and how it is now evaluated introduces some uncertainty to the evaluation, and possible improvements are discussed. In summary, the thesis lays the ground-work for the topic and suggests important aspects to iterate upon, giving a direction for future research.

Keywords: interpretable, AI, forest, segmentation, preference-based, pairwise comparison, naturalness, classification.

Acknowledgements

Big thanks to my supervisor (and examiner) Marco L. Della Vedova for his guidance and support throughout the project, as well as several thoughtful and fruitful discussions, ultimately leading to a better thesis in terms of code, report, and project as a whole. Thanks also to Andreas Bauner[1] for providing access to his code for transforming polygon-based georeferenced data into pixel-wise raster-based data.

Oscar Stålnacke, Gothenburg, 2024

[1]baunera@student.chalmers.se

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis listed in no particular order:

CHM	Canopy Height Model
RLHF	Reinforcement Learning with Human Feedback
AI	Artificial Intelligence
ML	Machine Learning
RKHS	Reproducing Kernel Hilbert Space
UI	User Interface
UX	User Experience
SLIC	Simple Linear Iterative Clustering
TP	True Positive
FP	False Positive
TN	True Negative
FN	False Negative
QP	Quadratic Programming
WKT	Well-Known Text
RAG	Region Adjacency Graph

Contents

List of Figures

List of Tables

1 Introduction

Artificial Intelligence (AI)-based technologies together with increasing availability of ecological data play an increasingly significant role in ecological research and there is a growing interest in applying AI-based methods to take advantage of this data. Automatically evaluating the naturalness of forests in an interpretable way is one such application. Doing this in an interpretable way is of major interest as interpretability provides transparency into decisions as well as an elevated ability of understanding and validating the AI models, compared to "black-box" models. The knowledge of a forests naturalness is important in order to address environmental challenges, maintain biodiversity, and in order to preserve and manage natural ecosystems (specifically forests in this case), among others. National authorities such as Sweden's *Skogsstyrelsen*[1] and *Naturvårdsverket* [2] currently have methods based on field surveys to asses the naturalness, where these are based on factors such as indicator species, characteristics of forests and trees and past forest management strategies. This requires large amounts of time and resources, and introducing automatic evaluation from data gathered through remote sensing will streamline the process and potentially open doors for a more standardized, objective method of measurement.
Remote sensing has become an increasingly important source of data in the context of forest ecology and forest management [26], which presents the opportunity to apply AI-based methods on such data, in this case to evaluate the naturalness of forests. AI techniques have proved to be an excellent tool applied on tasks of analyzing remote sensing data [52], where a few examples are estimating forest biomass [20], and integrating AI into drone-based applications within forestry [6].
However, the topic of interpretably assessing the naturalness of forests is not as well explored, but recent work part of the TolkAI[3] research project has created a simple and interpretable Machine Learning (ML) approach to classify the naturalness of forests using the Canopy Height Model (CHM) as single source of data. This model classifies forests based on a group of features such as the height of trees and the tree density [27]. The goal is for this naturalness classifier to be part of a larger pipeline taking as input a larger area of interest (e.g. all forests of one municipality) and giving as output a naturalness map of sorts. The missing stage of this pipeline is logic to split an area of interest into sensible, smaller segments of forest to then pass on to the classifier. This can be seen as an image segmentation task, and a particularly difficult one at that, as the desired segmentation should provide segments of forest

[1] `https://www.skogsstyrelsen.se/`
[2] `https://www.naturvardsverket.se/`
[3] `https://research.chalmers.se/en/project/10951`

that make sense to individually evaluate the naturalness on, which is difficult to translate into a segmentation algorithm. Image segmentation is a typical task where neural networks are dominant, but in order to circumvent a lack of ground-truth data and keep the method interpretable, only classical methods such as graph-based algorithms [14], region-growing and region-merging algorithms [21, 45] and watershed algorithms [17, 18, 4, 3] will be considered.
The aim of this thesis can be summarized as two major points:

- Evaluate and optimize image segmentation algorithms on top of the CHM to work well with forest segmentation in mind.
- Integrate the forest segmentation with the naturalness classifier in order to classify each segment of forest, and evaluate the entire pipeline from CHM to naturalness map.

1.1 Overview of thesis

The rest of this report will be structured as following: Chapter 2 will cover related work in the context of forest segmentation along with theory and background information that is relevant for this thesis. Then Chapter 3 will cover the methodology, wherein Section 3.1 covers adaptions/modifications to the segmentation algorithms, Section 3.2 covers the way in which the segmentation was evaluated, Section 3.3 how the segmentation was optimized, and Section 3.4 covers the integration with the naturalness evaluator. Chapter 4 will cover all results generated and discuss its implications. The report then ends with a conclusive Chapter 5 which summarizes the findings and briefly discusses interesting future work ahead of this thesis.

2

Background

This chapter will cover most of the relevant theory and background information necessary in order to follow the remainder of the thesis. Section 2.1 will cover related work specifically in the context of forest segmentation from a general perspective, while Section 2.2 will cover theory with regard to the set of classical segmentation algorithms being used in this thesis. Section 2.3 finally covers methods of evaluation.

2.1 Related work in forest segmentation

There is a multitude of different approaches to segmenting forests and the issue of segmentation in the context of remote sensing in general. There are some methods which rely on only the CHM (Canopy Height Model), such as generating a coarse segmentation using a mean-shift algorithm followed by refining the segmentation with a superpixel-based region growing [50], merging triples of trees based on their crown area and the area of the convex hull of the triple and thus building a forest mask [13], creating Voronoi cells based on local maxima of the CHM (i.e. approximately tree-tops) and merging cells based on certain criteria [29], and a Hierarchical Watershed Transform to segment the CHM at several scales [53]. There are also papers that present methods which use both laser-scanned height data as well as multi-spectral image data [11, 28, 10], or using only aerial images [47, 48, 41]. More recent papers also suggest methods to efficiently handle segmentation of high-resolution and large raster data [9, 10], e.g. by essentially decreasing the resolution by looking at superpixels generated by Simple Linear Iterative Clustering (SLIC) [1] instead of the original pixel resolution [9].

2.2 Segmentation algorithms

The three segmentation algorithms that will be considered during the thesis are: watershed by flooding, SLIC (Simple Linear Iterative Clustering) superpixel-based segmentation, and lastly, Felzenszwalb's graph-based segmentation algorithm. This section will cover the theory behind these three.

2.2.1 Watershed

The *watershed* in the watershed algorithm is inspired by the elevated terrain which separates water basins in nature, which is commonly called a watershed [12]. Let us use this analogy in order to get an intuitive understanding of watershed instead of covering algorithmic details (there are many algorithms to perform watershed segmentation, but the general idea is the same). The algorithm works by first treating a (single-channel) image as a terrain map, then marking a set amount of points in the image as “water sources”, let us call them *markers*. The algorithm then proceeds to fill n water basins from these sources, and each time two different basins meet, that location is considered a boundary between segments. This continues until every pixel belongs to a water basin, and at the end this will result in an image which has been separated into n segments. Watershed has been successfully applied to the CHM for forest segmentation with a Hierarchical Watershed Transform [53], and for images in general using a standard watershed followed by region merging or region growing [4, 18].

2.2.2 SLIC

Superpixel algorithms is a popular concept and key building block in many computer vision and image processing algorithms that groups pixels into meaningful and roughly uniform regions. This can then be used instead of the pixel grid to perform subsequent image processing tasks efficiently on as it effectively reduces the complexity and resolution of the image. SLIC is a version of k-means clustering modified to handle superpixel generation, and is proven to be computationally more efficient, consume less memory and achieve higher or equal segmentation quality compared to other superpixel algorithms [1]. SLIC is typically applied to multi-channel images, but with a modified selection of parameters it can also be applied to single-channel data. SLIC has previously been used for forest stand delineation and segmentation of remote sensing data [50, 9] and has the potential to be applied to perform efficient and fast segmentation on high resolution data [9].

2.2.3 Felzenszwalb

Felzenszwalb’s algorithm is a graph-based segmentation algorithm which is based on the idea that nodes, in this case pixels of the raster, in the same component or region should be similar, while pixels in different regions should be dissimilar. This is achieved by merging regions or marking boundaries based on pairwise comparisons between regions using their internal differences and differences between regions. The internal difference is based on the maximum weight of the Minimum Spanning Tree (MST) of the region, while the differences between regions are based on the minimum weights of edges between regions. This algorithm is supposed to avoid both under- and over-segmentation. See Felzenszwalb’s paper [14] for details.

2.3 Segmentation Evaluation

When evaluating segmentations of images there is a plethora of methods to consider. One of the most convenient and familiar methods, especially in the context of neural-network based segmentation, is supervised segmentation by comparing with ground-truth segmentations. But there are many more approaches to choose from, e.g. unsupervised evaluation where human-interpretable metrics are calculated, subjective evaluation where humans evaluate the segmentations, or even neural-network based evaluation methods [46]. In this case of segmenting remote sensing data, more specifically segmentation of forests, there is no ground-truth reference segmentations to compare with available for the geography being considered, and creating sufficient volumes of such data is unfeasible, thus leaving supervised evaluation out of the question. This leaves some sort of unsupervised evaluation or evaluation by human preference. There are applications of unsupervised evaluation in remote sensing, but many of them rely upon measuring homogeneity within segments and heterogeneity between segments [46], which is likely ill-fitted given that all segmentation methods in this thesis utilize homo- and heterogeneity conditions in one way or another. Thus what is left is subjective evaluation. This however should not prove to be any hindrance to the quality of the evaluation as domain expertise from Skogsstyrelsen has been accessible.

Subjective evaluation and optimization by subjective evaluation (could also be called human-in-the-loop evaluation) can come in several shapes. One such method is reinforcement learning from human feedback (RLHF) [32], where human preferences are used to provide feedback and guide reinforcement learning algorithms [7, 2, 37]. RLHF is well-suited for applications where rewards are difficult to define for complex tasks. Most recently RLHF has been applied in the context of natural language processing and understanding [30, 39], where more specifically ChatGPT, Gemini [40], Sparrow [15] and Claude[1] are trained with the help of RLHF.

A common way to gather preferences from humans is by ranking outputs, where the most straightforward approach is to use pairwise (or k-wise) comparisons. These kinds of (mostly pairwise) preferential methods are present in many contexts, such as in creating a ranking from pairwise preferences [22, 19, 49], in performing Bayesian optimization [16, 8, 5] and in optimizing algorithm parameters using either evolutionary algorithms [23] or by approximating preference functions [24]. A/B testing could also be seen as evaluation by preference if for example trying to figure out what layout of a website end-users prefer [51]. Making decisions by pairwise preferential evaluation and optimization also has support by psychological research, see e.g. the law of comparative judgement [42, 25], and has been discussed in the context of decision theory [35, 36].

[1] `https://www.anthropic.com/news/claude-2-1`

3
Method

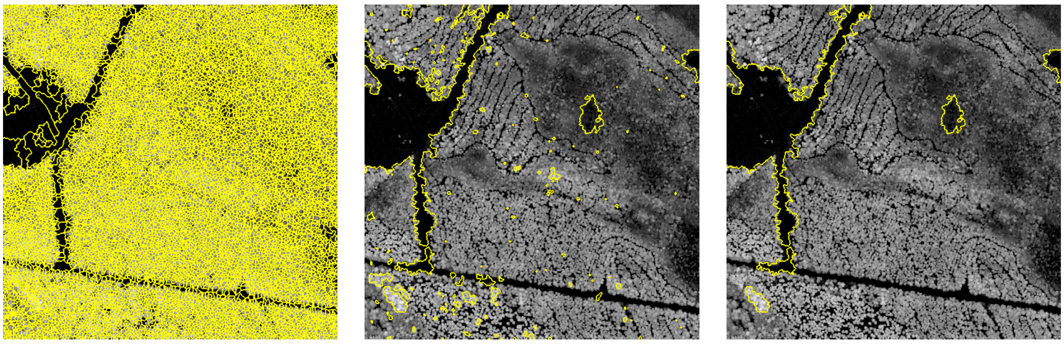
The task of segmenting a forest based on the Canopy Height Model (CHM) can be seen as an image segmentation task on a single-channel (gray-scale) image. Image segmentation is a vast field, especially including all the neural-network based segmentation methods. In the spirit of interpretability, but also in part because of a lack in ground-truth segmentation data, this thesis will only consider classical segmentation methods while also taking into account what is, and has, been used in the context of remote sensing and the CHM. The end goal of the segmentation is to find, adapt, and optimize existing methods to work well with the CHM.
Section 3.1 will initially cover further implementation details of the segmentation. Then Section 3.2 will cover the procedure in which the segmentation was evaluated, followed by Section 3.3 which covers the optimization of a segmentation algorithm. Lastly Section 3.4 will cover the methodology for applying and evaluating the performance of the segmentation in tandem with a naturalness evaluator.

3.1 Adapting the segmentation algorithms

Section 2.2 introduced the absolute basics of the segmentation algorithms with the purpose of getting an intuitive understanding of them. This section will further describe implementation specifics and any extra steps taken to adapt/improve the algorithm for the specific case of being applied on the CHM.

3.1.1 Watershed

For this thesis, the implementation of watershed segmentation in the Python-package scikit-image [43] was used. The parameters of importance, i.e. the only ones being considered during this thesis, are the *markers* and the *compactness*. The *markers*, while referring to 2.2.1, control where the water sources are set, and they are most reasonably set to the treetops of the CHM as they are local extrema of the forest. The treetops in turn are identified using a maximum-filter with a 3x3 kernel (using scipy's [44] implementation of a maximum filter on images). Also, given how the flooding works bottom-up, the CHM raster is first inverted, i.e. for every pixel p representing the tree height in the raster, $p := -p$, such that all trees are considered as inverted tree-shaped basins for the segmentation. The *compactness* parameter controls the general shape of each segment, where higher values result in more regularly shaped basins. The output from the watershed segmentation is a label-array with the same shape as the CHM raster, but where each pixel is given an integer label depending on which segment it belongs to. An example of the segmentation at this stage can be seen in Figure 3.1a.



(a) **(b)** **(c)**

Figure 3.1: Example of watershed segmentation on a 500x500 meter CHM raster using **a)** purely watershed segmentation with treetops as markers, **b)** watershed segmentation followed by region merging based on mean tree height, **c)** watershed segmentation followed by region merging and filtering of regions below a certain pixel area.

A common issue with watershed segmentation is over-segmentation, i.e. ending up with abundant amounts of segments (see Figure 3.1a for an extreme example). In the case of the method above, this is a prominent issue as there will exist approximately one segment per treetop, which results in thousands, or even tens of thousands of small segments, whereas the desired outcome is a handful of larger segments encapsulating regions of forest. One way to overcome this, and the method of choice in this thesis, is by merging regions based on similarity. There is no standard way to measure this similarity for the CHM, but given that the CHM represents height, and for the sake of simplicity, the similarity in mean tree height between segments was considered. The region merging was achieved by utilizing a Region Adjacency Graph (RAG). The RAG is built by creating one node for every unique label, i.e. n number of nodes, where each node has some associated features such as: the amount of pixels with that label and the mean tree height for each node/segment. The edges of the graph are calculated as the absolute value of the difference of the mean tree height between the two nodes connected by the edge. Edges only exist between segments which are directly connected. After this, the most similar pairs of nodes are greedily merged, i.e. the nodes connected by smallest edge always merge first, until no edges with a weight below a certain threshold remains. This threshold will in subsequent chapters be referred to as *merge_threshold*. When merging two nodes the features of the node and the weights of all connecting edges are updated accordingly. An example of the segmentation at this stage can be seen in Figure 3.1b.

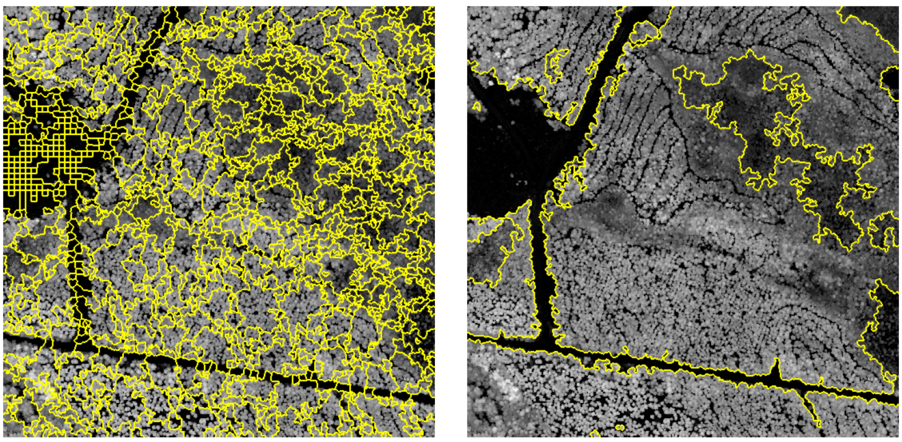
At this point, the amount of segments have drastically decreased, but there might still be segment-“specks” throughout the raster, a noise of sorts if you will. In order to get rid of these artifacts a third and final filtering step was introduced. This filtering works by looping over all nodes in the RAG (equivalent to all segments in the raster), and for every node not reaching a certain condition, merge the node with its largest neighbour. This filtering condition was chosen arbitrarily, and the filtering condition was chosen as any segment not surpassing $500m^2$, and the largest neighbour was chosen based on segment size in m^2. The watershed-based segmentation is now finished and the final label-image (or label-array) is returned as the final segmentation of the CHM. An example of the segmentation at this stage can be seen in Figure 3.1c. All variable parameters affecting the outcome of this segmentation is thus the *compactness* and *merge_threshold*.

3.1.2 SLIC

For this thesis the implementation of SLIC from scikit-image [43] was used. The parameters of importance for this implementation of the algorithm are:

- *compactness*, which works similarly to compactness in watershed by affecting the general shape of the segments. Higher values result in more uniform and cube-shaped segments, while lower values allow for more irregular shapes.
- *sigma*, controlling the width of the Gaussian kernel which is used in pre-processing to smooth the raster before segmentation.
- An approximate desired amount of segments in the output label array. The desired amount of segments will not be used directly in this thesis, and instead a variable, call it *segment_mult*, will be used. This variable is later used to scale the desired amount of segments with the size of the raster. In other words it is used to, in a raster-size-invariant way, determine an approximate superpixel size.

An example of only SLIC segmentation can be seen in Figure 3.2a.



(a) **(b)**

Figure 3.2: Example of SLIC segmentation on a 500x500 meter CHM raster using **a)** pure SLIC segmentation, **b)** SLIC segmentation followed by region merging based on mean tree height and filtering to remove tiny segments.

While SLIC on its own can act as a decent forest segmentation algorithm given very careful choices of parameters, it is not ideal as it is dependent on the CHM it is being applied on and is quite sensitive to changes in some of the parameters, making it unreliable when attempting to generate superpixels at forest-scale for a wide selection of CHMs. Instead, a more suitable approach is to use SLIC as it was intended, to create “large pixels” (superpixels) to reduce complexity and increase speed of subsequent tasks and computations. This was then followed up with the same approach as for watershed segmentation. That is, a region merging based on similarity followed by a filtering to remove noise in the segmentation before producing the final segmentation and label-image. All variable parameters affecting this segmentation are thus the *compactness*, *sigma*, *segment_mult* and *merge_threshold*.

3.1.3 Felzenszwalb

Scikit-image [43] also has an implementation of Felzenszwalb's algorithm, which was used here. The algorithm depends on three parameters of importance: the *scale* parameter which indirectly affects the amount of segments and the size of segments, a *sigma* which is the width of the Gaussian kernel used for smoothing in pre-processing, and lastly a minimum size parameter named *min_size*, which as the name suggest, determines the minimum size of segments this algorithm will produce. The idea of the minimum size parameter is similar to the filtering step in watershed and SLIC, in order to remove noise and avoid segments of undesirable sizes, i.e. sizes that generally do not make sense in the context of forests. This algorithm needs no additional steps and the labelled image directly from the algorithm is used as the final segmentation of the CHM. An example of this sort of segmentation can be seen in Figure 3.3.

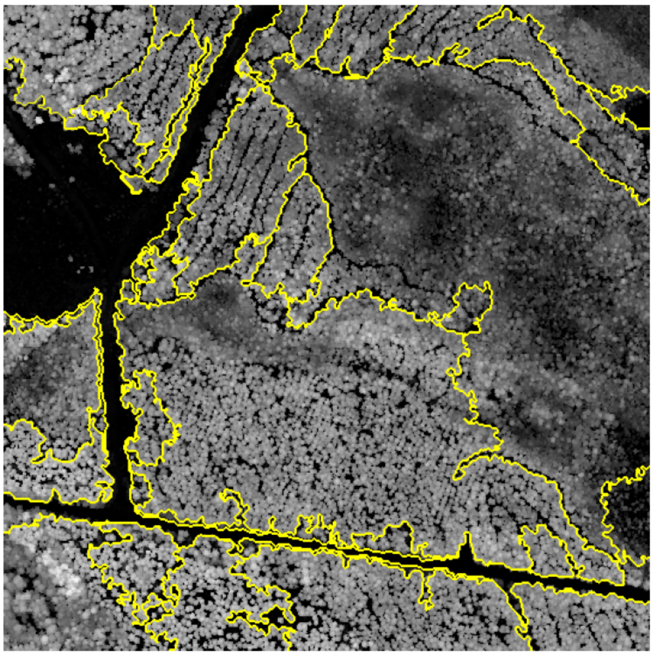


Figure 3.3: Example of Felzenszwalb segmentation on a 500x500 meter CHM raster.

3.2 Segmentation evaluation in practice

The evaluation and optimization is done somewhat in tandem for the segmentation algorithms in this thesis in order to successively decrease the complexity of the optimization and allow for simpler methods. The evaluation is based on pairwise preferences of outputs from the segmentation algorithms, and later on, data gathered by pairwise preference will be the foundation for optimizing a segmentation algorithm. The first stage of the evaluation is the "coarse" stage where the aim is to single out a sole superior segmentation algorithm. The second stage is a "fine" evaluation stage where the aim is to more rigorously evaluate and optimize a single algorithm.

3.2.1 Framework for gathering evaluations

As the method evaluation and optimization is subjective in nature, there is a need for some framework to collect this data, in this case from people at Skogsstyrelsen. The data collection framework of choice was a web-based User Interface (UI) created using *Streamlit*[1], a Python-framework to build data apps which efficiently turns simple Python scripts into web-based applications. The reasoning for using a web-based application for the data collection is very simple, it makes the process extremely convenient and efficient, both for the collector and the collectee. As for the choice of using specifically *Streamlit* the motivation comes mainly from two places: (i) it is already used in the context of the TolkAI project, (ii) it is used in the popular data annotation tool *Prodigy*[2].

The layout of the web application was very simple. It shows a pair of two different segmentations on top of the exact same CHM raster along with two primary buttons to submit which one is preferred. There was also a field for optional comments, and for part of the thesis there was also a secondary button to skip a pair in case of bad data and the option to view the raster without the segmentation. A screenshot of the web app can be seen in Figure 3.4.

Once the user submits a preference, i.e. clicks a button, any relevant data is saved for future analysis. The data being saved is:

- The segmentation algorithm and the parameter values for said algorithm for both the segmentations in the pair.
- Which of the segmentations was preferred.
- The geographical location of the raster.
- Other miscellaneous information, such as a timestamp and a session id to identify if the comparisons come from the same user and in which order.

The web app was hosted on a server at Chalmers and a CI/CD pipeline was set up using GitLab's CI/CD functionality. The pipeline consisted of running tests for a subset of the functionality and then automatically deploying the web app onto the server following a push to the main branch.

[1] `https://streamlit.io/`
[2] `https://prodi.gy/`

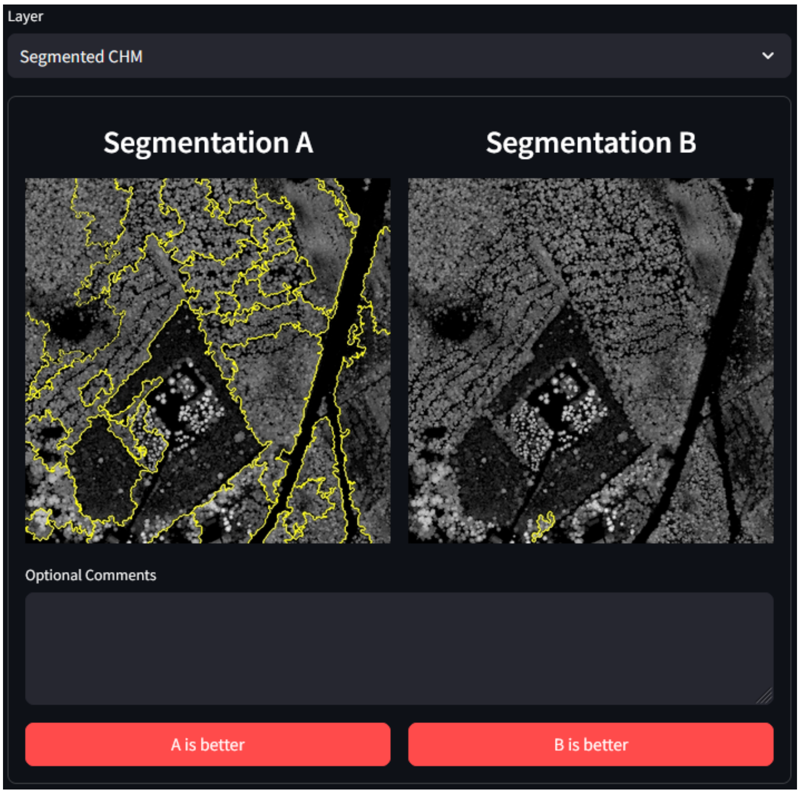


Figure 3.4: Screenshot of the web UI presented to users at Skogsstyrelsen in order to collect preference data.

3.2.2 First "coarse" stage

The purpose of the first stage is singling out one segmentation algorithm to proceed with in subsequent steps. This was done by utilizing pairwise preferences from domain experts at Skogsstyrelsen. The data-gathering flow was structured as following (see also Figure 3.5 for a more digestible overview):

1. Create a pre-defined and fixed set of 640×640 CHM rasters and call it S_{chm}. S_{chm} was created by randomly generating a larger set of rasters from all over Sweden and manually filtering out undesirable rasters. S_{chm} contained approximately 20 rasters. More details about how this will be covered in Chapter 4, along with Figure 4.1, where these rasters can be seen.
2. Create a pre-defined and fixed set of algorithm and parameter configurations called S_p. These were also generated randomly by creating a larger random set and manually filtering out some parameter sets. More details about this will also come during Chapter 4.
3. Fetch a raster r from S_{chm}.
4. From S_p, fetch a random subset k where k contains equally many configurations for each of the three segmentation algorithms, e.g. three of watershed, three of SLIC and three of Felzenszwalb.
5. Remove two random parameter sets p_1 and p_2 from k and compute two segmentations $s(p_1)$ and $s(p_2)$ on the raster r.
6. Show the segmentations $s(p_1)$ and $s(p_2)$ to the user.

7. If the user prefers $s(p_1)$, save necessary data from the comparison as described in Section 3.2.1, fetch a new p_2 from k and recalculate $s(p_2)$ and vice versa if user prefers $s(p_2)$.
8. Repeat step 7 until the set k is empty, meaning the user has roughly identified the best segmentation within the set k. Think of it a bit as a segmentation battle royale.
9. Start over from step 3.

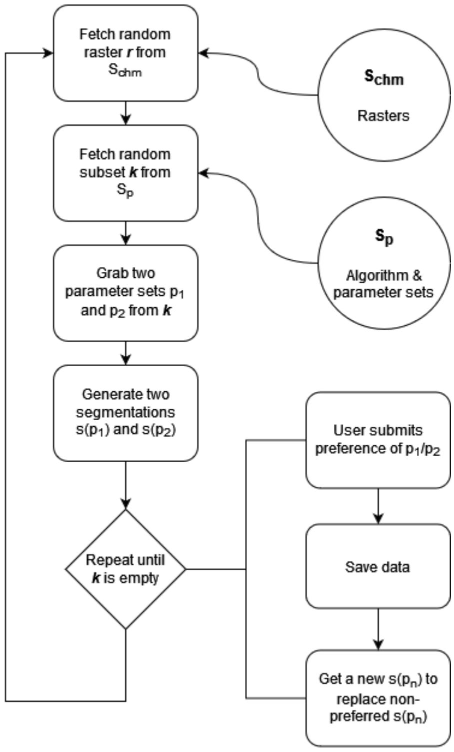


Figure 3.5: A non-detailed overview of the data-gathering flow as described in Section 3.2.2.

Except for the winner of each duel between two segmentations, the winner in each subset k can also be used to determine which of the three algorithms performs better than the other two, on average.

3.2.3 Second "fine" stage

The purpose of the second stage is to more rigorously evaluate the chosen segmentation algorithm and prepare it for optimization. This was done similarly to the first stage with the help of knowledge from Skogsstyrelsen and the *streamlit* web app. While the general flow is somewhat similar to the first stage, there are some meaningful differences, and the gathering of data was structured as following (see also Figure 3.6 for a visual overview):

1. Generate a geographical coordinate O, where O is a random coordinate inside a polygon encapsulating Sweden[3].

[3]https://www.naturalearthdata.com/downloads/110m-cultural-vectors/110m-admin-0-countries

2. Fetch a 640×640 CHM raster r centered at O.
3. Generate two parameters sets p_1 and p_2 randomly from the parameter space $\mathbf{P}$ of the segmentation algorithm being considered.
4. Generate two segmentations $s(p_1)$ and $s(p_2)$ on top of raster r.
5. Display the pair of segmentations to the user and wait for input.
6. Save the users preference and all other necessary data as described in Section 3.2.1 and restart from step 1.

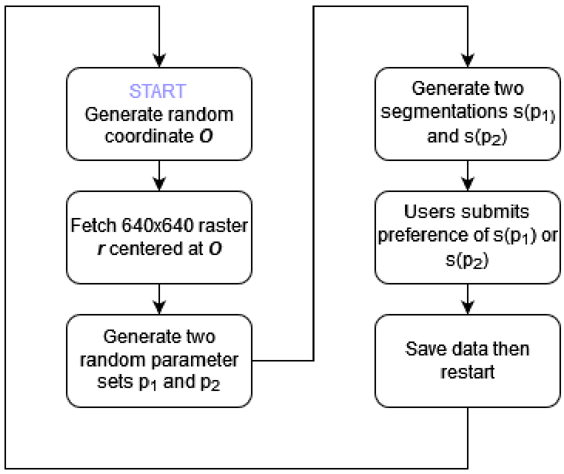


Figure 3.6: A non-detailed overview of the data-gathering flow as described in Section 3.2.3.

At this stage the user also had the option to skip the raster r in case the randomly generated raster was poorly chosen, along with the option to provide an optional comment if e.g. there was something notable about the comparisons that could not be summarized in having a preference for $s(p_n)$. The collected data from this evaluation was then used to optimize the parameters to the algorithm.

3.3 Segmentation optimization

The optimization was performed by following a procedure suggested by Kisilev and Freedman [24], that attempts to approximate the preference function $f(p)$, which attempts to model the users preference over parameter space, from pairwise comparisons. From the approximated function $f(p)$ it is then possible to follow standard optimization approaches, such as a random search, in order to acquire the optimal parameters p^*. What now follows is a terse overview of the theory for this algorithm with the purpose of giving a simplified way of following the steps required to arrive at this algorithm. This means many details will be skipped, so for the curious, please refer to the original paper [24].

When trying to model the user's preference as a function $f(p)$ over parameter-space $\mathbf{P}$, there are originally infinitely many functions to choose from. A reasonable first condition for the choice of function is the *smoothest* function, where the smoothest function is procured by minimizing the smoothness energy $S[f]$, which will be explained later on. After gathering comparisons of pairs of parameters p, the pairs are ordered in a list such that $f(p_{2i-1}) > f(p_{2i})$ ($f(p)$ is unknown at this point, but the relative preference in these points are known) where $i = 1..N$ and N is the number of comparisons made. The preferences can then be reformulated as constraints on the function f as:

$$f(p_{2i-1}) > f(p_{2i}) + C \tag{3.1}$$

where the constant C on the right hand side is important to include, but can be chosen pretty arbitrarily and is easiest chosen as $C = 1$. The optimization problem is then expressed as:

$$\min S[f] \quad \text{subject to} \quad f(p_{2i-1}) > f(p_{2i}) + 1,\ i = 1..N \tag{3.2}$$

The next step was to reduce the complexity of the problem from finding a function in an infinite function space into the much simpler problem of finding the optimal function in a finite function space. This is done by utilizing a Reproducing Kernel Hilbert Space (RKHS) [34] as well as the Representer Theorem [33, 38]. It is shown that the smoothness energy $S[f]$ can be expressed in the Fourier domain as:

$$S[f] = \int \frac{||\tilde{f}(\omega)||^2}{\tilde{G}(\omega)} d\omega \tag{3.3}$$

where $\tilde{f}$ is the Fourier Transform of f and $\tilde{G}$ is the Fourier Transform of a function G. $\tilde{G}(\omega)$ should be small when ω is large in order to punish highly oscillating functions. Now the RKHS comes into play, as $S[f]$ can further be expressed as the square norm $||f||^2$ ($||f|| = \sqrt{\langle f, f\rangle}$ where $\langle \cdot, \cdot \rangle$ is the inner product) inside a RKHS where the function G is the Gaussian kernel $k(x, x')$:

$$k(x, x') = e^{-||x-x'||^2/\sigma^2} \tag{3.4}$$

The problem at hand is now minimizing the norm $||f||^2$, which with a bit of rewriting and utilization of the Representer Theorem shows that the function f holds the form:

$$f(x) = \sum_{i=1}^{2N} \alpha_i k(x, x_i) \tag{3.5}$$

Now the problem has been reduced to finding the $2N$ scalars α of the vector $\boldsymbol{\alpha}$ which minimizes the smoothness energy of f in Equation 3.5, i.e. the function space to search is now finite-dimensional.

Minimizing $||f||^2$ is the same as minimizing the inner product $\langle f, f\rangle$, which using Equation 3.5 can be rewritten as:

$$||f||^2 = \langle f, f\rangle = \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha} \tag{3.6}$$

and the problem is now minimizing $\boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha}$. Again, $\boldsymbol{\alpha}$ is vector of length $2N$, and $\boldsymbol{K}$ is a matrix of shape $2N \times 2N$ where:

$$K_{ab} = k(p_a, p_b),\ a, b = 1..2N \tag{3.7}$$

The constraints in Equation 3.1 can be written as:

$$\sum_{j=1}^{2N} \alpha_i k(x_{2i-1}, x_j) - \sum_{j=1}^{2N} \alpha_i k(x_{2i}, x_j) \geq 1 \tag{3.8}$$

where the first term is found in the odd rows of $\boldsymbol{K}$ and the second term is found in the even rows of $\boldsymbol{K}$ (Note that the odd and even specification only works for an array with one-based indexing and *not* zero-based). If the odd rows are $\boldsymbol{K}_{odd}$ and the even ones $\boldsymbol{K}_{even}$, all constraints can be formulated as ($\boldsymbol{e}$ is a vector of length N filled with ones):

$$(\boldsymbol{K}_{odd} - \boldsymbol{K}_{even})\boldsymbol{\alpha} \geq \boldsymbol{e} \tag{3.9}$$

The minimization is finally:

$$\min \boldsymbol{\alpha}^T \boldsymbol{K} \boldsymbol{\alpha} \quad \text{subject to} \quad (\boldsymbol{K}_{odd} - \boldsymbol{K}_{even})\boldsymbol{\alpha} \geq \boldsymbol{e} \tag{3.10}$$

which can be solved using Quadratic Programming [31] in order to solve for $\boldsymbol{\alpha}$ to get the optimal $\boldsymbol{\alpha}^*$. This can then be substituted in Equation 3.5 which finally results in an expression for the approximated preference function f, which can be optimized by regular means by finding the parameter set p^* which maximizes (or minimizes, depending on how the problem has been defined) f. This covers the important steps and theory of the optimization algorithm, and once again, for more details please refer to the original paper [24].
In order to apply this algorithm the initial step was to reshape the data generated according to Section 3.2.3 into a sequence of length $2N$ filled with parameter sets p, where p_{2i-1} and p_{2i} was part of the same pairwise comparison and p_{2i-1} was the preferred one, i.e. the parameters sets with odd indices are the preferred ones and the following even indices are the non-preferred ones. Then it was possible to calculate the matrix K with elements K_{ab} as in Equation 3.7 and run the optimization to find $\boldsymbol{\alpha}^*$. The Python package *qpsolvers*[4] for Quadratic Programming with the *clarabel* solver was used for this task and $\boldsymbol{\alpha}^*$ was then used along with f from Equation 3.5 to empirically optimize f over parameter space and procure the optimal parameters p^*. This empirical optimization was done using *hyperopt*[5].

3.4 Naturalness prediction

As mentioned in earlier chapters, previous work has developed and trained a model for predicting the naturalness of forests[27]. This trained model will be applied on top of the segmentation and be evaluated by comparing with ground-truth naturalness data.

3.4.1 Data

While there is no reasonable ground-truth data for forest segmentation, there does exist ground-truth data for the naturalness of forests, although a limited amount of

[4] `https://qpsolvers.github.io/qpsolvers/quadratic-programming.html`
[5] `http://hyperopt.github.io/hyperopt/`

data in low resolution. This data consists of georeferenced polygons created by domain experts through field surveys. Such data has been provided by Skogsstyrelsen for areas with both high and low naturalness. More specifically, the high naturalness data is collected from:

- Habitat-classed areas within Natura 2000 from Naturvårdsverket.
- *SksBorealSyd*, an inventory of key habitats in the South Boreal region.
- *Storskogsbruket*, an inventory of key habitats as gathered by forestry companies.

and the low naturalness data is collected from:

- *BestandEjNaturvarden*, forest stands that have been assessed by forestry companies to have low naturalness.
- *Hyggen1990-2000*, an inventory of clearings that have been harvested from during the years 1990-2000.
- *Pskog30till80*, forest stands that are between 30 and 80 years old that have compromised levels of naturalness.

3.4.2 The prediction model

The model is ML-based, but given that an important aspect was to have it be interpretable, it was made to be quite simple and understandable, a logistic regression based on up to *eight* interpretable features calculated from the CHM. These eight features are:

- Tree density — the proportion of the area covered by trees above a certain height.
- Tree height mean — Average of the CHM.
- Tree height variation — Standard deviation of the CHM.
- Treetop density — Similar to tree density but only considering treetops.
- Treetop height mean — Average of treetops in the CHM
- Treetop height variation — Standard deviation of treetops in the CHM
- Edge-like pixels — Edge-like pixels calculated by the local binary pattern algorithm.
- Treetop spatial distribution — Measures regularity of spatial distribution of treetops.

The model was trained on the data mentioned in Section 3.4.1. More specifically the model in question is a pipeline of a min-max scaler to scale all features into the range $[0, 1]$, and then into a logistic regression giving as output a confidence score in how natural the forest area is.

3.4.3 Evaluation

The entire pipeline starts out with an area of interest, and then produces labels on a per-pixel basis when creating segments, and thus also naturalness labels on a per-pixel basis. The ground-truth data however comes in the shape of georeferenced geometries, which meant the ground-truth data had to be translated into per-pixel based data. An example of the ground truth versus prediction of naturalness can be seen in Figure 3.7.

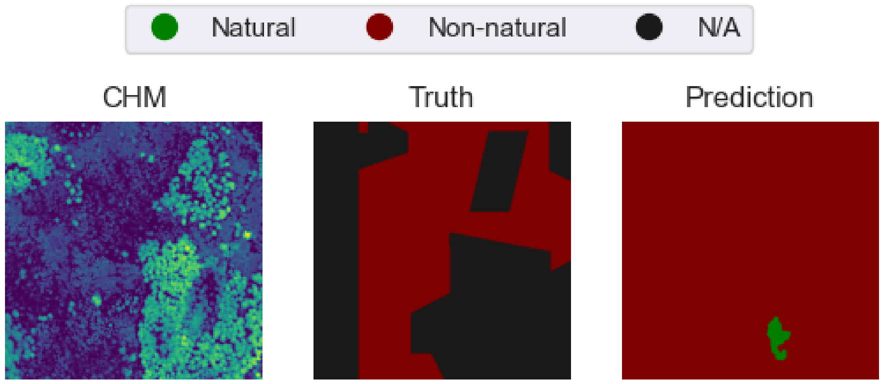


Figure 3.7: Example of CHM, truth class and predicted class for a 320×320 raster. This example shows strictly one natural class or the other instead of confidence for each class as will be done later on in figures.

The ground-truth data to compare with was limited to be contained in an approximately 50×50 kilometer area centered around $6434074\text{N}, 540904\text{E}$ (coordinates in SWEREF99). This area was split up into smaller, square areas (tiles) of size $N \times N$ meters, where only the tiles containing any sort of naturalness data were kept.
Standard ML metrics such as accuracy, precision, recall and F1-score were calculated by comparing per-pixel classifications, where only the pixels with available ground-truth data in each $N \times N$ patch were compared (don't count values marked as N/A in Figure 3.7 for example). A pixel with high naturalness classified as high naturalness is referred to as True Positive (TP), high naturalness classified as low naturalness as False Negative (FN), low naturalness classified as low naturalness as True Negative (TN) and low naturalness classified as high naturalness as False Positive (FP). See also Table 3.1. Given this mapping, the metrics are calculated according to Equations 3.11 - 3.14.

Truth / **Predicted**	Natural	Non-natural
Natural	True Positive (TP)	False Negative (FN)
Non-natural	False Positive (FP)	True Negative (TN)

Table 3.1: Classification map for predicted and true naturalness class.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.11}$$

$$Precision = \frac{TP}{TP + FP} \tag{3.12}$$

$$Recall = \frac{TP}{TP + FN} \tag{3.13}$$

$$F1 = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall} \tag{3.14}$$

4

Results & Discussion

This chapter will present all the results and findings of the thesis while also reiterating some of the important points of the Method. The first section, Section 4.1 covers everything strictly related to the segmentation of the CHM. Section 4.2 covers everything else, which mostly includes the combination with the naturalness evaluator and associated evaluation. Some degree of discussion and/or analysis is presented in close proximity to associated results, then Section 4.3 presents a broader discussion encompassing several of the results and the thesis as a whole, such as reflecting on method choice and uncertainties.

4.1 Segmentation

This first stage of evaluating the segmentation was designed with the purpose of choosing one single best segmentation algorithm to narrow down the problem of optimizing the segmentation. The fixed set of rasters S_{chm} chosen for the first stage can be seen in Figure 4.1 and was defined by the author with the goal of ensuring high forest coverage in this small set of rasters. This was done by randomly generating rasters throughout Sweden and manually filtering out those with low forest coverage while also attempting to make sure S_{chm} included both natural-grown forests as well as non-natural forests.

The set S_p of parameter configurations for the first stage was generated by randomizing and saving parameters-sets beforehand, i.e. by precomputation, and contains 21 configurations of the SLIC algorithm, 18 configurations of the Felzenszwalb algorithm and 15 configurations of the watershed algorithm. The distribution of the parameters in S_p can be seen for Felzenszwalb in Figure 4.2, for SLIC in Figure 4.3 and for watershed in Figure 4.4. Similarly to the pre-defined rasters, these configurations were randomized and then manually filtered to remove configurations in the extremes which resulted in e.g. obvious over-/under-segmentation or other strange post-segmentation artifacts.

With the help of Skogsstyrelsen approximately 80 pairwise evaluations were generated. In order to guide the selection of segmentation algorithm two sources of data based on these evaluations are used, where the first source is the final preference in each subset k and the second is the overall outcome of each and every comparison, where both these are found in Figure 4.5, where 4.5a shows the distribution of final preferences and 4.5b shows the outcomes of all comparisons.

From 4.5a it is evident that the watershed algorithm can be discarded as a viable option, while Felzenszwalb and SLIC perform similarly and no obvious selection

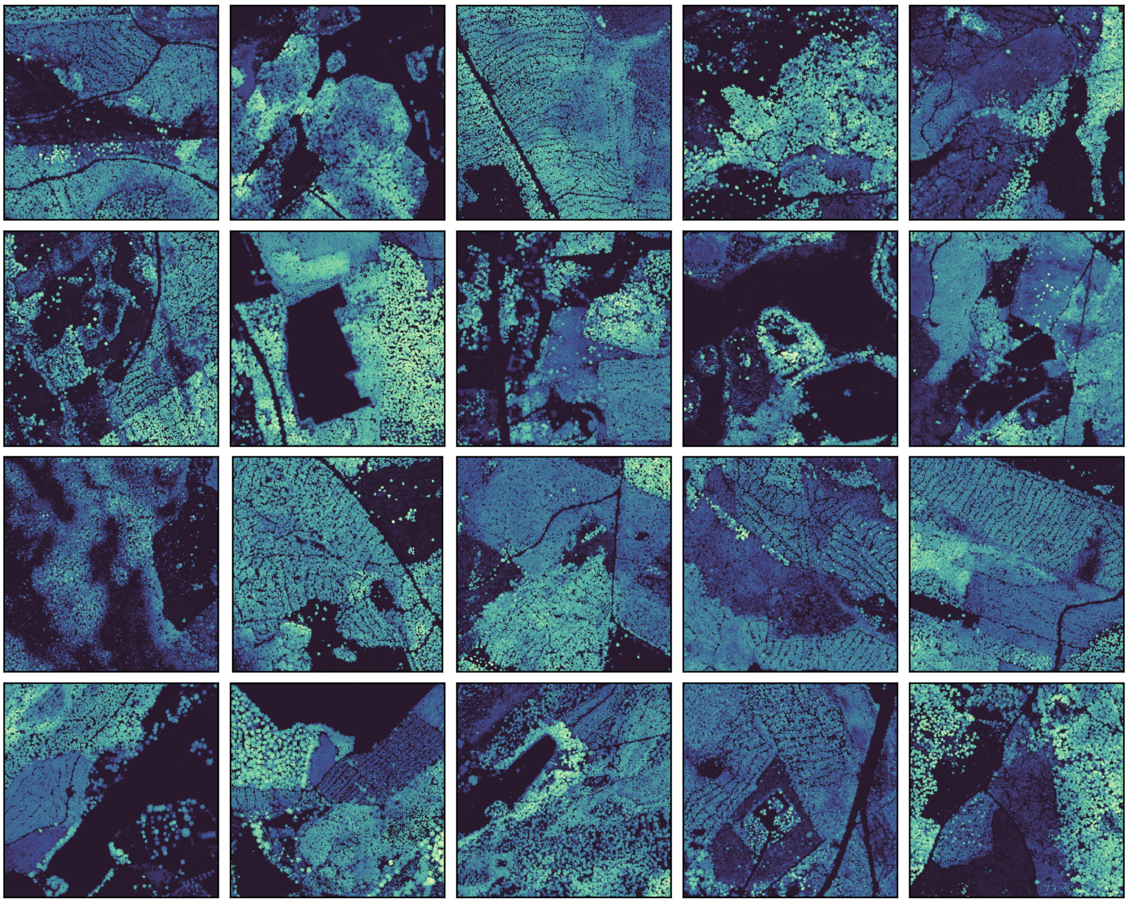


Figure 4.1: All pre-defined rasters in S_{chm} used in the first stage of the segmentation evaluation. The exact coordinates for these rasters can be found in Appendix A.

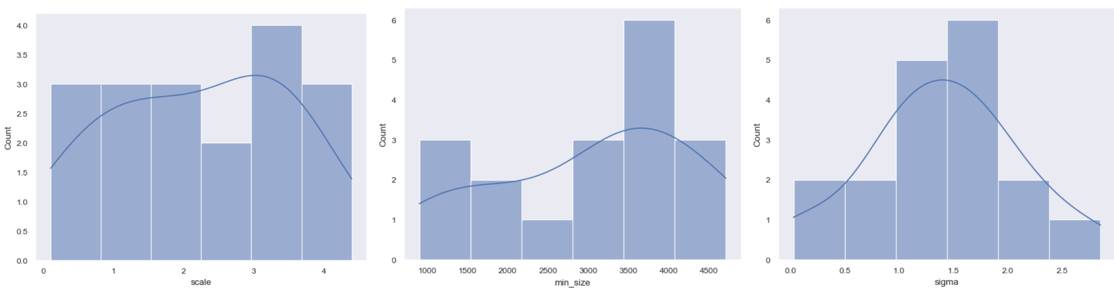


Figure 4.2: Distribution of parameters in the pre-selected set of configurations in the first segmentation evaluation stage for the Felzenszwalb segmentation.

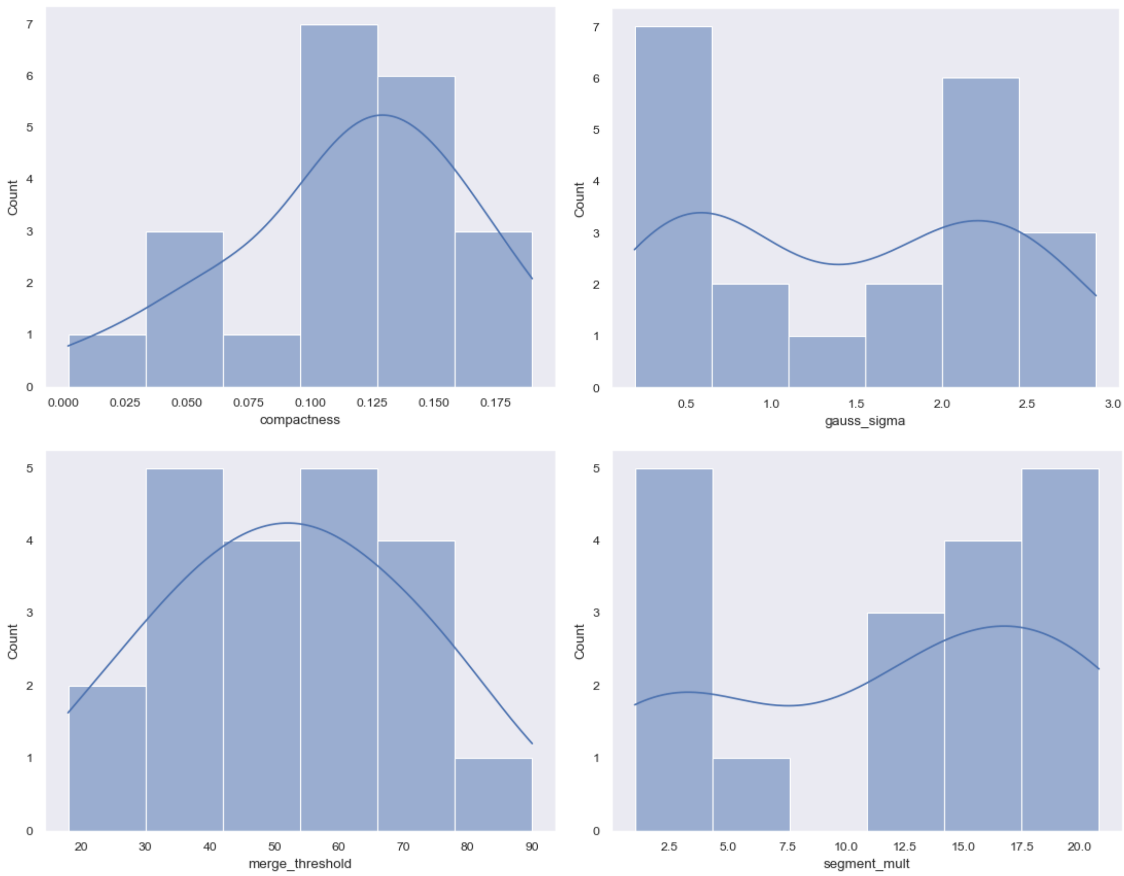


Figure 4.3: Distribution of parameters in the pre-selected set of configurations in the first segmentation evaluation stage for the SLIC segmentation.

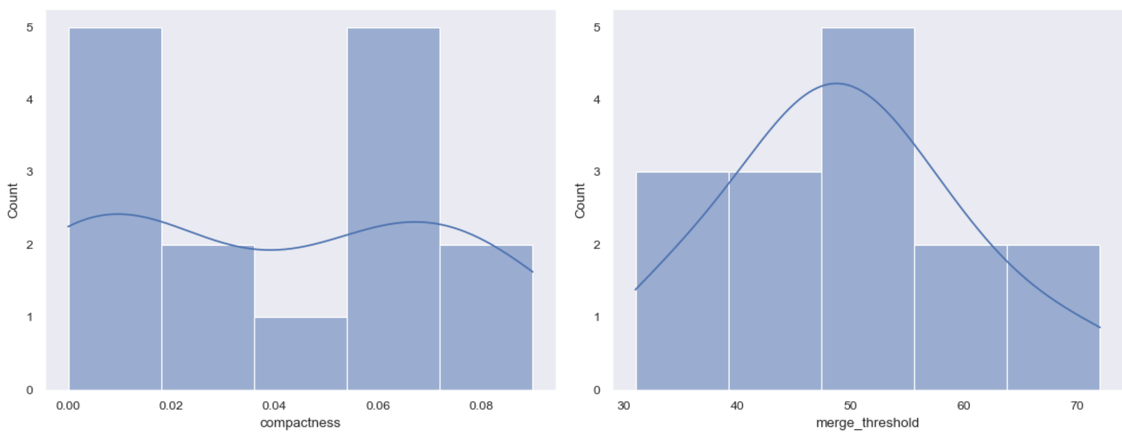
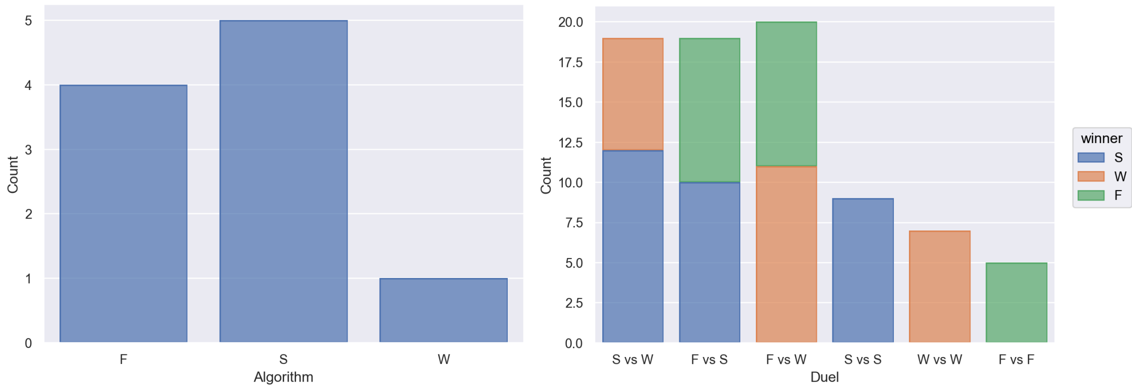


Figure 4.4: Distribution of parameters in the pre-selected set of configurations in the first segmentation evaluation stage for the Watershed segmentation.



(a) Final preferred algorithm in a sequence k.

(b) All "duels" between algorithms.

Figure 4.5: Final preference in each sequence k and overall outcome for each "duel" in the first evaluation stage. F is Felzenszwalb, S is SLIC and W is Watershed.

can be made solely from 4.5a. This makes sense and is according to expectations as watershed initially produces an extreme over-segmentation, approximately on a tree-level, in contrast to SLIC and Felzenszwalb which initially produces larger segments, which intuitively should yield better results when trying to capture more global features of forests rather than tree-level information. This is also a favorable result as the watershed algorithm is terribly slow compared to the others as a result of having to deal with the massive over-segmentation. 4.5b does not add any additional clear and conclusive insight, but does suggest that SLIC might be preferable compared to Felzenszwalb. This also showcases how the small amount of data is a disadvantage, as well as possible flaws with the method itself as, for example, the results may be skewed by the order in which the algorithms appear in each sequence k (a problem likely mitigated by having more data). One ideal solution to this inconclusiveness would of course be to gather more data, but this was not feasible during the time frame. Nevertheless a decision has to be made, and given that SLIC seemingly has a higher preference and the fact that superpixel-based (such as SLIC or SLICO) segmentation methods have been successfully applied to similar problems [9], SLIC is the algorithm that will be used going forward.

The second stage of evaluation was designed in order to optimize the segmentation algorithm to produce as good segments of forest as possible. Whether a segmentation is "good" or not was decided in a subjective manner by domain experts.

The procedure described in Section 3.2.3 was applied by precomputing steps 1 through 4 to generate a large set of randomly generated pairs of segmentations where the rasters are 640×640 meters. These were then randomly sampled to present pairs to evaluate to the user. The precomputation is necessary in order to improve the UX of the web application, since previous evaluation stage had proved the loading times of generating the segmentations on the fly to be disruptive in efficiently generating preference data. Figure 4.6 shows which areas of Sweden the rasters in this large set covers, and the parameter coverage in $\mathbf{P}$ is shown in Figure 4.7 as a sanity check that the evaluation is not biased toward some section of $\mathbf{P}$.

The reason not all of Sweden is covered (in Figure 4.6) is because the CHM data does not exist for some regions, either because the region is covered in e.g. lakes or mountains, or simply because data has not been collected for that region for any other arbitrary reason.
With the help of Skogsstyrelsen, a little over 100 pairwise comparisons were gathered for the SLIC segmentation algorithm, which are now used in order to choose the optimal parameters to segment forests before applying it in tandem with a naturalness classifier. After following the procedure described in Section 3.3, a set of optimal parameters are acquired, which can be found in Table 4.1. A visualization of how the preference function f_p reacts to changing individual parameters can be found in Figure 4.8. If f_p is to be trusted, compactness has basically no effect on the preference, sigma has a clear optimum around 2.0 and segment_mult has a few peaks but a seemingly clear optimum around just above 2. merge_threshold has a sharp spike around 65 and f_p has a very noisy response to change. This large spike could be, at least in part, due to noise, but through manual tweaking and visual inspection, 65 is found to not be an unreasonable value and this spike is likely no fluke.

Parameter	**Optimal Value** Random Search	**Optimal Value** TPE Search
segment_mult	2.14	2.21
compactness	0.10	0.12
sigma	2.04	2.01
merge_threshold	65.14	65.12

Table 4.1: Optimal parameter set for the SLIC segmentation. segment_mult describes how the number of approximate superpixels scales with the size of the raster, compactness is a parameter inherent to SLIC which affects the general shape of each superpixel, sigma is the kernel size of the pre-processing smoothing and merge_threshold is the threshold used to decide if two segments should merge or not in the merging of superpixels/segments.

4.2 Naturalness

The test area in Sweden spans 50×50km and is split up into square ($N \times N$) tiles of different sizes, both larger and smaller than the size data was gathered on. The spatial distribution for each tile size can be seen in Figure 4.9 and the area occupation in Figure 4.10. The most important to look at is 640×640m which is the same as preference data was gathered on. After running the segmentation and naturalness prediction on these and comparing the output pixel-wise against the ground-truth naturalness data, standard ML metrics have been gathered in Table 4.2 and 4.3. For reference, previous work which developed the naturalness predictor [27] reached an accuracy of approximately 90% on its test set, but keep in mind they dealt with classifying entire georeferenced polygons which had *not* been split up into tiles.

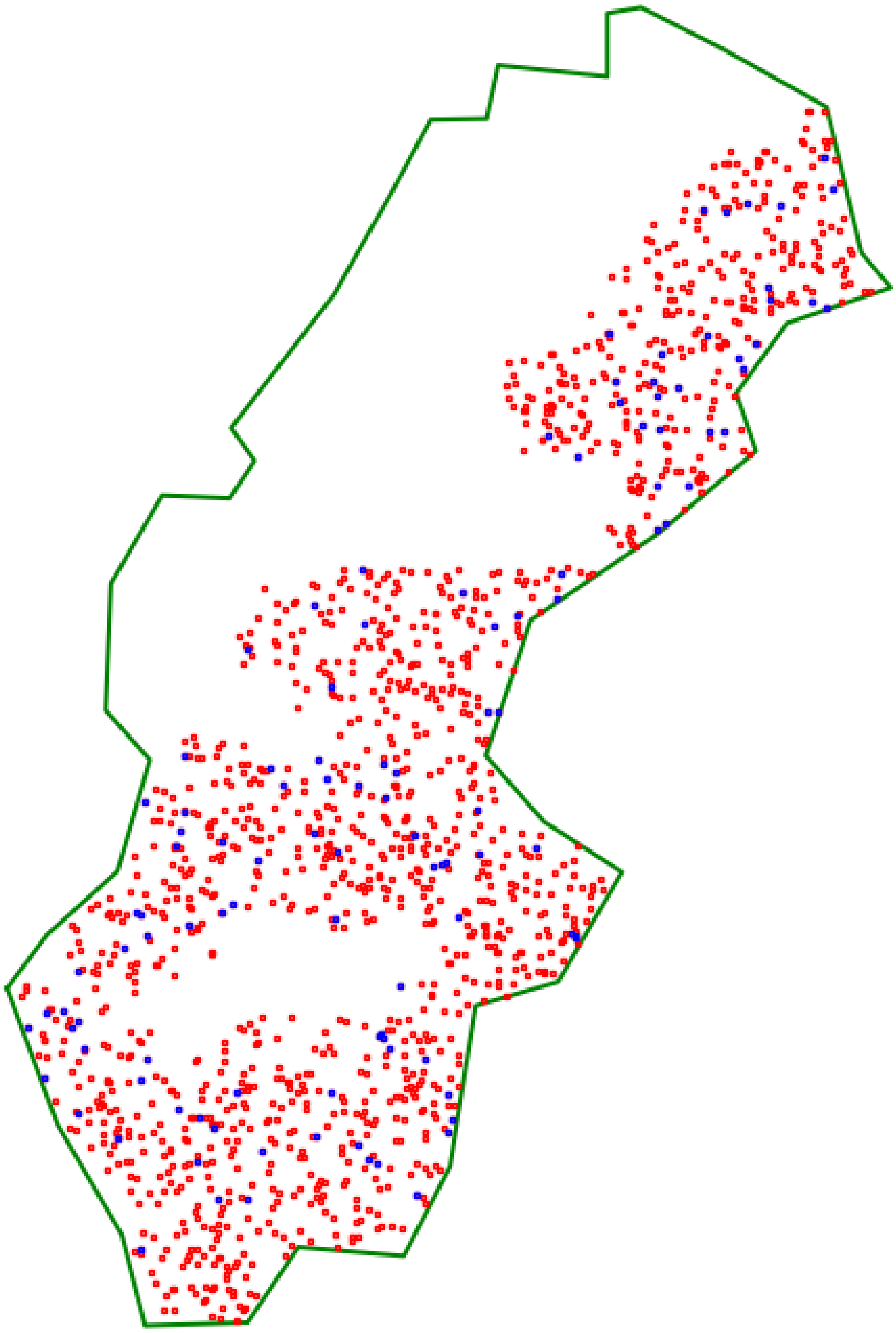


Figure 4.6: Raster coverage of Sweden for the fine-stage segmentation evaluation. Each square is a 640×640 meter raster where the red squares show all rasters part of the precomputations and the blue squares represent the subset of those rasters having received an associated preference.

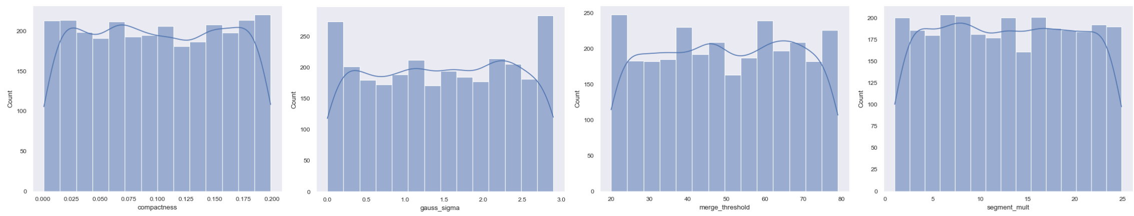


Figure 4.7: Distribution of parameters in the set of configurations in the second stage of segmentation evaluation stage for the SLIC segmentation. The distribution is even, as is expected for randomly selected parameters inside an interval.

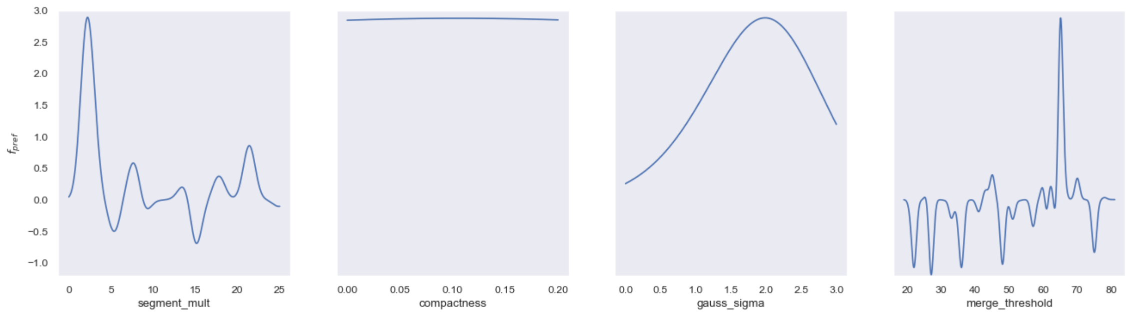


Figure 4.8: How the approximated preference function responds to varying one input parameter at a time. Remaining parameters are locked at the optimized values.

The accuracy is approximately 75% on the test area made up of 640×640 patches. This is a significant drop when comparing to 90%, but is within expectations as a more complex operation is being evaluated compared to classifications of entire rasters. Despite that, such a drop in accuracy still suggest that the segmentation does not necessarily work perfectly together with the naturalness classifier, which could be due to several factors. The main reason, while referring to Table 4.2, is due to a high ratio of false-positive predictions for high naturalness. Whether these false-positives are caused by the naturalness predictor or as a consequence of the segmentation or the evaluation method warrants further investigation before saying for sure.
It is also interesting to note that the accuracy and precision increases/decreases for smaller/larger raster sizes, while the recall (and kind of the F1-score as well) stays the same. This does not have to signify that the segmentation works better on smaller sizes and worse on larger sizes, and this will be discussed more later on.
Figure 4.11 shows examples of the CHM, the segmentation, and the naturalness probability, for increasing sizes of the raster, where each raster is centered around the same reference point. From visually inspecting this example the segmentations seems pretty okay, especially for the smaller sizes, but there are still some glaring issues. For larger raster sizes, the consistency and quality of the segmentation risks degrading, e.g. by merging segments that probably should not be merged. One way to see this is in the two bottom-most areas where there seems to be one large segment spreading over the entire image in strange ways. There is also still somewhat of an issue with “speck”-sized segments throughout the larger segments, which is

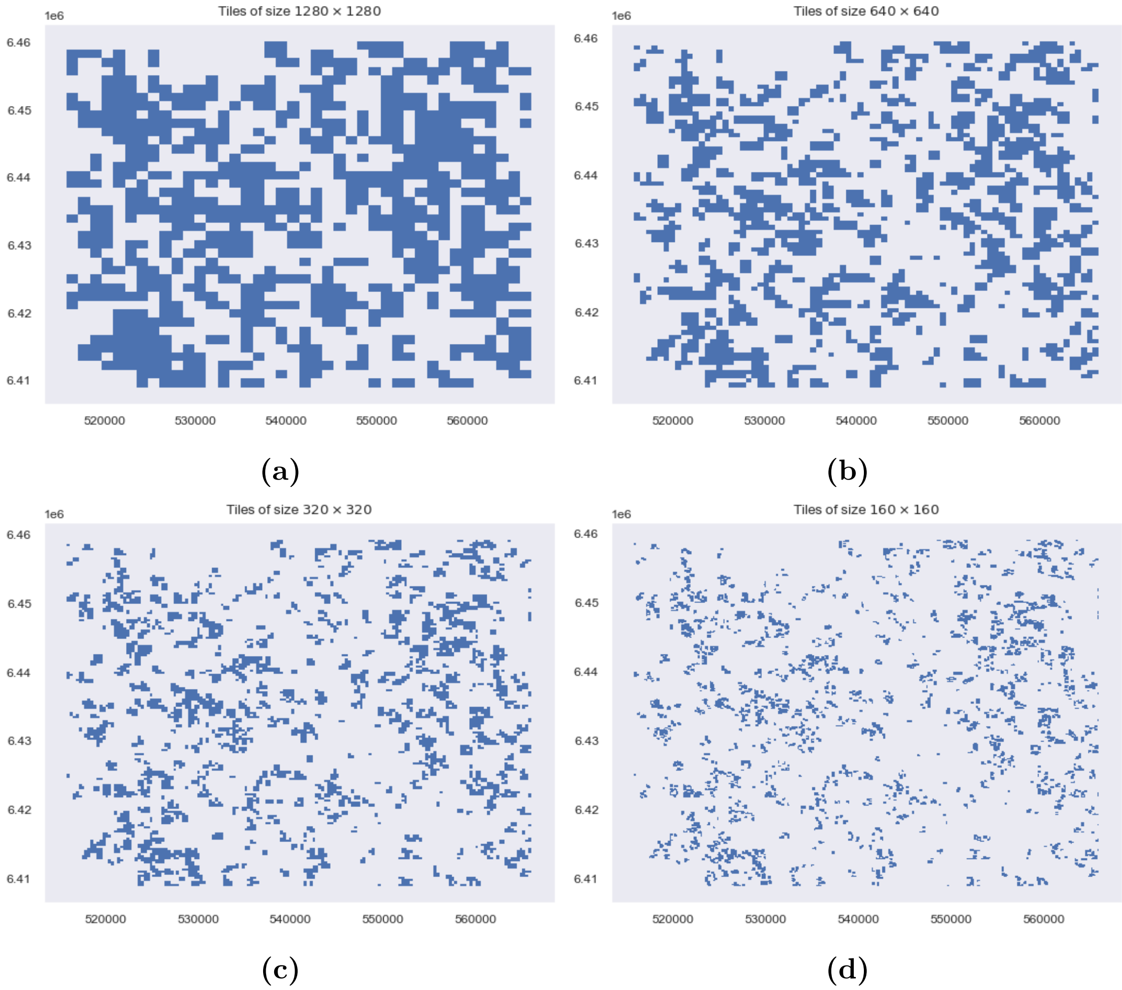


Figure 4.9: Coverage of tiles in the test region for tile sizes **a)** 1280×1280, **b)** 640×640, **c)** 320×320 and **d)** 160×160. Axis values are coordinates in SWEREF99. See also Figure 4.10.

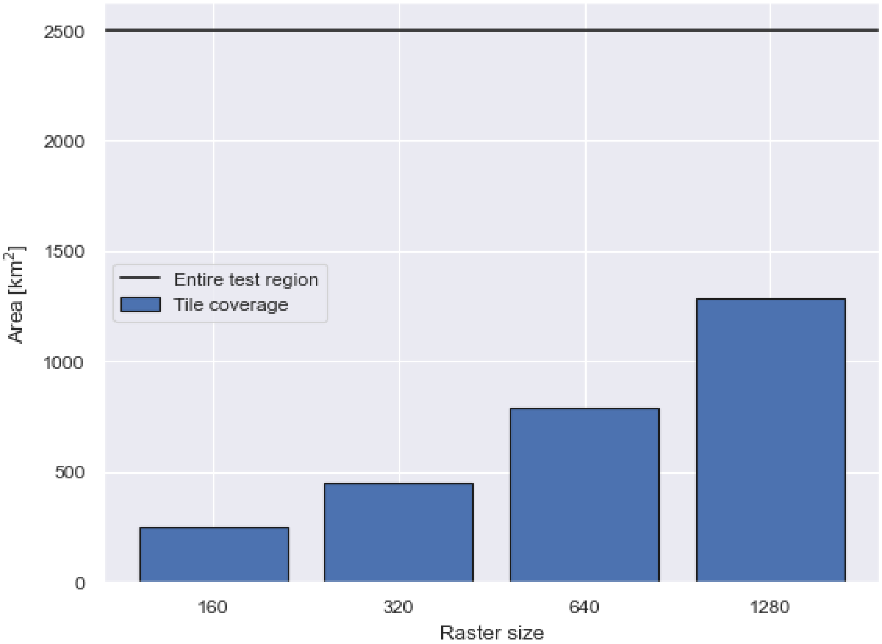


Figure 4.10: Total area covered by tiles for each tile size, compared to area of entire test region. See also Figure 4.9.

Type	Size	Value
True Positive	160	12 225 147
	320	13 163 037
	640	13 469 364
	1280	13 582 547
True Negative	160	40 098 492
	320	37 972 969
	640	34 278 216
	1280	32 039 370
False Positive	160	3 048 813
	320	5 436 979
	640	8 994 883
	1280	11 959 442
False Negative	160	7 235 637
	320	6 805 254
	640	6 791 034
	1280	6 805 254

Table 4.2: Pixel-wise counts (TP, FP, TN, FP) over the test set when comparing naturalness ground truth against output from the model.

Metric	Size	Value
Accuracy	160	0.83
	320	0.81
	640	0.75
	1280	0.71
Precision	160	0.80
	320	0.71
	640	0.60
	1280	0.53
Recall	160	0.63
	320	0.66
	640	0.66
	1280	0.67
F1	160	0.70
	320	0.68
	640	0.64
	1280	0.60

Table 4.3: Typical ML metrics for the test set when comparing naturalness ground-truth against the output from the entire segmentation & prediction pipeline.

not ideal. These issues not only produces worse segments, but also propagates to the naturalness estimation, resulting in less accurate estimations.

4.3 Discussion

This section presents broader discussion encompassing several parts of the thesis, roughly grouped into a handful of subsections. These sections are are interpretability (4.3.1), data (4.3.2), segmentation (4.3.3) and naturalness (4.3.4).

4.3.1 Interpretability

Throughout the thesis only classical segmentation methods were considered in the spirit of increased interpretability. When it comes to the naturalness estimator the interpretability is without a doubt an important step, as the estimated naturalness would be what affects several interested parties in the end such as forestry companies. However, for the segmentation, the interpretability is not necessarily as important, since the how and the why in how a segment of forest was determined is not as vital a piece of information. At the contrary, if the interpretable algorithm results in segmentations further away from the desired output than a non-interpretable method would, it will likely only reflect negatively upon the naturalness estimation. Thus a segmentation method based on e.g. deep learning could also be perfectly fine, disregarding the issue of requiring ground-truth data for typical deep learning approaches.
There is also no need to strictly limit oneself to only one or the other. It might be possible to use both an interpretable model as well as e.g. one based on deep

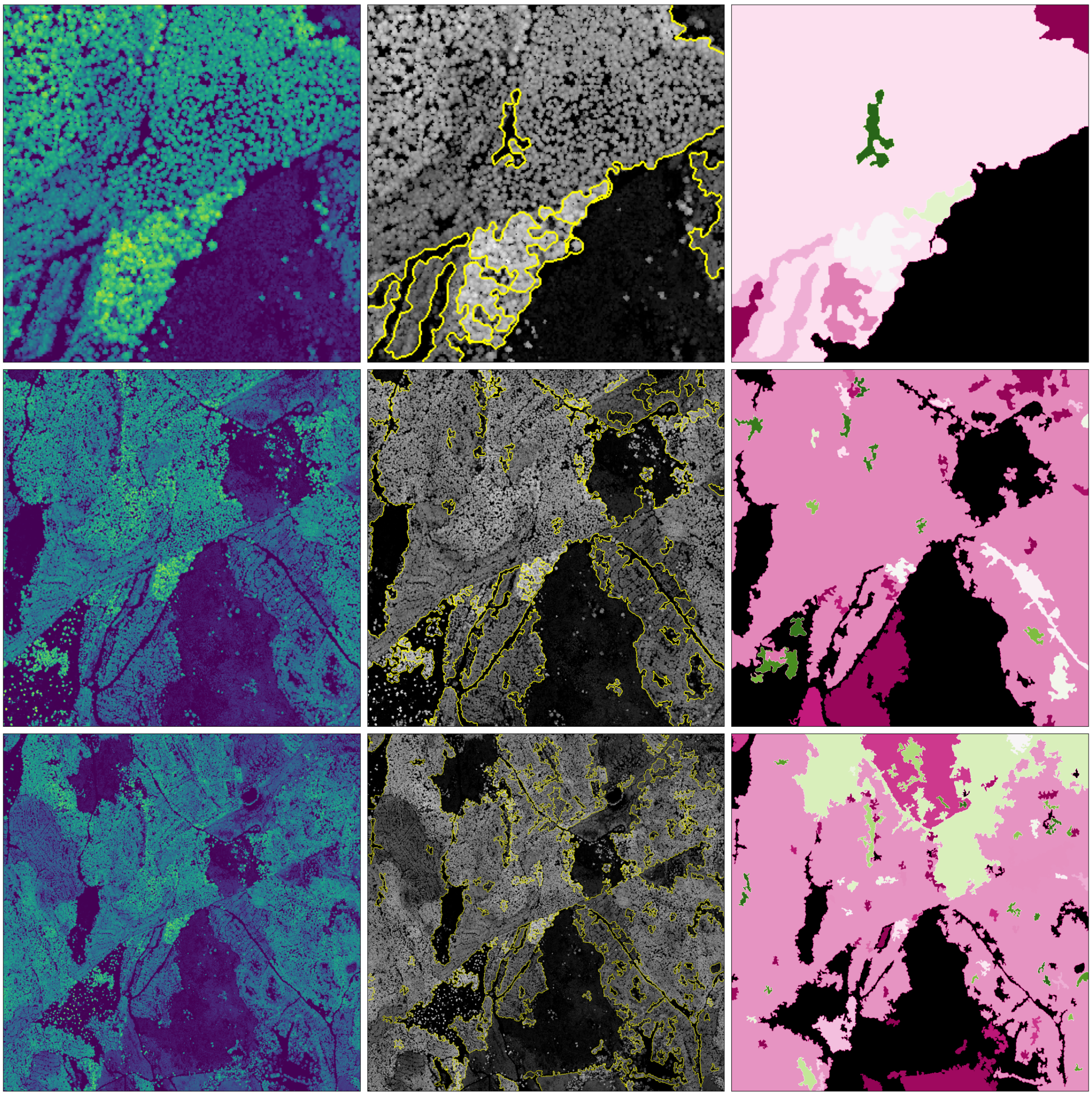


Figure 4.11: From left to right: CHM, segmentation and naturalness prediction, for increasing sizes of rasters. The sizes from top to bottom are 320×320m, 640×640m and 1280×1280m. The optimal SLIC segmentation optimized with preferences on 640×640m rasters was used here. The naturalness is colored in pink for non-natural with a higher saturation meaning higher confidence, and vice versa for natural and green color. Black segments in the naturalness represent segments determined to be non-forests.

learning, and take the output of both of these into account. One such way could be to utilize field studies on areas where different models do not agree or are uncertain on the naturalness, and such provide more valuable training data at the same time.

4.3.2 Data

The method of evaluating segmentations in this thesis was subjective in nature, and at first glance it is easy to dismiss this method as a worse one compared to e.g. comparisons with ground-truth data. While it is true that this would provide more typical and recognizable measures of a models performance, there is likely no other advantage in this case. This is because the target for each segment is quite abstract in this case, sort of "a reasonable region of forest". In order to generate ground-truth data for such segments, humans would have to annotate rasters with such segments, which requires high levels of time and effort, and will inherently also be subjective given the general and abstract idea a segment should capture. Given this, the pairwise comparison method should not yield worse segmentations, and also be more effective in gathering large amounts of data given the speed of these relative judgements of segmentation quality.

Something that undoubtedly is a flaw though, is the amount of data that was gathered being small. This is especially evident during the first stage of segmentation evaluation where the indecisiveness is obvious. The optimization most likely also suffers from lack of data, but not to the same degree, as the original paper for the optimization algorithm suggests decent performance even with a limited amount of data points.

Something also worth mentioning in this context is that due to subjective pairwise comparisons and small amounts of data, the entire evaluation and optimization is filled with some degree of noise and uncertainty, meaning e.g. the preference function and the exact values of the optimal parameters should be taken with a grain a salt and be treated as an approximate optimal point. Also, keep in mind that it could still be possible that the maximum of the approximated preference function does not represent the actual best parameters, whether this is the best parameters for good segments, or best parameters for high accuracy in the naturalness estimation.

4.3.3 Segmentation

When comparing several different segmentation algorithms, the conditions of creating the final segments (for watershed and SLIC) were deliberately chosen to mimic a typical image segmentation procedure and only be based on the pixel-values in the raster, i.e. the height, more specifically the mean height per segment. This was to keep the condition simple, computationally fast, and in some sense in order to make the comparison fair, as the graph-based algorithm only used the actual raster values to generate the final segments and had no additional region merging step. In the later stages, when only working with SLIC (superpixels), there is no reason to stick with the simple merging conditions and it would most certainly have been justified to explore other conditions to merge segments on. The entire method of region merging itself also warrants more investigation. Reason being for example to investigate how scale-invariant the naturalness of a forest is and how segment creating by region merging affects this. For example, might it be possible for a large region of forest to be considered non-natural, but if the same area is split into smaller segments, each segment instead be considered natural or vice versa.

4.3.4 Naturalness

It was noted that the collected metrics behaved in an interesting way when varying the tile size (see Figure 4.2 and 4.3). Unfortunately it is not reliable to draw any further conclusions about the tile sizes from this data, but instead, together with a couple other results, it is possible to identify some possible areas of improvements regarding the methods used. The first such insight is that the reason for worse metrics in larger tile sizes is mainly because of increasing False Positives, which is likely due to the increased amount of data to segment in larger tiles, resulting in higher risk of excessive merging of segments. For example, the naturalness classifier tends to give a higher naturalness score to non-forest segments, and larger tiles more often wrongfully merges non-forest segments with forest segments, for example in Figure 4.11. The second insight, from the perspective of why smaller tiles result in better metrics, is that smaller tiles provides less unlabelled data, as is pointed out in Figures 4.10 and 4.9. This means that for smaller sizes, the naturalness classifier receives segments with higher ratios of familiar data. This also brings us to the issue of creating a mismatch in the shape of data the model was trained on and the shape of the data it is being evaluated against for this thesis.
The naturalness model was trained on low resolution polygons encapsulating areas of forest which are labelled either natural or non-natural. The segmentation and naturalness prediction works on a pixel- and segment-basis with higher resolution. As mentioned, this creates a mismatch between the computed data and truth data, especially since the pixel-based truth data is split up into smaller cells of a grid, meaning what previously was one whole polygon of forest, is now several smaller regions split up into cells in a grid of rasters. This might affect the predictions of the model as it has to predict each sub-segment of the original polygon correctly in order to predict the entire polygon correctly. Thus it would be expected for the accuracy of the model to be worse compared to the accuracy when only working on polygons, when evaluated as in Section 4.2. This warrants further investigation into taking into account this split into a grid when training the model. It might also be valuable to consider if this kind of segmentation is the correct approach, or if some other method would be more useful or accurate. One such method could be to not do any segmentation, and instead strictly work with a grid of (small) rasters, where each raster as a whole is evaluated on its naturalness. This could also produce a naturalness map, but with much lower resolution, which further research will have to decide whether it is acceptable or not.

5 Conclusion

5.1 Summary

This thesis has applied and evaluated several segmentation algorithms on the Canopy Height Model with the goal of segmenting a raster into forest segments. The most suitable algorithm has been chosen and optimized based on pairwise comparisons by human feedback and then applied in tandem with a naturalness classifier. The aim with this was to reduce costly field studies and instead generate a naturalness map of Sweden's forests by applying interpretable AI-based methods on ever-growing amounts of data. The results is a complete model with an accuracy of about 75%. While this is by no means a perfect model, given that there is space for much improvement to both segmentation and classification, this interpretable model shows a lot of promise for the future. In conclusion, this thesis has achieved important ground work by finishing a basic and working pipeline from CHM to naturalness using interpretable methods, on top of having developed a simple tool to evaluate and optimize through preference-based feedback.

5.2 Future Work

While image segmentation is a well explored area in research, estimating the naturalness of forests is not, and there are lots of aspects to be improved or things to add from this point onwards. Probably the most important next step is creating a more complete method of generating naturalness maps using the segmentation and naturalness estimator. This comes with various issues such as dealing with the fact that only so large rasters can fit in memory at once, meaning some sort of division of Sweden into smaller, probably parallelizable, regions is necessary, as well as dealing with possible effects caused by the borders between such regions.

There are also many other topics of potential interest, where some of these could be:

- Reinforcement learning with human feedback (RLHF) is something that has been discussed recently in the context of e.g. large language models. Could adapt or change method to enable RLHF and continuous improvement of segmentation and naturalness estimation.
- Utilizing several sources of data when doing the segmentation and naturalness estimation instead of only the CHM. Such sources could for example be the orthophoto or the Digital Terrain Model.

- Explore the possibility of segmenting forests using Deep Neural Networks and compare.
- Try replacing SLIC segmentation with a more recent algorithm called SNIC (Simple Non-Iterative Clustering), which is supposed to be an objective improvement over SLIC.
- Explore other conditions to perform region merging on. Examples of this could be to merge on same features as naturalness estimator is based on.
- Optimize more algorithms in a similar way as for SLIC, and compare optimized versions inside the entire naturalness pipeline.
- Implement ability to defer to some heuristic to decide what is better in a pairwise comparison when a user can not decide which one is better, i.e. for highly similar pairs. Would improve experience from users perspective.
- Look into other methods of evaluation by human preference. One such example could be preferential Bayesian optimization, or some evolutionary algorithm based on preferences to calculate fitness.
- Attempt some other more sophisticated algorithm to create segments from superpixels except for simple region merging.
- Create a more robust method to determine whether a segment actually is a forest or not to filter these out from naturalness estimation if desired.
- Investigate the computational speed of the method, and potentially optimize to enable use for possibly similar tasks that are more time-critical.

Bibliography

[1] Radhakrishna Achanta et al. “SLIC Superpixels Compared to State-of-the-Art Superpixel Methods”. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34.11 (Nov. 2012). Conference Name: IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 2274–2282. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2012.120. URL: https://ieeexplore.ieee.org/document/6205760 (visited on 06/12/2024).

[2] Riad Akrour, Marc Schoenauer, and Michèle Sebag. *APRIL: Active Preference-learning based Reinforcement Learning.* Aug. 5, 2012. DOI: 10.48550/arXiv.1208.0984. arXiv: 1208.0984[cs]. URL: http://arxiv.org/abs/1208.0984 (visited on 03/14/2024).

[3] Serge Beucher. “Watershed, Hierarchical Segmentation and Waterfall Algorithm”. In: *Mathematical Morphology and Its Applications to Image Processing.* Ed. by Jean Serra and Pierre Soille. Red. by Max A. Viergever. Vol. 2. Series Title: Computational Imaging and Vision. Dordrecht: Springer Netherlands, 1994, pp. 69–76. DOI: 10.1007/978-94-011-1040-2_10. URL: http://link.springer.com/10.1007/978-94-011-1040-2_10 (visited on 02/02/2024).

[4] Andrè Bleau and L. Joshua Leon. “Watershed-Based Segmentation and Region Merging”. In: *Computer Vision and Image Understanding* 77.3 (Mar. 1, 2000), pp. 317–370. ISSN: 1077-3142. DOI: 10.1006/cviu.1999.0822. URL: https://www.sciencedirect.com/science/article/pii/S1077314299908226 (visited on 02/03/2024).

[5] Eric Brochu. “Interactive Bayesian optimization : learning user preferences for graphics and animation”. PhD thesis. University of British Columbia, 2010. DOI: 10.14288/1.0051462. URL: https://open.library.ubc.ca/soa/cIRcle/collections/ubctheses/24/items/1.0051462 (visited on 02/19/2024).

[6] Alexander Buchelt et al. “Exploring artificial intelligence for applications of drones in forest ecology and management”. In: *Forest Ecology and Management* 551 (Jan. 1, 2024), p. 121530. ISSN: 0378-1127. DOI: 10.1016/j.foreco.2023.121530. URL: https://www.sciencedirect.com/science/article/pii/S0378112723007648 (visited on 03/02/2024).

[7] Paul Christiano et al. *Deep reinforcement learning from human preferences.* Feb. 17, 2023. DOI: 10.48550/arXiv.1706.03741. arXiv: 1706.03741[cs, stat]. URL: http://arxiv.org/abs/1706.03741 (visited on 06/12/2024).

[8] Wei Chu and Zoubin Ghahramani. “Preference learning with Gaussian processes”. In: *Proceedings of the 22nd international conference on Machine learn-*

ing - ICML '05 (2005). Conference Name: the 22nd international conference ISBN: 9781595931801 Place: Bonn, Germany Publisher: ACM Press, pp. 137–144. DOI: 10.1145/1102351.1102369. URL: http://portal.acm.org/citation.cfm?doid=1102351.1102369 (visited on 02/16/2024).

[9] Ovidiu Csillik. "Fast Segmentation and Classification of Very High Resolution Remote Sensing Data Using SLIC Superpixels". In: *Remote Sensing* 9.3 (Mar. 2017). Number: 3 Publisher: Multidisciplinary Digital Publishing Institute, p. 243. ISSN: 2072-4292. DOI: 10.3390/rs9030243. URL: https://www.mdpi.com/2072-4292/9/3/243 (visited on 02/11/2024).

[10] Clément Dechesne et al. "Semantic segmentation of forest stands of pure species combining airborne lidar data and very high resolution multispectral imagery". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 126 (Apr. 1, 2017), pp. 129–145. ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2017.02.011. URL: https://www.sciencedirect.com/science/article/pii/S0924271616302763 (visited on 02/11/2024).

[11] Oliver Diedershagen, Barbara Koch, and Holger Weinacker. "Automatic segmentation and characterisation of forest stand parameters using airborne lidar data, multispectral and fogis data". In: *Institute of Forestry Economics. University Freiburg.* (Oct. 3, 2004).

[12] *Drainage divide*. In: *Wikipedia*. Page Version ID: 1184602912. Nov. 11, 2023. URL: https://en.wikipedia.org/w/index.php?title=Drainage_divide&oldid=1184602912 (visited on 03/11/2024).

[13] Lothar Eysn et al. "Forest Delineation Based on Airborne LIDAR Data". In: *Remote Sensing* 4.3 (Mar. 2012). Number: 3 Publisher: Molecular Diversity Preservation International, pp. 762–783. ISSN: 2072-4292. DOI: 10.3390/rs4030762. URL: https://www.mdpi.com/2072-4292/4/3/762 (visited on 01/25/2024).

[14] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. "Efficient Graph-Based Image Segmentation". In: *International Journal of Computer Vision* 59.2 (Sept. 2004), pp. 167–181. ISSN: 0920-5691. DOI: 10.1023/B:VISI.0000022288.19776.77. URL: http://link.springer.com/10.1023/B:VISI.0000022288.19776.77 (visited on 01/25/2024).

[15] Amelia Glaese et al. *Improving alignment of dialogue agents via targeted human judgements*. Sept. 28, 2022. DOI: 10.48550/arXiv.2209.14375. arXiv: 2209.14375[cs]. URL: http://arxiv.org/abs/2209.14375 (visited on 03/14/2024).

[16] Javier I. González et al. "Preferential Bayesian Optimization". In: International Conference on Machine Learning. Apr. 12, 2017. URL: https://www.semanticscholar.org/paper/caaba96ed80dd0e510145badb0129763b993f4de (visited on 02/16/2024).

[17] Annapoorani Gothandaraman. *Hierarchical Image Segmentation using The Watershed Algorithim with A Streaming Implementation*. URL: https://trace.tennessee.edu/utk_gradthes/1966.

[18] K. Haris et al. "Hybrid image segmentation using watersheds and fast region merging". In: *IEEE Transactions on Image Processing* 7.12 (Dec. 1998),

pp. 1684–1699. ISSN: 10577149. DOI: 10.1109/83.730380. URL: http://ieeexplore.ieee.org/document/730380/ (visited on 02/02/2024).

[19] Reinhard Heckel et al. "Approximate Ranking from Pairwise Comparisons". In: *ArXiv* (Jan. 4, 2018). URL: https://www.semanticscholar.org/paper/Approximate-Ranking-from-Pairwise-Comparisons-Heckel-Simchowitz/a8d1e71ae98b7ee0d6d76a5a07bc069f8ec9fa30 (visited on 02/18/2024).

[20] Yifeng Hong et al. "Combining Multisource Data and Machine Learning Approaches for Multiscale Estimation of Forest Biomass". In: *Forests* 14.11 (Nov. 2023). Number: 11 Publisher: Multidisciplinary Digital Publishing Institute, p. 2248. ISSN: 1999-4907. DOI: 10.3390/f14112248. URL: https://www.mdpi.com/1999-4907/14/11/2248 (visited on 03/02/2024).

[21] N. Ikonomatakis et al. "Region growing and region merging image segmentation". In: *Proceedings of 13th International Conference on Digital Signal Processing*. 13th International Conference on Digital Signal Processing. Vol. 1. Santorini, Greece: IEEE, 1997, pp. 299–302. ISBN: 978-0-7803-4137-1. DOI: 10.1109/ICDSP.1997.628077. URL: http://ieeexplore.ieee.org/document/628077/ (visited on 02/03/2024).

[22] Kevin G Jamieson and Robert Nowak. *Active Ranking using Pairwise Comparisons.*

[23] Leonid Keselman et al. "Optimizing Algorithms from Pairwise User Preferences". In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (Oct. 1, 2023). Conference Name: 2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) ISBN: 9781665491907 Place: Detroit, MI, USA Publisher: IEEE, pp. 4161–4167. DOI: 10.1109/IROS55552.2023.10342081. URL: https://ieeexplore.ieee.org/document/10342081/ (visited on 02/16/2024).

[24] Pavel Kisilev and Daniel Freedman. "Parameter Tuning from Pairwise Preferences". In: *Procedings of the British Machine Vision Conference 2010*. British Machine Vision Conference 2010. Aberystwyth: British Machine Vision Association, 2010, pp. 4.1–4.11. ISBN: 978-1-901725-40-7. DOI: 10.5244/C.24.4. URL: http://www.bmva.org/bmvc/2010/conference/paper4/index.html (visited on 02/18/2024).

[25] *Law of comparative judgment*. In: *Wikipedia*. Page Version ID: 1184981018. Nov. 13, 2023. URL: https://en.wikipedia.org/w/index.php?title=Law_of_comparative_judgment&oldid=1184981018 (visited on 03/14/2024).

[26] Alex M. Lechner, Giles M. Foody, and Doreen S. Boyd. "Applications in Remote Sensing to Forest Ecology and Management". In: *One Earth* 2.5 (May 22, 2020). Publisher: Elsevier, pp. 405–412. ISSN: 2590-3330, 2590-3322. DOI: 10.1016/j.oneear.2020.05.001. URL: https://www.cell.com/one-earth/abstract/S2590-3322(20)30206-2 (visited on 03/02/2024).

[27] Marco L. Della Vedova and Mattias Wahde. "An Interpretable AI Method for Assessing Naturalness of Forests from the Canopy Height Model". (Unpublished Manuscript). Dec. 26, 2023.

[28] Jukka Mustonen, Petteri Packalén, and Annika Kangas. "Automatic segmentation of forest stands using a canopy height model and aerial photography". In: *Scandinavian Journal of Forest Research* 23.6 (Dec. 1, 2008). Pub-

lisher: Taylor & Francis _eprint: https://doi.org/10.1080/02827580802552446, pp. 534–545. ISSN: 0282-7581. DOI: 10.1080/02827580802552446. URL: https://doi.org/10.1080/02827580802552446 (visited on 01/25/2024).

[29] Kenneth Olofsson and Johan Holmgren. "Forest stand delineation from lidar point-clouds using local maxima of the crown height model and region merging of the corresponding Voronoi cells". In: *Remote Sensing Letters* 5.3 (Mar. 4, 2014), pp. 268–276. ISSN: 2150-704X. DOI: 10.1080/2150704X.2014.900203. URL: https://doi.org/10.1080/2150704X.2014.900203 (visited on 01/25/2024).

[30] Long Ouyang et al. *Training language models to follow instructions with human feedback*. Mar. 4, 2022. DOI: 10.48550/arXiv.2203.02155. arXiv: 2203.02155[cs]. URL: http://arxiv.org/abs/2203.02155 (visited on 03/14/2024).

[31] *Quadratic programming*. In: *Wikipedia*. Page Version ID: 1197129173. Jan. 19, 2024. URL: https://en.wikipedia.org/w/index.php?title=Quadratic_programming&oldid=1197129173 (visited on 03/18/2024).

[32] *Reinforcement learning from human feedback*. In: *Wikipedia*. Page Version ID: 1213624647. Mar. 14, 2024. URL: https://en.wikipedia.org/w/index.php?title=Reinforcement_learning_from_human_feedback&oldid=1213624647 (visited on 03/14/2024).

[33] *Representer theorem*. In: *Wikipedia*. Page Version ID: 1184856247. Nov. 13, 2023. URL: https://en.wikipedia.org/w/index.php?title=Representer_theorem&oldid=1184856247 (visited on 03/25/2024).

[34] *Reproducing kernel Hilbert space*. In: *Wikipedia*. Page Version ID: 1212054777. Mar. 5, 2024. URL: https://en.wikipedia.org/w/index.php?title=Reproducing_kernel_Hilbert_space&oldid=1212054777 (visited on 03/25/2024).

[35] Thomas L. Saaty. *Decision Making for Leaders: The Analytic Hierarchy Process for Decisions in a Complex World*. Google-Books-ID: c8KqSWPFwIUC. RWS Publications. 336 pp. ISBN: 978-1-888603-13-2.

[36] Thomas L. Saaty. "Relative measurement and its generalization in decision making why pairwise comparisons are central in mathematics for the measurement of intangible factors the analytic hierarchy/network process". In: *RACSAM - Revista de la Real Academia de Ciencias Exactas, Fisicas y Naturales. Serie A. Matematicas* 102.2 (Sept. 1, 2008), pp. 251–318. ISSN: 1579-1505. DOI: 10.1007/BF03191825. URL: https://doi.org/10.1007/BF03191825 (visited on 03/14/2024).

[37] Dorsa Sadigh et al. "Active Preference-Based Learning of Reward Functions". In: *Robotics: Science and Systems XIII*. Robotics: Science and Systems 2017. Robotics: Science and Systems Foundation, July 12, 2017. ISBN: 978-0-9923747-3-0. DOI: 10.15607/RSS.2017.XIII.053. URL: http://www.roboticsproceedings.org/rss13/p53.pdf (visited on 02/16/2024).

[38] Bernhard Schölkopf, Ralf Herbrich, and Alex J. Smola. "A Generalized Representer Theorem". In: *Computational Learning Theory*. Ed. by David Helmbold and Bob Williamson. Red. by G. Goos, J. Hartmanis, and J. Van Leeuwen. Vol. 2111. Series Title: Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 416–426. DOI: 10.1007/3-540-44581-

1_27. URL: http://link.springer.com/10.1007/3-540-44581-1_27 (visited on 03/25/2024).
[39] Nisan Stiennon et al. "Learning to summarize with human feedback". In: *Advances in Neural Information Processing Systems*. Vol. 33. Curran Associates, Inc., 2020, pp. 3008–3021. URL: https://proceedings.neurips.cc/paper/2020/hash/1f89885d556929e98d3ef9b86448f951-Abstract.html (visited on 03/14/2024).
[40] Gemini Team et al. *Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context.* Apr. 25, 2024. DOI: 10.48550/arXiv.2403.05530. arXiv: 2403.05530[cs]. URL: http://arxiv.org/abs/2403.05530 (visited on 06/12/2024).
[41] C Thiel et al. "Radar remote sensing for the delineation of forest cover maps and the detection of deforestation". In: *Forestry: An International Journal of Forest Research* 79.5 (Dec. 1, 2006), pp. 589–597. ISSN: 0015-752X. DOI: 10.1093/forestry/cpl036. URL: https://doi.org/10.1093/forestry/cpl036 (visited on 01/25/2024).
[42] Louis L. Thurstone. "A Law of Comparative Judgment". In: *Scaling*. Num Pages: 12. Routledge, 1974. ISBN: 978-1-315-12894-8.
[43] Stéfan Van Der Walt et al. "scikit-image: image processing in Python". In: *PeerJ* 2 (June 19, 2014), e453. ISSN: 2167-8359. DOI: 10.7717/peerj.453. URL: https://peerj.com/articles/453 (visited on 03/11/2024).
[44] Pauli Virtanen et al. "SciPy 1.0: fundamental algorithms for scientific computing in Python". In: *Nature Methods* 17.3 (Mar. 2, 2020), pp. 261–272. ISSN: 1548-7091, 1548-7105. DOI: 10.1038/s41592-019-0686-2. URL: https://www.nature.com/articles/s41592-019-0686-2 (visited on 03/11/2024).
[45] Haoyu Wang et al. "Improvement of Region-Merging Image Segmentation Accuracy Using Multiple Merging Criteria". In: *Remote Sensing* 13.14 (Jan. 2021). Number: 14 Publisher: Multidisciplinary Digital Publishing Institute, p. 2782. ISSN: 2072-4292. DOI: 10.3390/rs13142782. URL: https://www.mdpi.com/2072-4292/13/14/2782 (visited on 02/03/2024).
[46] Zhaobin Wang, E. Wang, and Ying Zhu. "Image segmentation evaluation: a survey of methods". In: *Artificial Intelligence Review* 53.8 (Dec. 1, 2020), pp. 5637–5674. ISSN: 1573-7462. DOI: 10.1007/s10462-020-09830-9. URL: https://doi.org/10.1007/s10462-020-09830-9 (visited on 02/12/2024).
[47] Zuyuan Wang and Ruedi Boesch. "Color- and Texture-Based Image Segmentation for Improved Forest Delineation". In: *IEEE Transactions on Geoscience and Remote Sensing* 45.10 (Oct. 2007). Conference Name: IEEE Transactions on Geoscience and Remote Sensing, pp. 3055–3062. ISSN: 1558-0644. DOI: 10.1109/TGRS.2007.896283. URL: https://ieeexplore.ieee.org/abstract/document/4305379 (visited on 01/25/2024).
[48] Zuyuan Wang, Ruedi Boesch, and Christian Ginzler. "Forest delineation of aerial images with Gabor wavelets". In: *International Journal of Remote Sensing* 33.7 (Apr. 10, 2012), pp. 2196–2213. ISSN: 0143-1161. DOI: 10.1080/01431161.2011.608087. URL: https://doi.org/10.1080/01431161.2011.608087 (visited on 01/25/2024).

[49] Fabian L Wauthier, Michael I Jordan, and Nebojsa Jojic. *Efficient Ranking from Pairwise Comparisons.*

[50] Zhengzhe Wu et al. "ALS data based forest stand delineation with a coarse-to-fine segmentation approach". In: *2014 7th International Congress on Image and Signal Processing.* 2014 7th International Congress on Image and Signal Processing. Oct. 2014, pp. 547–552. DOI: 10.1109/CISP.2014.7003840. URL: https://ieeexplore.ieee.org/abstract/document/7003840 (visited on 01/25/2024).

[51] Scott W. H. Young. "Improving Library User Experience with A/B Testing: Principles and Process". In: *Weave: Journal of Library User Experience* 1.1 (2014). ISSN: 2333-3316. DOI: https://doi.org/10.3998/weave.12535642.0001.101. URL: http://hdl.handle.net/2027/spo.12535642.0001.101.

[52] Lefei Zhang and Liangpei Zhang. "Artificial Intelligence for Remote Sensing Data Analysis: A review of challenges and opportunities". In: *IEEE Geoscience and Remote Sensing Magazine* 10.2 (June 2022). Conference Name: IEEE Geoscience and Remote Sensing Magazine, pp. 270–294. ISSN: 2168-6831. DOI: 10.1109/MGRS.2022.3145854. URL: https://ieeexplore.ieee.org/document/9756442 (visited on 03/02/2024).

[53] Kaiguang Zhao and Sorin Popescu. *Hierarchical watershed segmentation of canopy height model for multi-scale forest inventory.* 2007.

A

Appendix 1

A.1 WKT representation of Figure 4.1

WKT
POLYGON ((595216 6760583, 595216 6761083, 594716 6761083, 594716 6760583, 595216 6760583))
POLYGON ((295389 6510885, 295389 6511385, 294889 6511385, 294889 6510885, 295389 6510885))
POLYGON ((530618 6757365, 530618 6757865, 530118 6757865, 530118 6757365, 530618 6757365))
POLYGON ((567520 6365474, 567520 6365974, 567020 6365974, 567020 6365474, 567520 6365474))
POLYGON ((665971 6611575, 665971 6612075, 665471 6612075, 665471 6611575, 665971 6611575))
POLYGON ((435114 6750524, 435114 6751024, 434614 6751024, 434614 6750524, 435114 6750524))
POLYGON ((439634 6481148, 439634 6481648, 439134 6481648, 439134 6481148, 439634 6481148))
POLYGON ((410190 6763578, 410190 6764078, 409690 6764078, 409690 6763578, 410190 6763578))
POLYGON ((511985 6357948, 511985 6358448, 511485 6358448, 511485 6357948, 511985 6357948))
POLYGON ((380479 6330684, 380479 6331184, 379979 6331184, 379979 6330684, 380479 6330684))
POLYGON ((639510 7167470, 639510 7167970, 639010 7167970, 639010 7167470, 639510 7167470))
POLYGON ((432492 6391505, 432492 6392005, 431992 6392005, 431992 6391505, 432492 6391505))
POLYGON ((610754 6654323, 610754 6654823, 610254 6654823, 610254 6654323, 610754 6654323))
POLYGON ((532774 6804598, 532774 6805098, 532274 6805098, 532274 6804598, 532774 6804598))
POLYGON ((549772 6785768, 549772 6786268, 549272 6786268, 549272 6785768, 549772 6785768))
POLYGON ((548580 6406637, 548580 6407137, 548080 6407137, 548080 6406637, 548580 6406637))
POLYGON ((398160 6148409, 398160 6148909, 397660 6148909, 397660 6148409, 398160 6148409))
POLYGON ((369465 6563082, 369465 6563582, 368965 6563582, 368965 6563082, 369465 6563082))
POLYGON ((795491 7378559, 795491 7379059, 794991 7379059, 794991 7378559, 795491 7378559))
POLYGON ((551597 6262049, 551597 6262549, 551097 6262549, 551097 6262049, 551597 6262049))

Table A.1: WKT string representation of each polygon surrounding rasters in Figure 4.1 from up to down and left to right. The coordinates are in SWEREF99.

DEPARTMENT OF MECHANICS AND MARITIME SCIENCES
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY