Tree Species Classification from Very High-Resolution UAV Images Using Deep Learning

Integrating Approaches Using Individual Tree Crowns for Semantic Segmentation

Master's Thesis for obtaining the degree of the Master of Science

Sustainable Resource Management

at the TUM School of Life Science at the Technical University of Munich

Supervised by Prof. Dr. rer. silv. Thomas Knoke

Professur für Waldinventur und nachhaltige Nutzung (TUM),

Dr. Jiaojiao Tian

Deutsches Zentrum für Luft- und Raumfahrt (DLR)

Submitted from Mareike Weishaupt

[redacted]

Submitted on December 20, 2024 in Munich

I confirm that this master's thesis in sustainable resource management is my own work and I have documented all sources and material used.

Munich, 20.12.2024 Mareike Weishaupt

Abstract

Very high-resolution red-green-blue (RGB) imagery captured by uncrewed aerial vehicles (UAVs) enables a new perspective on forest mapping, focusing on spatial information. For now, it is common to use satellite imagery using spectral information. However, this limits the research to forest areas rather than classification per tree due to low resolution and the inability to capture individual tree-level details. Driven by deep learning (DL), tasks like tree species (TS) classification using spatial data, such as the texture of a tree, are possible. This use of very high-resolution imagery for forest mapping has yet to be fully explored. This work uses the convolutional neural network (CNN) U-Net and a transformer model named Fully Convolutional Transformer (FCT) for semantic segmentation with the benchmark dataset "Bamforest" with a two cm resolution and 27,160 annotated trees within 105 hectares. In the summer of 2022, it was collected in Germany with a UAV. Ten classes were chosen to distinguish seven common German TS, coniferous, and deciduous TS, one combined class for all other trees, one for dead trees, and one for background. The objective is to develop a semantic segmentation model that accurately classifies each pixel into distinct classes. An individual tree crown (ITC) loss function is added to the training to improve results by using the ground truth of each tree crown to refine predictions through a voting process. In postprocessing, the semantic segmentation is combined with instance segmentation, which was previously trained separately. This process involves assigning the semantic segmentation results to each correct tree shape and deciding on one species using a voting mechanism, which helps refine the tree shape and species decision. The final results show the importance of data availability for each class with the F1-score and the intersection over union (IoU) and the advantage of using the second ITC loss for better classification with the disadvantage of additional computational time. The results show that the voting process in postprocessing has a positive effect on the output by comparing first, a test dataset similar to the training data and second, a different test area. The results support forest management, biodiversity assessment, and environmental monitoring by leveraging UAV and advanced DL techniques. The extensive dataset makes the generalization of models for other similar forest areas possible and creates the ability to generate single-tree information in dense forests.

Contents

1. Introduction

Tree species (TS) classification provides essential information on the distribution of TS, which is crucial for forest management. It can be used for various purposes, such as forest inventory (BMEL 2024), biodiversity assessment (Sun et al. 2019), and monitoring changes in forests, such as disease spread or shifts in species distribution. In Germany, for instance, monitoring the decline of spruce trees because of the beetle disease highlights the importance of TS classification (BMEL 2024). Recent advancements in remote sensing (RS) technologies and deep learning (DL) techniques have significantly advanced this field, mainly using Uncrewed Aerial Vehicles (UAVs) and high-resolution imagery.

TS can be classified using high-resolution red-green-blue (RGB) images, even in dense mixed forests. Research has shown that high spatial resolution data is sufficient for classifying TS, especially in environments with fewer classes or lower variability (Schiefer et al. 2020). Schiefer et al. (2020) successfully classified nine TS, while Ramesh et al. (2024) achieved classification for 13 species. However, multiple challenges persist. The rapid advancement of DL models has led to many new architectures. Yet, many of these models have not been thoroughly tested with high-resolution RGB imagery, particularly not for TS classification. Most research is limited to using convolutional neural network (CNN) architectures. As transformer-based and language models gain popularity, their potential for improving classification tasks in forest imagery remains underexplored.

Very high-resolution datasets like the "BAMFOREST" dataset (Troles et al. 2024), which is used for this research, are rare and focus primarily on RGB images, presenting both an opportunity and a challenge for DL models specialized for forest imagery. Additionally, many models currently used in TS classification are adapted from fields such as medicine or general object detection (Ronneberger et al. 2015), which necessitates fine-tuning to address the unique characteristics of forest imagery. Dense forests, in particular, present considerable challenges in distinguishing between many individual TS, further highlighting the need for more effective models. The challenge of classifying multiple TS from high-resolution datasets becomes more critical as the number of species increases.

Further research is necessary, especially in adapting and testing emerging DL on high-resolution forest datasets to overcome these challenges and improve classification performance. Given the challenges outlined above, this thesis aims to answer the following research questions:

1. How successfully can deep learning be used for tree species classification?
2. What percentage of training data per class is required to achieve robust classification of tree species, measured by intersection over union (IoU) and F1-score?
3. To what extent can individual tree crown (ITC)-based training and postprocessing enhance pixel-based tree species classification?

This research aims to develop an advanced segmentation framework for semantic segmentation, which involves classifying each pixel in an image into a specific category of TS in high-resolution RGB imagery, also referred to as classification. Specifically, the study will focus on accurately classifying ten classes, of which seven are TS. The data is from a dense natural forest in Germany. Two state-of-the-art DL models, U-Net and the transformer-based Fully Convolutional Transformer (FCT), are used for the classification. The research proposes a second individual tree crown-based loss function and a postprocessing technique with a ITC approach to achieve high classification performance, combining semantic and instance segmentation. It aims to enhance semantic segmentation by incorporating knowledge of the tree crown shape, improving model accuracy and scalability for reliable TS classification systems.

Improving TS classification is crucial for effective forest management and biodiversity monitoring. This research will contribute to more accurate and efficient forest inventories and ecological surveys by applying advanced DL models. The findings could also be used for similar efforts in other forest ecosystems, contributing to global biodiversity conservation by giving information on an individual tree basis.

The thesis is structured as follows: Chapter two provides a background on the key concepts and terms relevant to this work, including TS classification and DL principles. Chapter three presents a comprehensive overview of related work, exploring datasets, existing methods in TS classification, DL techniques, and associated technologies. Chapter four introduces the used dataset. Chapter five describes the methodology, including the DL networks (U-Net and FCT) and the proposed approaches for this research. Chapter six presents the experimental setup and implementation, explaining the design and execution of the experiments with additional explanations of terms needed. Chapter seven shows the final test results, whereas chapter eight discusses the findings, addressing the primary research outcome. Finally, chapter nine concludes the thesis with results, limitations, and future research directions.

2. Background

This chapter defines specific terminology and techniques used in this work and displays a foundation of what the work is based on. The topics RS platforms, spatial resolution, machine learning (ML) methods and classification types will be defined to provide an understanding of the usage of terms and meaning.

2.1. Remote Sensing Platform: Uncrewed Aerial Vehicle

There are many different RS platforms, from satellites to helicopters. However, a new attention-gaining technology has been used increasingly in recent years. UAVs, also known as drones, offer a middle ground between detailed ground-based surveys and traditional RS methods using aircraft or satellites. Their ability to fly low above ground and access difficult places allows them to generate very high-resolution images at a relatively low cost. They can collect high spatial and temporal resolution data, combined with access to hard-to-reach areas, providing valuable insights into forest ecosystems and enabling more accurate and thorough monitoring. Equipped with active and/or passive sensors, UAVs can safely, efficiently, and repeatedly capture continuous data on tree health (Ecke et al. 2024).

2.2. Spatial Resolution

The use of UAV allows the generation of a spatial resolution that is not imaginable with former satellite images. Therefore, the spatial resolution gains importance. Since this research focuses on classifying TS with very high-resolution images, the spatial resolution is a highly important factor of datasets. To distinguish between the different resolutions, different terms from Troles et al. (2024) will be used to define the ground sample distance (GSD) that are obtained by different RS platforms. Low resolution, typically associated with satellite imagery, is greater than 1.2 m per pixel. High resolution, commonly achieved using airplane-based sensors, ranges from 6 to 40 cm per pixel. Very high resolution, acquired through UAV platforms, is defined as less than 5 cm per pixel.

2.3. Machine Learning Background for Tree Species Classification

Before DL models became standard for applications like TS classification, other ML techniques were applied. Depending on the task, also until today, ML techniques like Random Forest Classifier are used, a learning method combining multiple decision trees to improve classification or regression accuracy (Zhang et al. 2021).

Current research focuses on applying DL models, either CNNs or vision transformers (ViTs) for TS classification. These DL networks are one of the fastest-changing influences in TS classification. For different purposes, there are different models. For this research, the focus is on differentiating between CNN and transformer architectures. CNNs are widely used for TS classification, such as image classification and object detection. Zhong et al. (2024) provide an overview of purpose and CNN model types, see Table 2.1. These models consist of two layers, called convolutional layers and pooling layers. The convolutional layers create the feature maps by sliding small filters, called kernels, over the input image. Each filter looks for specific patterns, like edges or textures, and then applies a non-linear function to add complexity to the features. This process is repeated in multiple convolutional layers. Afterward, pooling layers reduce the size of the feature maps by taking the maximum or average value from small areas, which helps simplify the information while keeping the important features. Finally, many CNN designs include fully connected layers after the convolutional and pooling layers. The last fully connected layer uses a softmax function to calculate scores for each object category, helping the network identify what is present in the image. However, they cannot capture long-range dependencies during training (Zhong et al. 2024).

Table 2.1.: Overview of CNN models by task from Zhong et al. (2024).

Task	**Models**
Instance Segmentation	YOLACT, Mask-R-CNN
Semantic Segmentation	PointNet, PointNet++, ResUNet, UNet, UNet++, DeepLab
Object Detection	YOLO, R-CNN
Classification CNN	Self-designed CNN, 3DmVNet, Attention-based CNN, Graph-based CNN, ModelNet, RetinaNet, Inception, EfficientNet, GoogleLeNet, ResNet, SqueezeNet, VGGNet, AlexNet, LeNet

ViTs, can model long-range dependencies in the images. ViTs have been initially developed for natural language models. They process images with sequences of patches. By dividing an image into fixed-size patches and embedding each patch into a vector, ViTs can capture global relationships and model long-range dependencies within the image. This makes it possible for the models to understand global relationships and model long-range dependencies within the image. This capability allows them to understand complex patterns and interactions across different image regions, making them particularly powerful for image classification and object detection tasks. As a result, vision transformers have demonstrated competitive performance against established CNN architectures, especially when trained on large datasets, paving the way for innovative approaches in computer vision (Aleissaee et al. 2023).

2.4. Classification Methods

TS classification can vary in several classification types. First, the essential terms of segmentation must be explained to further distinguish semantic and instance segmentation. The terms classification and semantic segmentation have the same meaning in this work.

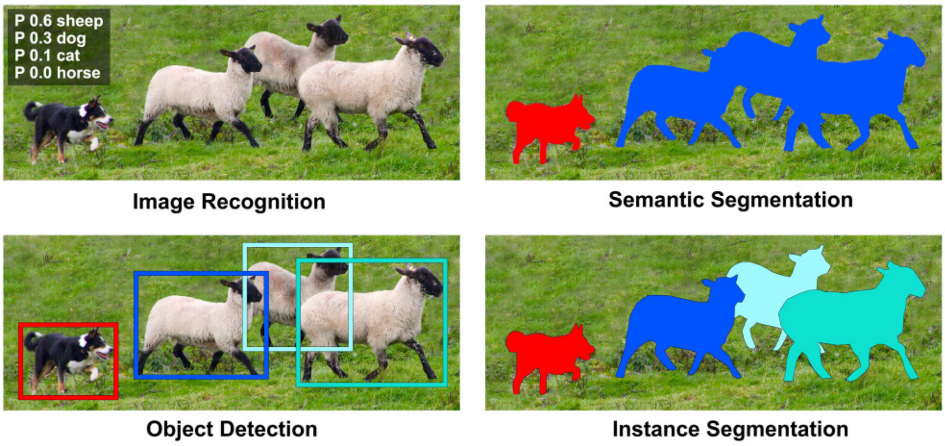


Figure 2.1.: An annotated image from the COCO dataset showcasing image-level labels, object-level bounding boxes, and segmentations at both class (semantic and instance) levels obtained by Tedrake (2024).

Figure 2.1 shows the difference between an instance and semantic segmentation needed for this work. Semantic segmentation refers to classifying classes on a pixel-based level. Each pixel is classified, not putting neighboring pixels into a relation. Classes are assigned to each pixel without distinguishing individual objects within a class. Meanwhile, instance segmentation uses an approach that can relate regions to objects, not only classifying different classes. This is done by adding the task of segmenting each object instance and separating individual instances of objects within the same class. This region-based approach can be understood as a tree-level segmentation (Lin et al. 2014). In the following, it will be continuously named ITC-based approach because the instance segmentation is used to classify individual tree crowns.

The choice of classification method is only one aspect to consider in segmentation. It is equally important to determine how the classes to be segmented are grouped. This is important to determine the task's difficulty, differentiating how many objects in what details are classified. The labeling of the data sets this differentiation. This can be categorized into three distinct approaches. In single-species classification, a single instance is classified per image. Multiple distinct instances are identified and classified within an image in multi-species classification. Finally, in region-based classification, numerous instances are grouped and analyzed as a single region.

3. Related work

This literature research aims to synthesize the current status on TS classification using RGB imagery and DL methodologies. TS classification has a wide field of applications, adapting to trends from other fields like sensor development or computer vision. Three review papers from Zhong et al. (2024), Chen et al. (2024), and Aleissaee et al. (2023) focus on TS classification and give a good overview of state-of-the-art papers and current development and trends from 1998 to 2023. The review output can divide research fields into categories displayed in Figure 3.1. These categories will be used to sort and arrange selected state-of-the-art papers. The literature research used papers focusing on TS classification of multiple trees in one image. Papers were searched with Google Scholar, Connected Papers, and Web of Science, using the terms "tree species classification," "deep learning," "tree species," and "tree crown." 14 papers have been selected according to the highlighted categories in Figure 3.1. The search was conducted with specific restrictions to ensure relevance and quality. It focused on studies discussing TS classification or segmentation, utilizing tree canopy images captured by satellites, airplanes, or UAVs, with a preference for very high-resolution images (< 5 cm spatial resolution). The research emphasized machine learning, particularly DL, and included publications from the past five years (published after 2019). Additionally, datasets centered on forests, with a preference for heterogeneous forest regions, were prioritized. The overview of all selected papers' specifications and sorted values can be found in the Appendix A.1.

3.1. Forest Datasets

Much of the data used for TS classification has been obtained from satellite images. However, there is a change since UAV can acquire very high-resolution imagery relatively cheaply. They make it possible to see detailed characteristics or traits in tree crowns, such as leaf and canopy shape as well as branching patterns (Fassnacht et al. 2016; Kattenborn et al. 2019; Natesan et al. 2020).

Datasets can vary tremendously depending on their research focus and location. A natural forest in Brazil differs significantly from a German natural forest; a plantation or an urban

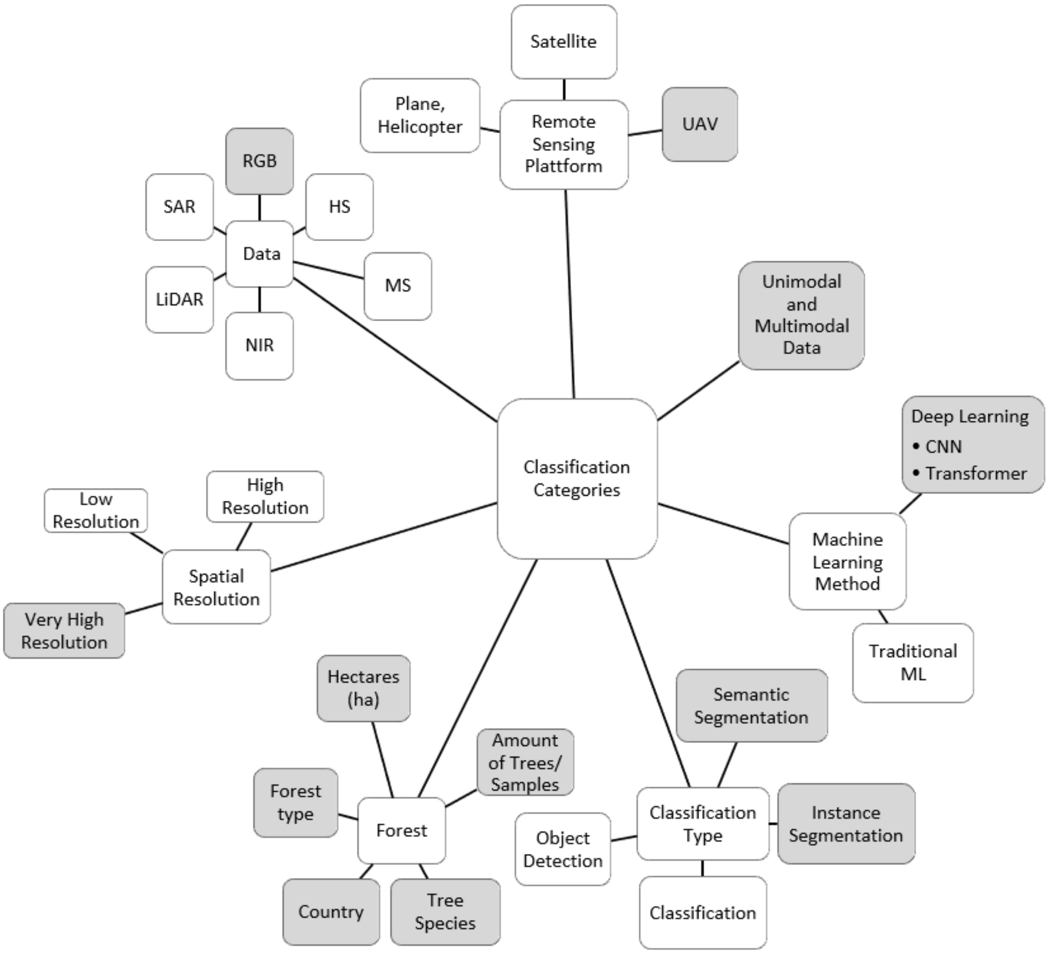


Figure 3.1.: Overview of categories that can be distinguished in TS classification. Highlighted items are categories that will be further investigated in the literature research. (HS = Hyperspectral, MS = Multispectral, NIR = Near Infrared, SAR = Synthetic Aperture Radar, ML = Machine Learning)

forest can be much easier to overview than a natural forest. Furthermore, depending on the purpose of the research, TS classification can be applied to single TS or multiple species. Pearse et al. (2021) detected only one species in various sites all over New Zealand to detect a disease spreading over the country. Whereas Ramesh et al. (2024) used a dataset from Quebec, Canada, to classify 14 TS in only one area for multiple species per image in a natural forest. Also Schiefer et al. (2020) classified 14 TS in Germany, Southern Black Forest in a temperate mixed forest. Sun et al. (2019) and Zhang et al. (2021) cover various sites in China, focusing on tropical wetlands and multiple sites in east China in urban and rural areas. They classified ten and 17 species.

The forest type is very important since it reflects the difficulty of classifying a forest. It is a different challenge to classify a natural forest, an urban forest (Ferreira et al. 2024) or a plantation (Cao and Zhang 2020; Cha et al. 2023). Since the focus is on achieving the classification of multiple TS, many papers use data from research (Egli and Höpke 2020) or natural mixed forests (Ecke et al. 2024; Nezami et al. 2020; Schiefer et al. 2020; Ramesh et al. 2024).

Another crucial factor to consider is the RS platform. There are different possibilities to obtain data, like satellite, plane, helicopter, or UAV. Historically, TS classification focused on classifying forest regions (Zhong et al. 2024). This is because the main source for the imaging was satellite imagery with a high spectral resolution. Other sources have been much more expensive than satellite images. But recently, the use of UAV has empowered data for TS classification since spatial resolution is very important to classify a tree in a dense forest. This is because it is impossible to now classify single TS in a dense forest with satellite images (Chen et al. 2024), and the UAV images are much cheaper than the ones taken by plains. Figure 3.2 displays the spectral resolutions used by the 14 selected papers. Since the focus is on TS classification, the majority used RGB data. However, some of the papers also use multimodal data. Investigating TS classification with different data sources to extract all important features and information of the trees is also a highly developing field (Nezami et al. 2020). Sun et al. (2019) combined RGB and Light Detection and Ranging (LiDAR). Also, Nezami et al. (2020) combined RGB with hyperspectral (HS) data.

Figure 3.3 shows which spatial resolution was used in each paper. It ranges from 1.6 cm to 60 m. The selected papers show a trend for a very high-resolution because of the usage of UAVs and the advantage of using the traits of the different species. Four papers use a dataset with around two cm resolution (Schiefer et al. 2020; Ramesh et al. 2024; Nezami et al. 2020; Egli and Höpke 2020). This data with a very high-resolution is taken by UAV RS platforms. This

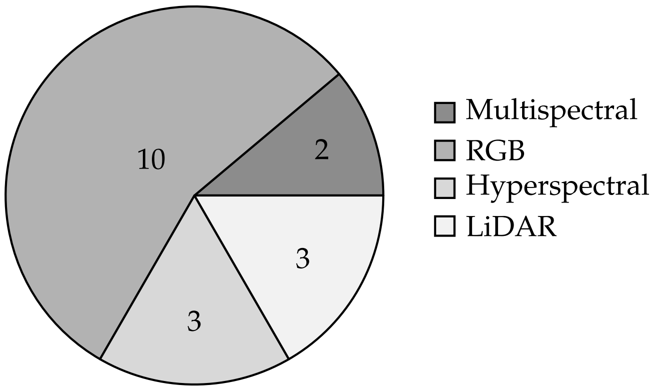


Figure 3.2.: Overview of the spectral resolutions utilized in the studies. Some integrated multimodal data by combining the spectral resolutions.

also shows the ability to classify TS with mostly high spatial resolution. Still, a research section is trying to perform TS classification with a lower resolution, combined with other approaches.

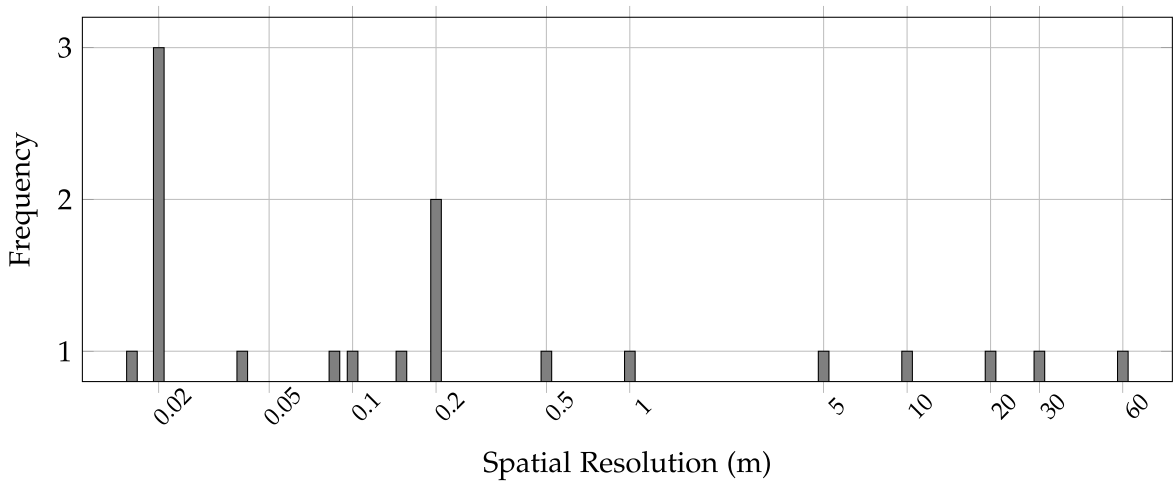


Figure 3.3.: Histogram of spatial resolutions utilized in the studies. Several studies use multiple spatial resolutions for their analysis.

3.2. Tree Species Classification

Chen et al. (2024) discuss the complexity of TS classification as one of the most challenging tasks in forest studies. The variety of vegetation and the structure of forests, with different heights, densities, morphology, health status, and the mixture of vegetation types, make the task challenging. Current research is very diverse in the number of TS classified by a DL model. Figure 3.4 shows the number of species classified. Most of the current research can classify three to six classes. Only Ramesh et al. (2024), Schiefer et al. (2020), and Sun et al. (2019) can classify 14 or 17 classes, from which nine classes are TS for Schiefer et al. (2020) and 13 classes for Ramesh et al. (2024). Their results are differentiating in performance. For Ramesh et al. (2024), the IoU varies from 30 percent to 79 percent with an overall result of 55 percent. Schiefer et al. (2020), using F1-sore as the result index, achieved values from 26 to 93 percent per class with an average of 69 percent as their best output for RGB image input. Schiefer et al. (2020) have classified TS using RGB and digital surface model (DSM), whereas Ramesh et al. (2024) classified with multi-temporal images. Sun et al. (2019) use RGB, LiDAR and DSM as input data for the training and also do not pursue a pixel-based segmentation of the trees. They rather classify forest regions according to the dominant species in the image patch.

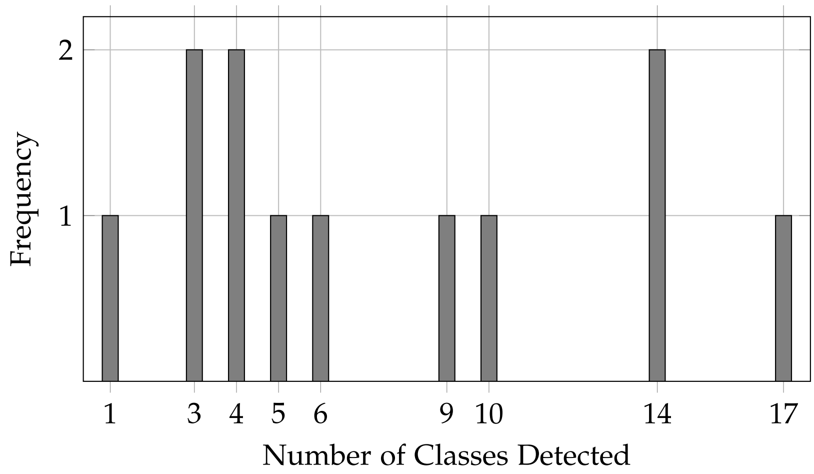


Figure 3.4.: Distribution of detected classes per study. In addition to TS, these classes may include various surrounding or environmental categories.

The classification focus is important because classifying a single species or a forest region is less complex than classifying multiple species. The majority of the selected paper in Figure 3.5 classifies multi-species (Beloiu et al. 2023; Ecke et al. 2024; Ferreira et al. 2024; Mäyrä et al. 2021; Nezami et al. 2020; Ramesh et al. 2024; Schiefer et al. 2020). One of the challenges of being able to classify multi-species is that the ground truth provided with the dataset must be from experts, especially in dense forests. Multi-species classification will be needed to understand a natural mixed forest since knowing about a single tree within its environment is important.

Unimodal data refers to the use of a single input data type. For example, using RGB data is unimodal. Further, there is mulitmodal data. Image processing or RS refers to multiple input data types. For example, RGB data and LiDAR data. Another aspect of multimodal data is multi-temporal data. This can include only one data type but from multiple input image sources, taken at different times (Zhong et al. 2024). Many current studies use unimodal data (Pearse et al. 2021; Ferreira et al. 2020; Egli and Höpke 2020; Cao and Zhang 2020; Zhang

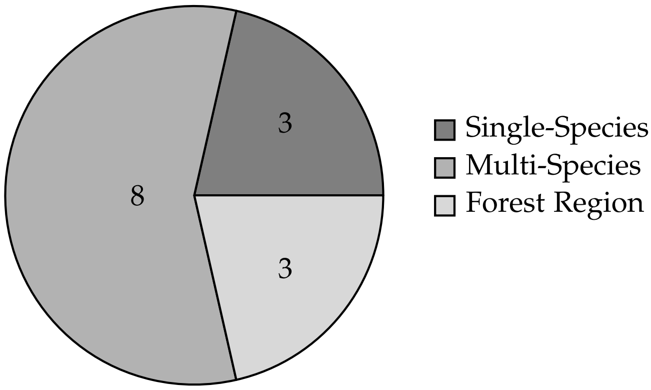


Figure 3.5.: Classification focus of the papers according to number of species classified or region classification.

et al. 2021). However, there is a trend using very high-resolution RGB images combined with additional information using multimodal training. Several approaches have emerged in the literature. For example, some studies combine multitemporal very high-resolution RGB images captured during different seasons to provide varied inputs (Ramesh et al. 2024). Others integrate hyperspectral images with LiDAR data, leveraging 3D CNN for improved analysis (Mäyrä et al. 2021). Another approach compares the effectiveness of RGB, canopy height model (CHM), and hyperspectral data using DL models (Nezami et al. 2020). Additionally, some research combines RGB and DSM derived from LiDAR to capture both color information and tree height (Sun et al. 2019). These methods underscore the potential of multimodal data integration to enhance the accuracy and robustness of tree classification systems.

3.3. Deep Learning Models

In Figure 3.6, the used DL models of the literature research are provided. Some models are fundamental and often used for TS classification. ResNet and U-Net are good examples of CNNs often used for semantic segmentation. They are robust and have been implemented for similar tasks for years. Similar models are AlexNet, VGG-16 and 3D-CNN. However, some new models use a transformer-based approach. Also, in this literature research, nearly only CNN models have been selected, although ViT models or others are up on the rise. That is because integrating newly developed models into forest applications takes time (Zhong et al. 2024).

DeepLab3+ Chen et al. (2018) was used by Ferreira et al. (2020) as a state-of-the-art deep learning model for semantic image segmentation. This hybrid model combines CNN and a transformer model. Further, Mask2Former (Cheng et al. 2022) and U-Net with Temporal Attention Encoder (U-TAE) (Fare Garnot and Landrieu 2021) have been used by other papers,

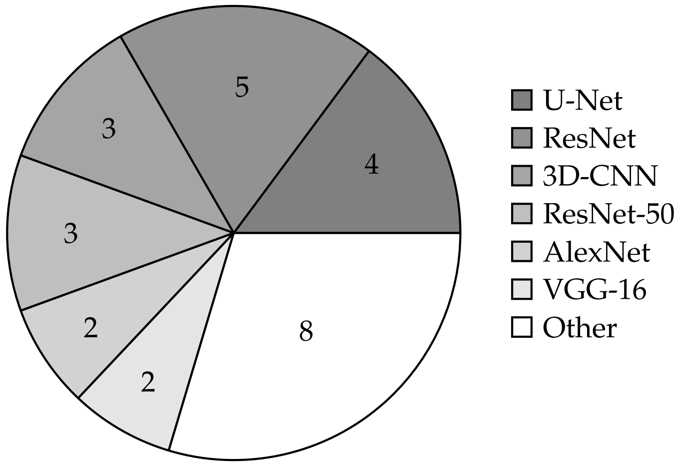


Figure 3.6.: DL models employed in the studies, with some papers utilizing multiple models.

being vision transformer models.

Most papers used one or two models to test the dataset or determine which combination of data inputs was the best. Others compared different models against each other. Ramesh et al. (2024) has one of the most focused approaches on using DL. They trained U-Net, DeepLabv3+, Mask2Former, 3D-UNet and U-TAE. U-Net, with a ResNet-101 backbone, was performing best on the dataset. As formerly described in Table 2.1, depending on the task, different models have been used. For semantic segmentation, many papers use U-Net (Schiefer et al. 2020; Cha et al. 2023; Cao and Zhang 2020).

4. Dataset

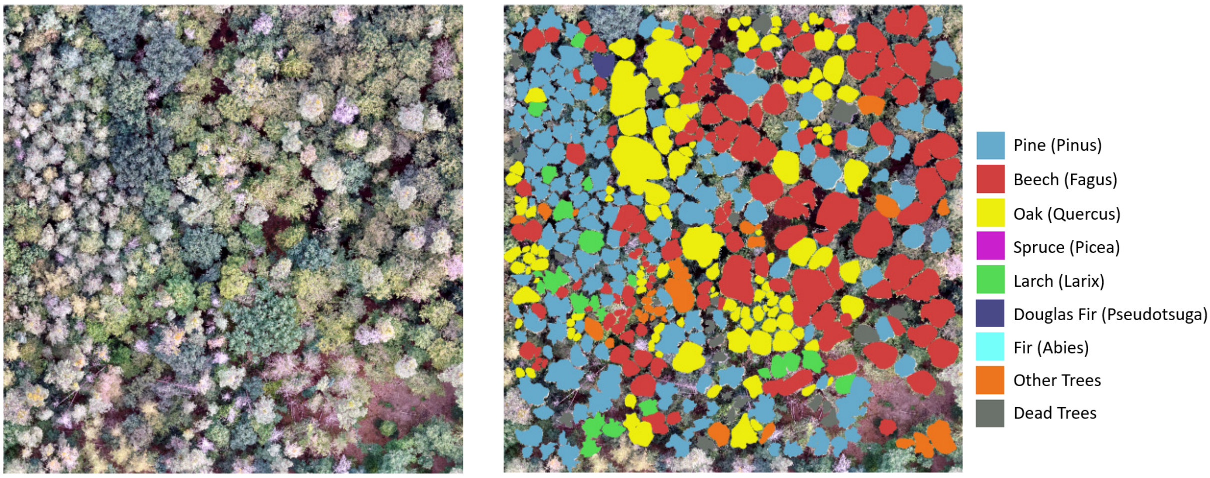
The dataset is a key component in TS classification, as its quality and structure significantly influence the performance of the models. This chapter provides an overview of the dataset specifications, including the types and functions of the data files, as well as the preprocessing steps applied to prepare the data for analysis.

4.1. Dataset Description and Tree Species Analysis

The dataset "BAMFOREST" used for this research was obtained by Troles et al. (2024). It is a dataset captured with a UAV with a very high resolution of about 2 cm from two drones with different sensors. The captured forest is in the area around Bamberg with three different forest areas (Hain, Stadtwald & Tretzendorf), which is a total of 105 hectares. An expert has labeled 27,160 individual delineated tree crowns for all regions. The data acquisition of all images in BAMFORESTS occurred in the summer of 2022. Both Tretzendorf areas of interest (AOIs)s were captured on 5 July, the Stadtwald AOI on 6 July, and the Hain AOI on 3 August. The dataset comprises images from three areas, featuring a GSD of 1.82 cm for Hain, 1.7 cm for Stadtwald, 1.61 cm for Tretzendorf-1-AOI, and 1.79 cm for Tretzendorf-2-AOI, with a total of 15 images from Hain, 46 images from Stadtwald, and 44 images from Tretzendorf. The Hain area is only used for testing since its tree distribution varies a lot from the other. The Hain area contains primarily deciduous TS, opposite the other two regions. The test data consists of two different test regions, one within the training region (Test-2) and one from a different area (Test-1).

Further, the number of trees and the percentages of each species are displayed. Figure 4.1(a) depicts one example image of the training data. It shows the dense forest, with only some background in between. Also, the varying size of the trees can be seen. In Figure 4.1(b), the corresponding detailed labels can be seen. Besides, the image indicates that not all trees have been labeled due to the image corners or the size of the trees.

Table 4.1 displays the TS labeled in the dataset. The table contains the number of trees and species distributed in the dataset through training, validation, and test data. The share of each TS is also shown, and the vitality status of the trees with vital, degrading, and dead is also



(a) Input image (b) Ground truth

Figure 4.1.: Visualization of an image from the Stadtwald region. Input image (a) and ground truth (b) display an example of the input data used for this work.

displayed. Pine, Beech, Oak, Spruce, Larch, Douglas Fir, and Fir are the species that will be classified. Other species that are very low in the share or trees that could not be classified are summarized in one class as "Other Trees".

All TS and their traits can be seen in Figure 4.2. The structure and texture of the different species appear very differently. Beech (b) and Oak (c) have a more evenly dense tree canopy because they are the broad-leaved deciduous species. The coniferous trees have a sparser tree canopy with more structure of the tree shape. Spruce (d), Douglas Fir (f), and Fir (g) have a star-like shape. The texture and shape are the first factors of texture that are interesting for the TS classification. The second factor is the different colors of each TS. They appear as shades of green to the eye, but as seen in Figure 4.2, they vary greatly. But in general, not every tree is entirely different from the others. Since many TS are firmly related, some appear very similar, such as the Fir (Abies) and the Douglas Fir (Pseudotsuga), and share similar characteristics.

Another factor regarding the trees' traits is the TS' size. As displayed in Table 4.2, the relation of the number of trees to the number of pixels per tree is different. This means some trees grow larger and more prominent, and some remain relatively small. Comparing the number of trees to the share of pixels, it can be seen that deciduous TS are relatively larger than the coniferous species. Also, trees of the same species appear in various sizes, according to their age and surrounding factors like light supply, nutrients, and competition with other trees (Burslem and Swaine 2003). This leads to the factor of the status of each tree. Trees of the same species can look very different because of their age and vitality. This causes the same class to vary a lot in terms of size and color.

Table 4.1.: Summary of tree distribution in the dataset according to the classes. The percentages are according to the number of trees after Troles et al. (2024).

	Train-Set	**Val-Set**	**Test-Set-2**	**Test-Set-1**
N of trees shapes	17,212	4,390	3,580	1,978
Pinus (Pine)	36.23%	31.94%	27.54%	1.11%
Fagus (Beech)	23.11%	20.71%	23.10%	23.96%
Quercus (Oak)	23.30%	20.27%	22.43%	19.01%
Picea (Spruce)	5.53%	7.22%	9.11%	1.06%
Larix (Larch)	2.70%	1.75%	2.21%	1.26%
Pseudotsuga (Douglas Fir)	1.12%	1.80%	1.20%	0.15%
Abies (Fir)	1.19%	1.12%	1.01%	0.00%
Other	6.83%	15.19%	13.41%	52.88%
Vital	86.34%	84.99%	83.97%	91.20%
Degrading	11.88%	12.35%	13.18%	8.49%
Dead	1.78%	2.67%	2.85%	0.30%

4.2. Data File Types

The dataset utilized for TS segmentation and refinement consists of several vital data file types, each serving a distinct role in the training and postprocessing phases of the model. These data types include input image files, raster files (TIFs), ground truth shape files, shape files (SHPs), and single shape JSON files, each contributing unique information critical to accurate tree segmentation.

The TIF files serve as the primary input images, capturing high-resolution visual data of forested areas. These images include detailed depictions of tree variations, such as size, shape, and density, and are essential for training segmentation models. Ground truth data for these images is provided by the SHP files, the geospatial vector data. It provides precise information about tree characteristics, including size, shape, species, vitality, and more. Figure 4.1(b) illustrates an example of the SHP data, with species and colors in the accompanying legend. The SHP files ensure accurate class and shape annotations, forming the basis for model validation and performance assessment. Labeling the trees correctly is crucial for successfully segmenting the classes for the training. However, labeling a forest is a challenging task with many obstacles. First, the tree crowns are tough to separate since they overlap. This causes the labels to also overlap. The labels for this dataset are labeled to the best knowledge, but some are missing, and some are misclassified.

In addition to the TIF and SHP files, individual tree shapes are represented in single-shape

(a) Pine (Pinus)

(b) Beech (Fagus)

(c) Oak (Quercus)

(d) Spruce (Picea)

(e) Larch (Larix)

(f) Douglas Fir (Pseudotsuga)

(g) Fir (Abies)

(h) Other Trees

(i) Dead Trees

Figure 4.2.: Presentation of all nine tree classes (excluding background) in the dataset. Class eight (h) is only one example tree of many different.

JSON files, which provide a focused view of tree structures within image patches. These JSON files encode single tree shapes in a binary format, where a value of one indicates the presence of a tree, and a value of zero represents the background. The JSON files serve two primary functions: First, during the training phase, they are incorporated into the model's loss function to improve the accuracy of tree shape representations. This data is directly extracted from the ground truth. Second, after training, they are used for postprocessing from the output of an instance segmentation to refine the semantic segmentation results, thereby enhancing the overall model performance.

4.3. Data Preprocessing

Since the dataset "BAMFOREST" contains more information than needed, some modifications have been made to use it. Since not all TS and the information provided will be used, the species and the information selected have been categorized. Therefore, the correct classes were selected, and the SHP files were modified to extract only the needed information. Table 4.2 shows the dataset's distribution of each TS. First, the dataset was trained with only the TS, but since the dead trees can not be distinguished in their TS, they were later restructured as their own class.

Table 4.2.: Summary of classes used with common names, and pixel percentages with adjusted pixel percentages relative to TS. (*Dead trees are within the TS but were not counted into species percentages. The percentage of dead trees is from the percentage of dead trees against vital ones.)

Class	**Tree Type**	**Name (Latin Name)**	**Distribution of TS (%)**		**Overall Pixels (%)**
			Number of Trees	**Pixel-based**	
0	-	Background	-	-	45.61
1	Coniferous Trees	Pine (*Pinus*)	31.8	26.59	14.46
2	Broadleaved Trees	Beech (*Fagus*)	22.8	27.73	15.08
3	Broadleaved Trees	Oak (*Quercus*)	22.4	25.53	13.88
4	Coniferous Trees	Spruce (*Picea*)	5.95	4.56	2.48
5	Coniferous Trees	Larch (*Larix*)	2.37	1.89	1.03
6	Coniferous Trees	Douglas Fir (*Pseudotsuga*)	1.17	0.81	0.44
7	Coniferous Trees	Fir (*Abies*)	1.07	0.35	0.19
8	-	Other Trees	12.4	11.53	6.27
9	-	Dead Trees	1.96*	1.01	0.55

The distribution of the number of TS was provided in Troles et al. (2024), but because trees vary much in size, the pixel-based distribution was also calculated to show how much weight the TS have in training. As displayed in Table 4.2, when class Pine and Beech are compared, the number of Pine trees is much higher than for beech trees. But if their pixel-based share is compared, it is nearly the same. Also, the overall pixel distribution was calculated to see how much of the data is background. The ten classes are very differently distributed because it is a natural forest. Considering this is a pixel-based approach, this is essential information.

Since the original data contains four channels, the TIF files have been modified only by extracting the three RGB channels. The SHP files have been modified since they contain much information like species, polygon shape, an ID, and vitality. The classes have been assigned from zero to nine with the according species or background (see Table 4.2). Then, the SHP files were modified so that only the class species is used as a feature. Additionally, the SHP files have been converted to TIF files to make them usable as the ground truth. The conversion output is a TIF file containing pixel-wise information about the decided species. The JSON files were previously converted as TIF files, and the values were then modified by setting them to one and zero to be directly used in further processes.

The last data preparation step was to create patches from the original images to make DL possible. Two patch sizes have been extracted. First, patches of 2048 x 2048 pixels have been created with a stride of 50 percent. The patch size of 512 x 512 pixels with a stride of 50 percent was also implemented. With 2048 x 2048, more complete tree shapes per image can be located, but for 512 x 512, the texture and features of each tree are easier to see since the 512 x 512 patch only displays 1/16 of the size of the other patch. In Figure 4.3, the two tile sizes are displayed from the same image. This makes the size of 512 x 512 beneficial since the tinier patch size can better learn the texture and pattern of the trees.

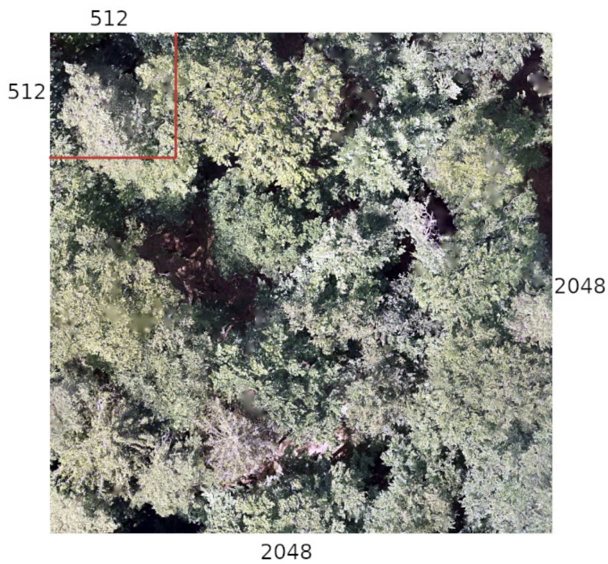


Figure 4.3.: Images sizes after preprocessing the images with creating patches, comparing the size difference of a 512 x 512 to a 2048 x 2048 patch.

5. Methods

The methods provide a comprehensive overview of the model development. It details the baseline networks and the idea and implementation of the approaches, focusing on implementing a ITC aware training and postprocessing.

5.1. Baseline Semantic Segmentation Architectures

Two architectures were used to train the dataset. Since the dataset "BAMFOREST" has been newly introduced, a fundamental and robust CNN model, U-Net, was selected as an initial approach. A state-of-the-art transformer model, FCT, was chosen to leverage its advanced capabilities. Both models originated in biomedical imaging, making them well-suited for detailed segmentation tasks like TS classification.

5.1.1. U-Net

The CNN architecture U-net was adapted for the TS segmentation after Ronneberger et al. (2015). Figure 5.1 shows the U-Net architecture. The blue boxes correspond to a multi-channel feature map with the number of channels denoted on top of the box. The x-y-size of the image is shown at the left bottom edge of the blue box. The white boxes can be referred to as the copied feature maps. In the legend, the arrow functions can be derived.

U-Net is a CNN that holds an Encoder-Decoder structure. The encoder uses convolutional layers followed by max pooling for downsampling. The decoder is the upsampling path that enables precise location through the transposed convolutions. The skip connections between the encoder and decoder layers preserve the spatial information and gradients. The standard convolutional layers (3 x 3) are used for feature extraction. After each convolution, the activation function (ReLu) is applied. Also, max-pooling layers are used in the encoder to reduce the spatial dimensions. Then, the final layer, usually a 1x1 convolution, is used to map feature maps to the desired number of output classes (Ronneberger et al. 2015).

For this TS classification, minor changes were made to adapt the model to the desired output since the model was adopted from a binary semantic segmentation with different image input

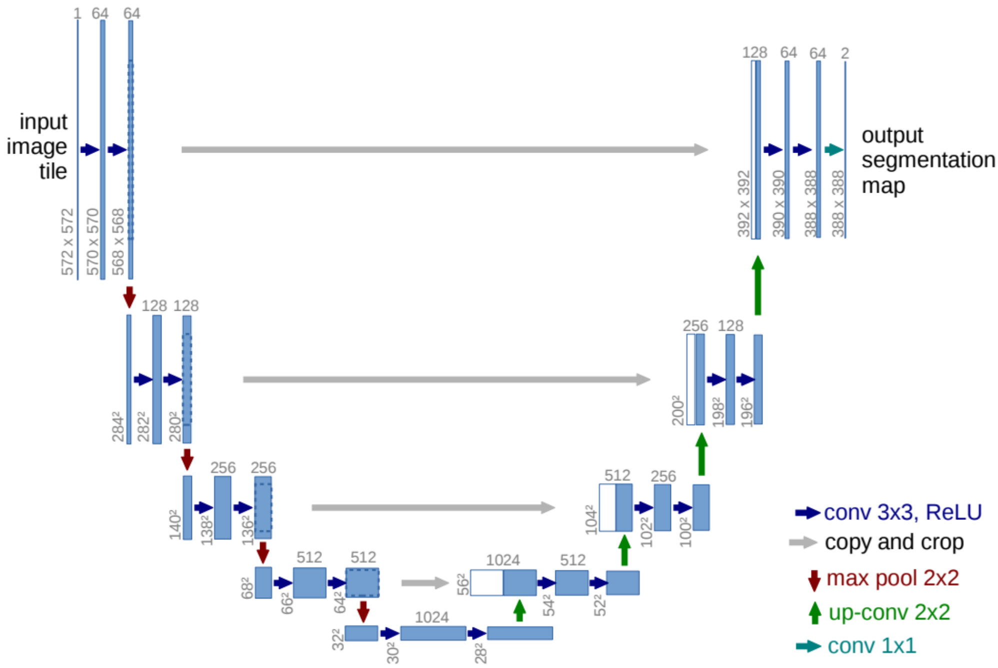


Figure 5.1.: U-Net model architecture after Ronneberger et al. (2015).

sizes. The input size was alternated from 2048 x 2048 pixels, and 512 x 512 pixels according to the input image size. With the input image size of 512 x 512, the downsampling is performed from 512 x 512 to 256 x 256 until 32 x 32 for the bottleneck, going upsampling again until the original input shape 512 x 512. The final output layer was also modified to classify the desired number of output classes. This was changed from a binary output to the first nine and then ten classes. The loss function was originally a Binary Cross-Entropy (BCE) loss and was changed to a categorical Cross-Entropy (CE) loss. The final layer added a softmax activation to obtain probabilities for each class.

5.1.2. FCT

Since U-Net is a CNN introduced in 2015, a newer transformer model is used for comparison. Therefore, the 2023 published FCT was chosen, displayed in Figure 5.2 with the transformer part on top and the U-Net structure on the bottom. It combines a CNN with a transformer model to leverage local and global feature extraction. The convolutional layers consist of a standard convolutional block to capture spatial features. However, there are also transformer blocks with attention mechanisms that allow the model to weigh the importance of the different spatial regions across the input image. These consist of multi-head self-attention layers, enabling the model to simultaneously attend to various parts of the image. A positional encoding is added to the input of the transformer layers to retain spatial information since transformers are permutation-invariant. As well as for U-Net, skip connections can enhance feature propagation for earlier layers. The decoder structure is built so that the output from the transformer blocks can be passed through additional convolutional layers for upsampling and reconstruction (Tragakis et al. 2023).

Similar changes to U-Net have been made for the FCT architecture. The input layer was adapted according to the input dimensions of the input image based on the patch size of the dataset. The output layer has also been modified using nine or ten channels for the TS classes. The CE loss function has been used, and softmax has been implemented as the activation function. Additionally, to simplify the model, the output was modified to use only one output instead of the proposed three outputs for deep supervision. Another important factor is using the attention head and filters for the transformer part of the FCT. Attention heads can simultaneously focus on different parts of the input image to capture various relationships. Changing the number of heads can help balance the model's complexity and performance. For this training, it was alternated between the following:

```
attention heads 1 = [2, 2, 2, 2, 2, 2, 2, 2, 2]
attention heads 2 = [2, 4, 8, 12, 16, 12, 8, 4, 2]
attention heads 3 = [2, 2, 4, 4, 8, 4, 4, 2, 2]
```

Additionally, the filters have been changed. The filters identify patterns, texture, and structure in the image. The filters can be modified in their size or number. It can enhance feature extraction and avoid overfitting the model. The following filters have been used:

```
filters 1 = [8, 16, 32, 64, 128, 64, 32, 16, 8]
filters 2 = [32, 64, 128, 256, 512, 256, 128, 64, 32]
filters 3 = [16, 32, 64, 128, 384, 128, 64, 32, 16]
```

5.2. Pixel-Based Semantic Segmentation Using U-Net and FCT

Pixel-based semantic segmentation is applied to very high-resolution images to classify each pixel into predefined categories, enabling precise TS classification. This process is critical for tasks such as tree crown segmentation, where accurate delineation of boundaries and handling overlapping structures are essential. U-Net and FCT are employed to achieve these goals, complementing each other with their respective strengths.

U-Net is particularly effective for tasks requiring detailed spatial accuracy. Its encoder-decoder architecture with skip connections ensures that fine-grained spatial features are

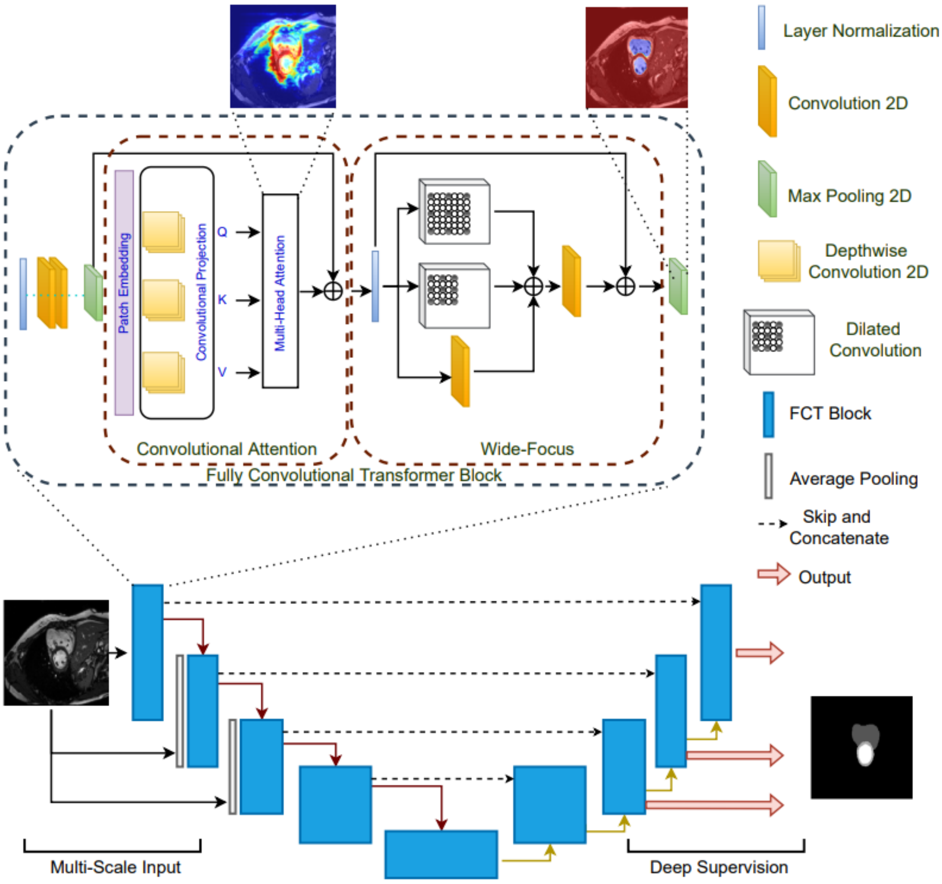


Figure 5.2.: FCT model architecture after Tragakis et al. (2023).

preserved while capturing global context, as described before. This capability makes it well-suited for segmenting tightly packed tree crowns or distinguishing individual objects within dense regions. By leveraging its ability to combine local details with semantic features, U-Net generates segmented masks that align closely with the boundaries of the objects, even in complex and cluttered areas.

Conversely, FCT excels in capturing global dependencies across the entire image, addressing challenges such as overlapping or similarly colored tree structures. Through self-attention mechanisms, FCT identifies long-range relationships between pixels, crucial for distinguishing objects where local features alone are insufficient. Positional encodings ensure spatial coherence, accurately segmenting large or complex forested areas. The combination of global context and pixel-level precision allows FCT to produce segmentation masks that are robust to variations in scale and texture.

5.3. Combined Pixel- and ITC-Based Segmentation

Including ITC-based segmentation approaches extends beyond pixel-based methods by integrating ITC-aware strategies to enhance accuracy and robustness. Building on the best-performing pixel-based model U-Net, additional methods are implemented to incorporate region-level information. This includes using adapted ITC segmentation results for postprocessing and refining segmentation boundaries through a ITC-based approach. Furthermore, an accuracy loss function, referred to as ITC loss, is introduced, utilizing ground truth data to better learn the segmentation process. This combined approach bridges pixel-level precision with ITC-aware contextual understanding, improving segmentation outcomes in complex scenarios.

5.3.1. Combination of Pixel-Based Classification and ITC Segmentation in Postprocessing

This is the first approach to optimizing the model performance. After training and applying the model to the testing data, a postprocessing procedure is applied to the testing data to refine the initial predictions and improve accuracy in identifying individual TS. This postprocessing step leverages the results of instance segmentation, called tree crown segmentation, obtained by Tian et al. (2025). The instance segmentation segments each object instance and separates individual instances of objects within the same class. The instance segmentation was trained on the same dataset, "Bamforest," to segment all tree crown shapes. With the output, the individual tree crowns are isolated by combining their shapes with the class predictions generated from

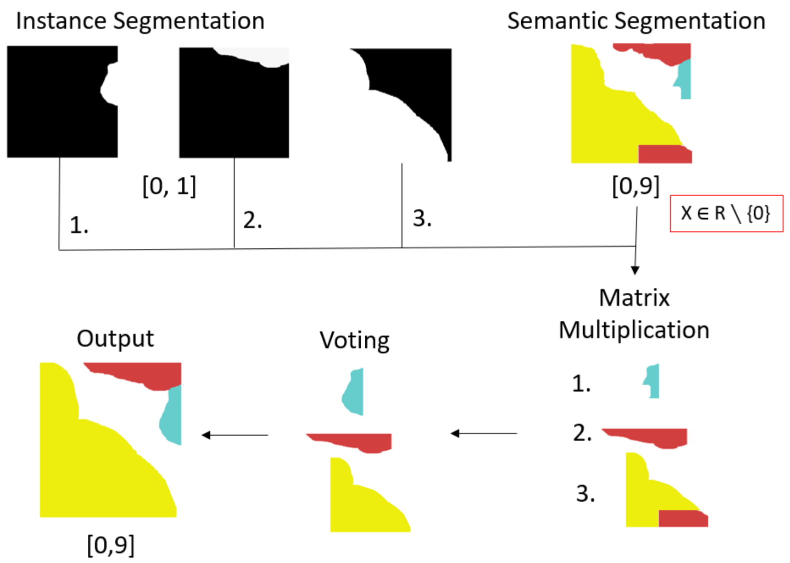


Figure 5.3.: Workflow diagram combining instance and semantic segmentation with matrix multiplication and evaluating the right class per shape with voting.

semantic segmentation, which can be seen in Figure 5.3. This approach enables the model to produce accurate species classifications and spatially distinct tree identifications.

Each tree crown shape, identified through instance segmentation, is associated with the species predicted through semantic segmentation. Integrating the spatial and classification outputs ensures that each tree is represented by its location and species classification. This integration is particularly valuable for complex and dense forested areas, where individual tree crowns must be distinctly identified and accurately classified, challenging only by semantic segmentation. Combining ITC segmentation and semantic segmentation requires an additional data input derived from the instance segmentation output. Specifically, the instance segmentation produces ITC shapes saved as JSON files representing each tree's boundary. These JSON files are converted into TIF images with the tree crown converted to "ones" and the other area as "zeroes" to create a matrix format compatible with the semantic segmentation output. The converted TIF image matrix is then multiplied with the semantic segmentation output matrix to link each tree crown shape to its predicted class. The output will only be the tree crowns containing the predicted classes. Next, a voting procedure is applied to assign a species label to each tree crown based on most pixels within its shape. This voting step refines the class prediction by considering all pixels within each tree crown shape, ensuring the final prediction is consistent and aligned with the dominant species for each tree crown. This step will be repeated for each ITC. Finally, the individual predictions for each tree are combined to produce a single, unified output image representing the classified TS across the entire area.

Two types of voting mechanisms for the segmented tree crowns were evaluated to achieve a robust classification for each identified tree shape.

The first approach is Majority Voting. It identifies the species for each tree crown based on most pixels within its shape. In majority voting, the species label assigned to the tree shape is determined by the most frequently occurring class among its pixels. This method assumes that each crown's most common pixel label corresponds to the correct species, providing a straightforward and effective way to aggregate pixel-level predictions for each tree.

The second approach, Weighted Majority Voting, applies a weighted majority voting scheme to address class imbalances in the dataset. In this method, each class is assigned a weight based on its frequency in the dataset, giving minority classes a higher relative weight. When calculating the majority class for each tree crown, these weights are factored into the count of pixels for each species, effectively boosting the likelihood that minority species are correctly identified. This weighted approach is instrumental in ensuring fair representation of less common species, preventing them from being overlooked due to lower pixel counts.

The instance segmentation process used two image patch sizes to evaluate the pest option: 512 x 512 pixels, and 2048 x 2048 pixels. The 512 x 512 patches were chosen to match the image patch size of the semantic segmentation, enabling easy integration and straightforward implementation. This size proved advantageous for maintaining consistency with the existing semantic segmentation framework and required minimal adjustments to the workflow. In contrast, the 2048 x 2048 patches were selected to incorporate more extensive context areas, mainly to capture more complete tree shapes within a single patch. This has benefited the ITC segmentation before. The semantic segmentation results had to be adapted to fit the new dimensions and align with this larger patch size, introducing an additional processing step. However, this adjustment allowed for more holistic representations of tree structures, which is particularly beneficial in tasks that rely on capturing the overall morphology of trees. Thus, the choice of patch size involved a trade-off between implementation simplicity with smaller patches and the ability to capture broader contextual information with larger patches. Both approaches were tested to identify the better approach. The results using 2048 x 2048 patches created much better ITC shapes because bigger trees could be segmented. Therefore, this size was chosen for the postprocessing.

5.3.2. ITC Loss Function

An end-to-end DL architecture is a second approach that proposes an ITC loss function, which helps to improve the learning in the training. This approach helps to improve the training

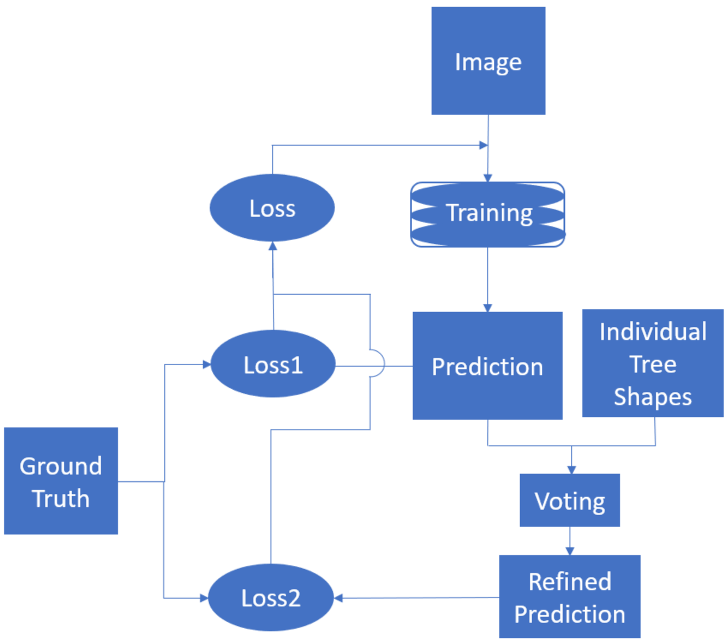


Figure 5.4.: Workflow diagram of implementing the ITC loss (Loss 2) and the combination with the CE loss (Loss 1).

of the baseline architecture. The ITC-based loss was incorporated into the loss function of the existing model. The idea is to incorporate a second loss function based on the tree crown segmentation. The proposed method utilizes tree shapes from the ground truth provided as individual JSON files. These JSON files represent each tree's shape and can be converted into TIF files for processing. Each shape generates a single-class prediction with a majority voting, subsequently compared with the model's output.

As displayed in Figure 5.4, the procedure begins with generating predictions. The first loss is calculated by applying a CE loss to the initial model output. The predictions generated during training are then refined using the tree shapes derived from the ground truth extracted files. This refinement is achieved by multiplying the predictions with the corresponding tree shapes, thereby isolating the predictions specific to each tree. This can be done since the JSON files contain only zeros and ones, only differentiating for a tree or no tree.

Subsequently, a majority voting mechanism determines the dominant class within each isolated tree shape, displayed in Figure 5.5. This ensures the entire shape is assigned a single class since a tree can only belong to one species. This process is repeated for all tree shapes, and the resulting predictions are aggregated into a complete image. An example of this refinement output can be seen in Figure 5.5b, derived from the prediction in Figure 5.5a.

The refined predictions are then compared to the ground truth to evaluate the consistency of shapes and species classification. The number of shapes not overlapping with the ground

(a) Prediction output from training.

(b) Refinement from shaping and voting.

(c) Ground truth for refinement and to calculate loss function.

Figure 5.5.: Output comparison of implementing voting for refinement during training.

truth classes is quantified as one, and all true predictions are zero. The overall sum of pixels is summed up, as well as the sum of the wrongly predicted classes. The final accuracy is calculated, yielding a value between zero and one, described in Equation 5.1 as the ITC Loss (L_{ITC}).

$$L_{\mathrm{ITC}} = \frac{\text{Number of False Predictions}}{\text{Total Number of Predictions}} \tag{5.1}$$

This second ITC-based loss is then summarized with the first pixel-based loss to one total loss. This total loss Equation 5.2 combines the CE loss and the ITC loss. After the loss is calculated, it is fed into backpropagation.

$$\text{Loss} = \alpha \times L_{\mathrm{CE}} + \beta \times L_{\mathrm{ITC}} \tag{5.2}$$

The parameters can be weighted differently with α and β, allowing to balance out the impact of each loss. This is another crucial step since the CE loss is higher than the ITC loss, and it should be evaluated to what extent both interact the best.

5.3.3. Strengths of The Approaches

The proposed methods showcase significant strengths that contribute to their effectiveness and potential impact. The methods address key challenges in very high-resolution TS classification by leveraging innovative techniques and targeted optimizations. It demonstrates notable strengths in advancing semantic segmentation within the context of a novel benchmark dataset.

A key innovation lies in incorporating a postprocessing step that transitions from semantic to ITC-based instance segmentation. This step facilitates the extraction of individual tree crowns and enables a species-level classification by performing a voting mechanism on each tree instance. The ability to identify and classify individual trees makes the approach particularly relevant for applications in forestry and biodiversity monitoring.

By integrating an ITC-based loss function, termed ITC loss, the approach effectively refines segmentation precision, particularly by focusing on the structural coherence of individual tree crowns. This loss function enhances the model's ability to delineate distinct tree regions, a critical requirement for high-resolution ecological mapping.

Because both methods individually improve the baseline training method and are adapted at different steps, they are both implemented together. This means the final method is to incorporate and test both methods together, first integrating the loss into training and finally performing the postprocessing.

6. Experiments

This chapter presents the experiments, including the data preparation steps, parameter tuning of the models, computational resources used for training, the evaluation metrics for the results, and the experimental setup.

6.1. Data Preparation

The data preparation was conducted to make the data suitable for training. Therefore, patches were created, and the data was normalized and augmented.

6.1.1. Preprocessing

The different patch sizes were tested. The 2048 x 2048 patches did not perform as well as the 512 x 512 patches. This was because the training showed that the focus on the texture and features of each tree are more important to extract than complete tree shapes. Therefore, the patch size 512 x 512 was chosen. Furthermore, the training was first done with nine classes, but since dead trees could not be classified as their species, ten classes performed better than nine. Combining dead species as one class made it possible to correctly classify them as one class.

6.1.2. Data Normalization and Augmentation

Before training both models, the data was normalized and augmented. The library torchvision was used for both modifications. First, for the normalization of the dataset, the standard deviation and mean were calculated and used to normalize the dataset afterward. Additionally, transformations were implemented and performed on the training data to enhance the dataset's variability.

Data augmentation techniques enhance model robustness by introducing variability to the training dataset. Random horizontal flipping is performed with a probability of 0.25 to create spatial diversity. Color jittering introduces random changes to brightness and contrast, ensuring the model adapts to varying lighting conditions. A random solarization effect, with a threshold of 128 and a probability of 0.1, is applied to simulate specific lighting artifacts.

Additionally, Gaussian blur is used with a kernel size of (5, 9) and a sigma range of (0.1, 5) to introduce smoothness variations and mimic out-of-focus effects. These augmentations collectively improve model generalization by exposing it to diverse transformations.

6.2. Parameter Tuning and Baseline Model Training

After the data is set and the model and data preparation are ready, each semantic segmentation network is optimized to find the best parameters according to the model's performance. Therefore, in the training procedure, an essential part is to try out different parameters to achieve the best output for each model. This section will outline the training procedure of the two architectures chosen, giving an insight into the parameters used, like loss function, batch size, epoch, regularization techniques, and class balancing. Afterward, the best-performing parameters will be displayed, showing how the model was trained for its best performance.

Using a combination of techniques, parameter tuning was performed on both the U-Net and FCT models. Based on existing literature, a manual search was conducted to establish reasonable parameter ranges. Following this, grid search was employed to systematically explore specific ranges for each parameter. The parameters considered for tuning included batch size, learning rate, number of epochs, augmentations, optimizer choice, and various weights for the different loss functions. Performance evaluation of the different parameter combinations was based on key metrics, including F1-score, IoU, and confusion matrix analysis. This comprehensive approach ensured a thorough optimization process aimed at enhancing model performance.

6.2.1. Parameter Definition

For U-Net and FCT, the CE loss and Weighted Cross-Entropy (WCE) loss were implemented. For the weighted loss, the calculated weights for the dataset according to the number of pixels per class were used as displayed in Equation 6.2. The equation for the weight w_c for class c in Equation 6.1 is calculated as the inverse of the number of pixels N_c in that class, with normalization across all classes.

$$w_c = \frac{\frac{1}{N_c}}{\sum_{c'} \frac{1}{N_{c'}}} \tag{6.1}$$

$$weight = [0.218, 0.543, 0.715, 0.683, 3.953, 9.482, 39.028, 116.370, 3.805, 17.858] \tag{6.2}$$

First, the CE loss was used as a well-known robust loss function. It measures the difference between two probability distributions, the true labels, and the predicted probabilities from a model.

$$L_{\text{CE}} = -\sum_{c} y_c \log(p_c) \tag{6.3}$$

$$L_{\text{WCE}} = -\sum_{c} w_c \cdot y_c \log(p_c) \tag{6.4}$$

The Equation 6.3 displays the CE loss with y_c as the ground truth label for class c and p_c as the predicted probability for class c. The loss evaluates the discrepancy between the predicted and actual class probabilities. To make the loss aware of the unbalanced dataset and adapt the loss to focus more on the minority classes, a loss can be adjusted to concentrate on weights to balance the datasets. The CE loss can be adapted by multiplying the weight w_c to perform the WCE loss shown in Equation 6.4 (Jadon 2020).

Other parameters set are the number of workers that control how many parallel processes are used to load and preprocess data batches during training, helping to optimize data pipeline efficiency and reduce potential bottlenecks. Adam optimizer was used initially for the optimizer, an algorithm used during training to adjust the model's parameters to minimize the loss function. The AdamW optimizer was also tried and applied with a weight decay. The learning rate is also significant because it is crucial for the training process, as it controls how much the model updates its weights in response to the gradient during training. Also, the batch size was set to different values throughout the process because it sets the number of samples the model can process. The number of epochs per training was also regulated according to performance and was set to a relatively higher value because training will be stopped according to early stopping.

Also, since the pixel distribution of the classes is so unbalanced, as introduced in Table 4.2, there were more attempts to handle the class imbalance. Besides using weights for the loss functions and optimizer, the dataset was used to outbalance the training. Therefore, the minority classes have been retrained with only images that contain the minority species. This means all patches that contain classes six and seven were collected (Fir and Douglas Fir). The augmentations were used to avoid overfitting with this selected data, and then previously trained models were reloaded to retrain the models with the selected data.

An early stopping technique was implemented to ensure the model's best output was used in the testing procedure. To mitigate the risk of overfitting during model training, stopping the process once the validation loss has no longer been demonstrated as improved is crucial.

This regulation method avoids overfitting and ensures that the best performance of the model is retained, specifically when the highest F1-score is achieved. Therefore, the F1-score of the validation data was used as the controlling parameter. The model could be saved at optimal output by employing early stopping rather than relying on predetermined training epochs. Therefore, a patience of three epochs was set to identify when the model was stopping to improve.

The training procedure was similar for training U-Net and FCT, but two additional parameters were set for FCT. The difference to the parameters of U-Net is that for FCT also, the model itself is varied. The filters and attention heads of the models are changed throughout the training.

6.2.2. Parameter Tuning Evaluation and Training

The baseline models were trained until the parameters were optimized according to the selected parameters. This is the training of the models. For both models, the best-performing epoch, regarding the overall best F1-score, was saved to use the trained model later in testing. All parameters were tested for parameter tuning for both baseline networks. For U-Net, the parameter tuning process is shown in Table 6.1. These values were selected according to the metrics' best output and training availability within the hardware possibilities. The parameters were limited to the ability to train the model.

Also, for the FCT model, the values used for training the model are described in the following in Table 6.2. The parameters have been tried in the range of the ability to train the model. FCT is more limited than U-Net in terms of computing complexity, not allowing, for example, large batch sizes or high learning rates.

6.3. Experimental Setup and Implementation of the Combined Pixel- and ITC-Based Segmentation

The combined pixel- and ITC-based segmentation was implemented in different steps of the segmentation procedure. The integration of the ITC loss was implemented during training. Meanwhile, the semantic and instance segmentation outputs were combined in the postprocessing step. This section will explain the experimental setup and implementation of both approaches. Because both methods optimize the model's performance, the final result of the experiments is a combination of both approaches to achieve the best results.

Table 6.1.: Table of parameters tested and the result of the best values achieved for U-Net (CE = Cross Entropy, WCE = Weighted Cross Entropy).

Parameter	Tested Values	Best Value
Number of Workers	2, 4	2
Optimizer	Adam, AdamW	AdamW
Weights for Optimizer AdamW	0.01, 0.1, 0.009, 0.011	0.011
Learning Rate	0.001, 0.0001, 0.00001	0.0001
Batch Size	8, 16, 32, 64	8
Number of Epochs	15, 25, 30, 40, 50, 80	30
Number of Classes	9, 10	10
Loss Function	CE Loss, WCE Loss	CE Loss
Weights in Loss	Yes, No	No
Retraining	Yes, No	No
Retraining Minority Classes	Yes, No	No
Augmentation	Rotation, Scaling, Flip, None	None

6.3.1. Parameter Tuning for the Integration of the ITC Loss Function

The second loss function, the ITC loss, was implemented into the training process. Because of the goal to create the best output, only the overall best-performing U-Net model was trained with the second loss. Therefore, all settings and parameters from the best model were used to train again using the ITC loss.

For this training, evaluating the factors α and β was important. The aim was to find their best balance, resulting in the best output. This was measured by the improvement of the metrics and the decline in loss. The best factors enhancing the combination of the losses is shown in Table 6.3. It shows the best parameter setting of the Equation 5.2. The best output for α=1 and β=1.5 have been achieved. The factors tested are shown in Table 6.3. The factors were both changed one at a time, setting the other factor to one. The result indicates that factorizing the ITC loss improves the results. The improvement was examined by testing it with Test-2 of the test dataset. This was done by testing the model's performance without the ITC loss in contrast to implementing it. The metrics IoU and F1-score were evaluated per class.

Table 6.2.: Table of parameters tested and the result of the best values achieved for FCT. The numbers for filter and attention head are explained in chapter 5 (CE = Cross Entropy, WCE = Weighted Cross Entropy).

Parameter	Tested Values	Best Value
Attention Heads	1, 2, 3	3
Filters	1, 2, 3	3
Number of Workers	2, 4	2
Optimizer	Adam, AdamW	AdamW
Weights for Optimizer AdamW	0.01, 0.1, 0.009, 0.011	0.011
Learning Rate	0.001, 0.0001, 0.00001	0.00001
Batch Size	4, 6, 8, 16, 32,	6
Number of Epochs	15, 25, 30, 40, 50, 80	40
Number of Classes	9, 10	10
Loss Function	CE Loss, WCE Loss	CE Loss
Weights in Loss	Yes, No	No
Retraining	Yes, No	No
Retraining Minority Classes	Yes, No	No
Augmentation	Rotation, Scaling, Flip, None	None

6.3.2. Image Selection and Patch Resizing for Combining Semantic Segmentation and Instance Segmentation

For postprocessing, 15 images of the source input images, bigger than 6144 x 6144 pixels, were selected to create at least three by three 2048 x 2048 patches. For the instance segmentation, generated outputs using 512 x 512 and 2048 x 2048 pixel patches were given, ensuring comprehensive classification of trees by applying a 50 percent overlap between patches. This approach was crucial to guarantee that all tree instances were fully captured. Since the 2048 x 2048 patches could generate much better results for the instance segmentation, they were further processed. From these results, the individual tree crown shapes were extracted as JSON files to be usable for voting. In contrast, the semantic segmentation process for the testing utilized 512 x 512 pixel patches without overlap, as this technique is primarily concerned with pixel classification rather than shape integrity. The image patches were extracted from the original test images, and for the 2048 x 2048 instance segmentation outputs, the 512 x 512 patches were concatenated

Table 6.3.: Factors α and β tested for the loss function, yielding the best-combined performance for CE and ITC loss.

Parameter	**Tested Values**	**Best Value**
α	1, 1.5, 2	1
β	1, 1.5, 2	1.5

to 2048 x 2048 patches to facilitate a direct comparison. To ensure compatibility in the analysis, the 512 x 512 patches were fused into 2048 x 2048 patches, with particular attention given to maintaining the 50 percent overlap of the instance segmentation results during this resizing process. This patch creation enabled a thorough comparison of both outputs. Ultimately, the combining and voting strategy was implemented with both prepared data. The majority voting was performed with the 2048 x 2048 patches and the individual tree crown shape. As a final step, the image patches of the same original image were put together into three by three images for the final output. Afterward, the metrics could be calculated for the resulting 15 images. For comparison, the same resulting images were produced without postprocessing, using the direct output after training the best-performing model. The performance was tested for both scenarios with and without postprocessing for the 15 selected images, calculating the IoU and F1-score for each class.

6.4. Computational Resources

For this work, the computational resources in the scope of the availabilities within the German Aerospace Center (DLR) have been used. Therefore, the portal Terrabyte was used. It is a cooperation between the DLR and the Leibniz Supercomputing Center (LRZ) to provide new information technology for researchers. Terrabyte includes infrastructure, services, and data management. The data storage and the high performance computing (HPC) SLURM for graphics processing unit (GPU) usage have been used for this work.

6.4.1. Hardware

For using SLURM to train the models, Terrabyte combines classic processors central processing unit (CPU) with accelerators and graphics processing units GPUs. Terrabyte provides 44.000 virtual CPU cores and 188 NVIDIA GPUs computing resources.

The computational resources used for this work included nodes with 100 GB of memory each and 8 CPUs per task, specifically Intel Xeon Gold 6336Y processors with 24 cores, operating at

2.4 GHz. A single GPU node with an NVIDIA HGX A100 featuring 80 GB of memory and a power rating of 500W was utilized for GPU-based processing. These resources ensured efficient handling of the training and processing demands.

Data is mainly stored in the Data Science Storage component of Terrabyte, which provides around 45 petabytes of usable storage. Also, data storage with one terabyte capacity was used for this project to easily store and use the dataset "BAMFOREST" very efficiently (*terrabyte Documentation* 2024).

6.4.2. Software

On Linux as the operating system, Visual Studio Code was utilized as the primary code editor in the DL development environment. It was selected for its flexibility and comprehensive set of extensions designed to streamline artificial intelligence workflows. PyTorch is the core framework for model development and training, offering dynamic computation graph capabilities that facilitate efficient and intuitive network construction. Torchvision aids in image processing and transformations, providing specialized tools for computer vision applications. Torchmetrics is used to compute various performance metrics to standardize the evaluation process. For image handling, Pillow is utilized for general image processing, while Rasterio is employed to handle geospatial raster data, offering robust support for reading and writing such data formats. Additionally, NumPy is best in performing efficient numerical operations and array manipulation, which is essential for data preprocessing and transformation tasks. For experiment tracking and metric visualization, Weights & Biases (wandb) is employed, enabling comprehensive logging and performance analytics throughout the research.

6.5. Evaluation Metrics

A selected combination of evaluation metrics was chosen to assess the model's effectiveness in accurately identifying and distinguishing between different TS: IoU, F1-score, and the confusion matrix. Each metric provides unique insights into the model's strengths and weaknesses, especially in handling class imbalances and accurately identifying individual classes in a multi-class classification setting.

IoU is a widely used metric in image segmentation and classification tasks. It measures the overlap between the predicted area and the ground truth, helping to quantify how accurately the model identifies each TS. It is calculated with the true positives (TP), false positives (FP) and false negatives (FN), displayed below in Equation 6.5. A higher IoU score indicates a better match between the predicted and true locations of a specific species, making it a robust metric for spatial accuracy in species identification tasks.

$$\text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}} \tag{6.5}$$

The F1-score combines Precision and Recall (Equation 6.7) into a single metric, taking their mean to balance the two. Precision measures the proportion of correctly identified instances for each species out of all the cases predicted as that species, indicating how many of the model's predictions are accurate. On the other hand, Recall measures the proportion of correctly identified instances out of all valid instances for each species, showing how well the model captures every instance of that species. By combining these two aspects, the F1-score (Equation 6.6) provides a balanced measure that is particularly valuable in TS identification, where certain species may be more common than others. This makes the F1-score robust against dataset imbalances, ensuring accuracy (Precision) and completeness (Recall) are considered. As a result, it offers a reliable assessment of the model's overall consistency and accuracy in distinguishing between species.

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{6.6}$$

$$\text{where} \quad \text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{6.7}$$

The structure of the confusion matrix, displayed in Table 6.4, offers a comprehensive view of the model's classification results, presenting a breakdown of TP, true negatives (TN), FP, and FN for each species class. It highlights any systematic misclassifications or confusions between specific species and allows to understand which species pairs are often mistaken for each other. This insight is crucial when evaluating the model's reliability, as it helps identify where additional training data or model fine-tuning might be necessary. The Confusion Matrix also allows us to visualize class-wise performance, making it easier to interpret and improve the model based on specific errors. In this context, identifying which species are misclassified as others is highly beneficial in understanding which TS are hard to distinguish for the model. Given the imbalance in the dataset's number of pixels per species, the output will be presented as percentages per species rather than total numbers in the output. This approach facilitates more precise visualization of the distribution of predictions for each class while also enabling a comparison of TP across all species.

Table 6.4.: Example of the distribution of predictions in a Confusion Matrix.

	Predicted Positive	Predicted Negative
Actual Positive	TP	FN
Actual Negative	FP	TN

In combination, these metrics allow for a nuanced understanding of the model's performance. The IoU assesses spatial accuracy, F1-score balances Precision and Recall for imbalanced datasets, and the Confusion Matrix exposes specific areas of confusion between species. They comprehensively evaluate the model's capability to accurately and consistently differentiate between TS, supporting more reliable ecological analysis and decision-making (Zhong et al. 2024).

For the testing of the models, the output metrics used are consistent, including the F1-score in Equation 6.6 and IoU in Equation 6.5. These were chosen since most comparable papers have chosen either to present their results. To ensure a comprehensive evaluation, the values for both metrics are also calculated as the overall output parameter of all classes' performance as the average F1-score and the mean intersection over union (mIoU) in Equation 6.8 where N is the total number of predictions. This averaging approach ensures a consistent and reliable measure of the model's overall performance.

$$\text{Average F1-score} = \frac{\sum_{i=1}^{N} F1_i}{N}, \quad \text{mIoU} = \frac{\sum_{i=1}^{N} IoU_i}{N} \tag{6.8}$$

7. Results

The results are presented, highlighting key results for evaluating model performance and effectiveness of the proposed methods. They include analyses of overall metrics as well as assessments of performance for each class.

7.1. Results of Pixel-Based Semantic Segmentation

The first results are the training outputs of the baseline segmentation models U-Net and FCT in Table 7.1. All trained models were evaluated for the two test scenarios, Test-1 and Test-2. The comparison with the F1-score and the IoU for the two models compares a familiar (Test-2) and a distinct forest area (Test-1).

The results demonstrate that U-Net outperformed FCT in both test regions for the F1-score and IoU, see Table 7.1. Specifically, in Test-1, U-Net achieved an F1-score improvement of 11.65 percent and an mIoU increase of 8.56 percent compared to FCT. For Test-2, U-Net also performed better, with a 5.03 percent higher F1-score and a 4.85 percent increase in mIoU. These results highlight a clear performance advantage and superior outcomes with the U-Net architecture.

Even though U-Net outperforms FCT in the overall output, the per-class performance is still crucial to examine. This is because the performance per class is very different due to the number of pixels per class and the various textures of the different species. For Test-1, the class-wise results for all classes show better results for U-Net, except for classes one (Pine), four (Spruce), and nine (Dead Trees), which are better for FCT, as demonstrated in Table 7.1. For Test-2, the class-wise results for all classes show better results for U-Net, except for class seven (Douglas Fir), which is better for FCT.

In addition to differences in overall performance, substantial variations are observed in the output values for the same class between the models. In Test-1, the result of U-Net is much higher compared to FCT for classes five (Larch) and eight (Other Trees). FCT, on the other side, has remarkably higher results for classes one (Pine), four (Spruce), and nine (Dead Trees). Test-2 results are more balanced, but U-Net performs much better for class six (Douglas Fir),

Table 7.1.: F1-score and IoU comparison of U-Net and FCT per class for Test-1 and Test-2. For comparison, the proportion of the classes of the dataset is provided.

		Test-1				Test-2			
	Proportion	F1		IoU		F1		IoU	
Class	of Pixel (%)	U-Net	FCT	U-Net	FCT	U-Net	FCT	U-Net	FCT
0 - Background	45.61	**0.7621**	0.7321	**0.6157**	0.5774	**0.7812**	0.7751	**0.6409**	0.6328
1 - Pine (Pinus)	14.46	0.0074	**0.0111**	0.0037	**0.0056**	**0.8136**	0.7642	**0.6857**	0.6161
2 - Beech (Fagus)	15.08	**0.2995**	0.2880	**0.1761**	0.1682	**0.6469**	0.5929	**0.4781**	0.4213
3 - Oak (Quercus)	13.88	**0.5011**	0.4233	**0.3343**	0.2685	**0.6140**	0.5800	**0.4430**	0.4084
4 - Spruce (Picea)	2.48	0.0054	**0.0092**	0.0027	**0.0046**	**0.7355**	0.4504	**0.5816**	0.2907
5 - Larch (Larix)	1.03	**0.1547**	0.0317	**0.0838**	0.0161	**0.7829**	0.3338	**0.6432**	0.2004
6 - Douglas Fir (Pseudotsuga)	0.44	0.0000	0.0000	0.0000	0.0000	**0.1401**	0.0000	**0.0753**	0.0000
7 - Fir (Abies)	0.19	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
8 - Other Trees	6.27	**0.4265**	0.0188	**0.2711**	0.0095	**0.3355**	0.1166	**0.2016**	0.0619
9 - Dead Trees	0.55	0.0766	**0.1667**	0.0398	**0.0909**	**0.5825**	0.5133	**0.4110**	0.3452
Average	-	**0.5630**	0.4464	**0.4182**	0.3326	**0.7008**	0.6505	**0.5545**	0.5060

with a remarkably higher IoU.

Generally, Test-1 is much more challenging because of the different species distribution compared to the training data. The results show that U-Net performs better for classes with a higher share of data, and for classes with a low share, FCT performs better. Since Test-1 had only a high number of trees for classes two (Beech), three (Oak), and eight (Other Trees), U-Net was superior. Also, class five was better detected for U-Net even though the class share was only 1.06 percent of the trees. For FCT, for classes with a lower share, it performed better. But the classes got detected at a very low rate. These results are not very meaningful. Only the dead trees could be detected meaningfully better than by U-Net. Also, U-Net is weaker for deciduous TS for Test-1, namely classes two (Beech) and three (Oak). This could outline a problem that the model is facing in having trouble classifying more even textured trees. Additional unknown factors could also contribute to this problem. Overall, the results for Test-1 are not satisfactory, leading to the inability to generalize the model on this different model. This could be because over 50 percent of the trees were delineated as "Other trees", a class that the segmentation architecture can not meaningfully learn since it contains many species.

Test-2 performed well for seven classes, for U-Net and FCT, but was not satisfactory for three classes. The problem with classes six (Douglas Fir), seven (Fir), and eight (Other trees) is that they are represented with too little data to successfully train, and class "Other Trees" is trained with an unknown combination of species, which also makes it unable to predict the class. Overall, the results from Test-2 are satisfactory, showing that the model can train the classes with sufficient data and one species per class.

Additionally to the per-class performance of the model using the metrics IoU and F1-score, the confusion matrix of each test is also computed for the overall best result of the U-Net model, showing potential misclassifications. Figure 7.1 displays the confusion matrix in percent for each class to show how much each class is classified correctly. The rows represent the true label, and the columns represent the predicted label. The diagonal shows the TP, which are classified right.

The results show that for Test-1 (Figure 7.1a), many classes are classified as background. The class that gets classified as background the most is class six (Douglas Fir). Classes zero (Background), two (Beech), three (Oak), eight (Other Trees), and nine (Dead Trees) are classified right to some extent. Looking at the columns of the matrix, there is a trend in which classes get misclassified as class zero (Background). Further, the three classes, two (Beech), three (Oak), and eight (Other Trees), are classified as different classes. Only class zero (Background) and three (Oak) get a high TP, indicating they are similar to the training data. Test-2 Figure 7.1b shows a more stable output, classifying all classes mostly right except for classes six (Douglas Fir), seven (Fir), and eight (Other Trees). For classes six (Douglas Fir) and seven (Fir), there is a lot of misclassification with many classes. The matrix shows that trees get misclassified as background for all classes, which is the biggest problem of wrong classification.

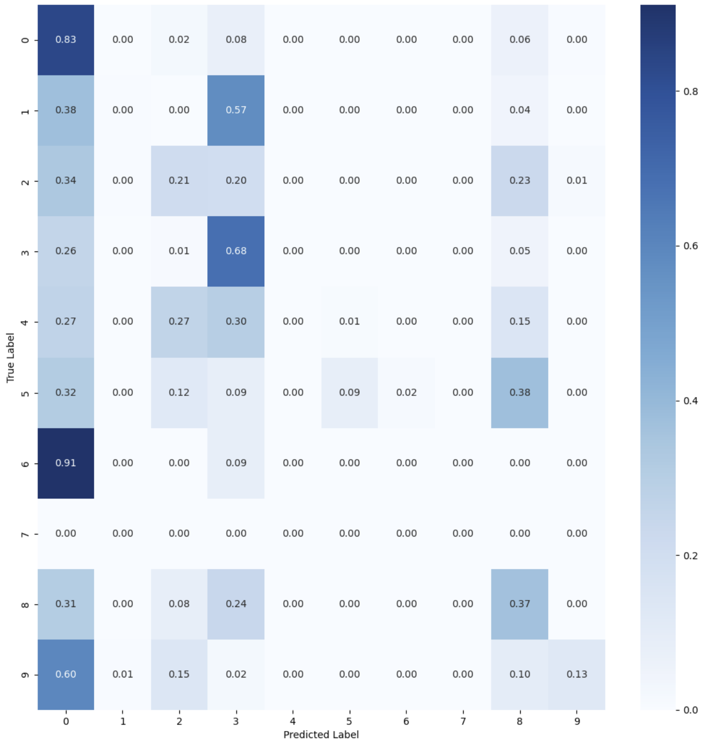
Also, because of the missing delineation, some errors are included, where classes are incorrectly classified as the class Background in the matrix, but this is because of the missing labels. There is a trend of confusion between classes two (Beech) and three (Oak). Class eight (Other Trees) is misclassified as the deciduous TS (classes two and three).

7.2. Combined Pixel- and ITC-Based Classification

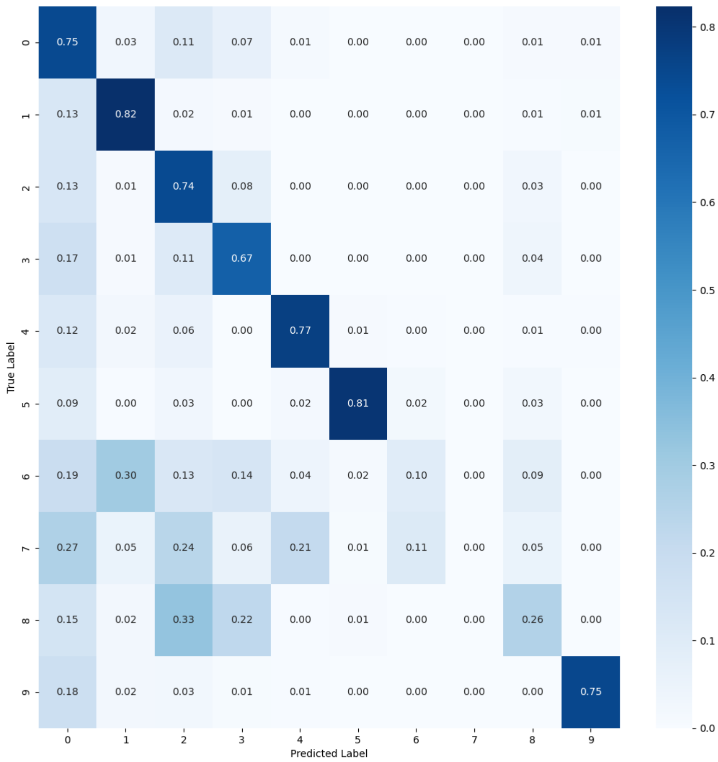
The results of the improvement methods are presented below. Therefore, the implementation of the ITC loss function will be shown first, followed by the results of the postprocessing approach. Because of the overall better performance of the U-Net architecture, this model was chosen to further be used for the ITC loss implementation and the postprocessing.

7.2.1. Results of Implementation of ITC Loss in U-Net

In Table 7.3, the F1-score and the mIoU before and after implementing the second loss function are compared on the complete test dataset. The result shows that with the implementation



(a) Confusion matrix for Test-1.



(b) Confusion matrix for Test-2.

Figure 7.1.: Confusion matrices in percent for U-Net showing the true and predicted classes for Test-1 (a) and Test-2 (b). (0 = Background, 1 = Pine, 2 = Beech, 3 = Oak, 4 = Spruce, 5 = Larch, 6 = Douglas Fir, 7 = Fir, 8 = Others, 9 = Dead Trees).

Table 7.2.: F1-score and IoU comparing CE loss with the combined loss (CE and ITC) for U-Net from predictions of Test-2.

	F1-Score		IoU	
Class	CE	CE + ITC	CE	CE + ITC
Background	0.7812	**0.7992**	0.6409	**0.6656**
Pine (Pinus)	0.8136	**0.8234**	0.6857	**0.6999**
Beech (Fagus)	0.6469	**0.6555**	0.4781	**0.4875**
Oak (Quercus)	0.6140	**0.6190**	0.4430	**0.4482**
Spruce (Picea)	**0.7355**	0.7340	**0.5816**	0.5798
Larch (Larix)	0.7829	**0.8268**	0.6432	**0.7047**
Douglas Fir (Pseudotsuga)	**0.1401**	0.0176	**0.0753**	0.0089
Fir (Abies)	0.0000	0.0000	0.0000	0.0000
Other Trees	0.3355	**0.3363**	0.2016	**0.2021**
Dead Trees	0.5825	**0.5970**	0.4110	**0.4255**
Average	0.7008	**0.7123**	0.5545	**0.5703**

of the ITC loss function, the F1-score improves by 1.15 percent, and the IoU improves by 1.58 percent. All classes improve except for classes four (Spruce) and six (Douglas Fir). Class five (Larch) showed a relatively significant improvement of 4.39 percent. Other improving classes show a change of about one percent or less. The results show that the ITC loss benefits classes with enough training data. The issue is that the classification gets worse in classes that are not well-trained because they only have little data. This indicates that the newly implemented loss function is most beneficial when used in a balanced or weighted dataset and can only benefit all classes if they are meaningfully trained.

7.2.2. Results of Integration of ITC-Based Instance Segmentation Results in the Postprocessing

After the best semantic segmentation model had been found using U-Net with the implementation of the ITC loss, the model was further refined in the postprocessing by combining the semantic segmentation result with an instance segmentation. The output in Table 7.3 shows the improvement of applying postprocessing on 15 testing images. It can be seen that both IoU and F1-score improve in postprocessing. All classes improve with postprocessing except for class seven (Fir), which is undetected. Also, the F1-score of class eight (Other Trees) is not increasing but is nearly the same. The improvement of each class differs, from about one percent to ten percent. The results show that postprocessing after training can further enhance the results. Using the postprocessing steps shows that the results can be significantly improved. A positive effect is also that the approach disadvantages no class. Postprocessing enhances the classified ITC further.

Table 7.3.: F1-score and IoU comparing the best training model before and after the postprocessing voting for U-Net from predictions of 15 selected testing images of Test-1 and Test-2.

	F1-Score		IoU	
Class	Before	After	Before	After
Background	0.7675	**0.7963**	0.6273	**0.6647**
Pine (Pinus)	0.5200	**0.5304**	0.4400	**0.4544**
Beech (Fagus)	0.4804	**0.5412**	0.3328	**0.3861**
Oak (Quercus)	0.4839	**0.5812**	0.3365	**0.4289**
Spruce (Picea)	0.3314	**0.3733**	0.2562	**0.3094**
Larch (Larix)	0.1244	**0.1345**	0.0962	**0.1105**
Douglas Fir (Pseudotsuga)	0.0059	**0.0096**	0.0031	**0.0051**
Fir (Abies)	0.0000	0.0000	0.0000	0.0000
Other Trees	**0.2331**	0.2313	0.1395	**0.1413**
Dead Trees	0.2638	**0.3473**	0.1865	**0.2695**
Average	0.4744	**0.5069**	0.3541	**0.3889**

7.3. Results Visualization

To understand the results further, they are visualized. This section displays the models' performance, the advantages of postprocessing, and the ITC loss function by showing output images. All classes are displayed to show good and not ideal output examples. The images are a selection of samples that represent all classes.

Figure 7.2 displays the first selected outputs for the training and the postprocessing. It shows the input image for training in (a), the ground truth showing the suitable class and shape in (b), the best model output from training with the ITC loss function in (c), and the output after the postprocessing in (d). The results show the right classification for most of the patches. Most classes are classified as their true species, but Fir (light blue) and the "Other Trees" (orange) are misclassified. This is because of the poor training data for the Fir class and the confusion of combining many TS into one class.

Furthermore, the outputs in (c) cannot generate a detailed delineation like the labels. Therefore, postprocessing (d) shows the advantage of implementing another method after training. The output provides greater clarity, separating merged ITCs into single ones. The segmentation challenges appear in areas where overlapping individuals of the same species make it difficult to distinguish single trees.

In row five, the postprocessing shows another advantage. The Douglas Fir (purple) shows the impact of this final step. The shape in (c) is corrected in (d), and the hard-to-classify species is computed as one TS, eliminating the multiple species classifications.

The relation between the Fir species can also be seen in row six, where a labeled Fir (light blue) gets misclassified as a Douglas Fir (purple). This stresses the problem of the similarity of closely related TS.

The results also show that some trees are not delineated but recognized by the model. If they appear at the edges of the images, they are too small or not labeled before for other unknown reasons. This means that some TS are trained as "Background", which is misleading for the training model. As shown in Figure 7.2 in the fifth and last row, it can be seen that a tree was not classified but then detected by the model, which will appear as a false classification in the output metrics. Also, in the last row, it can be seen that a dead tree was given a different vitality, and it appears as a species instead of a dead tree.

More output images are displayed in Figure 7.3. To compare the result in the sense of comparing many trees in an image, the output is additionally displayed in the original image size, with the final result after postprocessing and implementing the ITC loss function. In Figure 7.3, three images are shown with the ground truth (a), the prediction after the postprocessing (b), and the input (c). This gives a representative overview of how the results can classify forests accurately by showing many whole tree instances. This figure depicts the overall output showing the best result when combining both methods. Since one image of each area is displayed, it can be seen that for the area of Test-2, the first two images, the output is better than for the third image, which is an image of Test-1. The ground truth shows that many of the ITCs are classified as "Other Trees," which are not classified as such in the result. That shows how different the Test-1 set is, being more prone to misclassification.

In Figure 7.4, the results of both proposed methods are displayed on one of the input images. This figure is shown to compare the impact of each approach on the final results. This

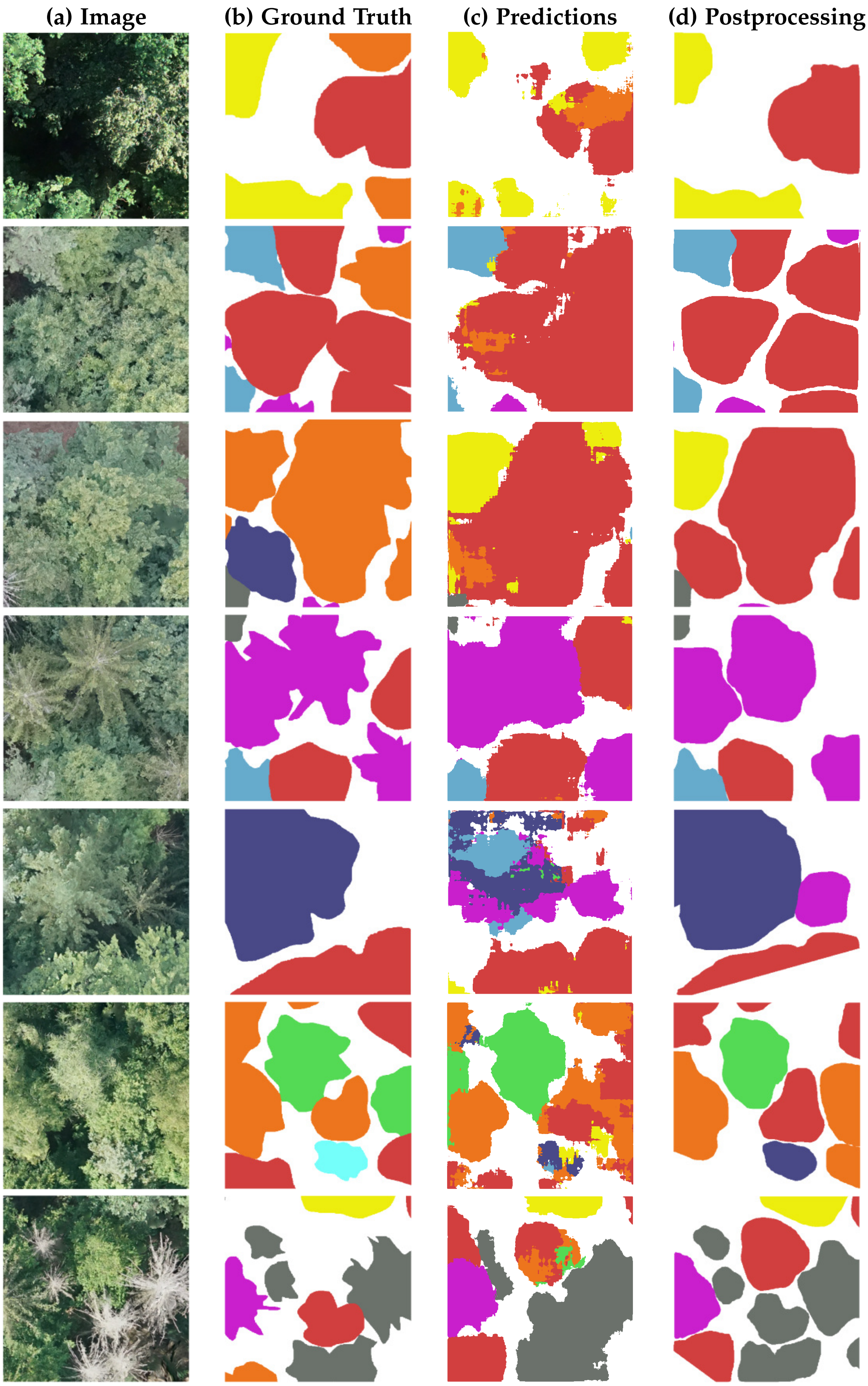


Figure 7.2.: Comparison of image (a), ground truth (b), predictions with ITC loss (c), and postprocessing (d). The first row represents image patches of the Hain region from Test-1, and the other six rows represent those from Stadtwald and Tretzendorf from Test-2.

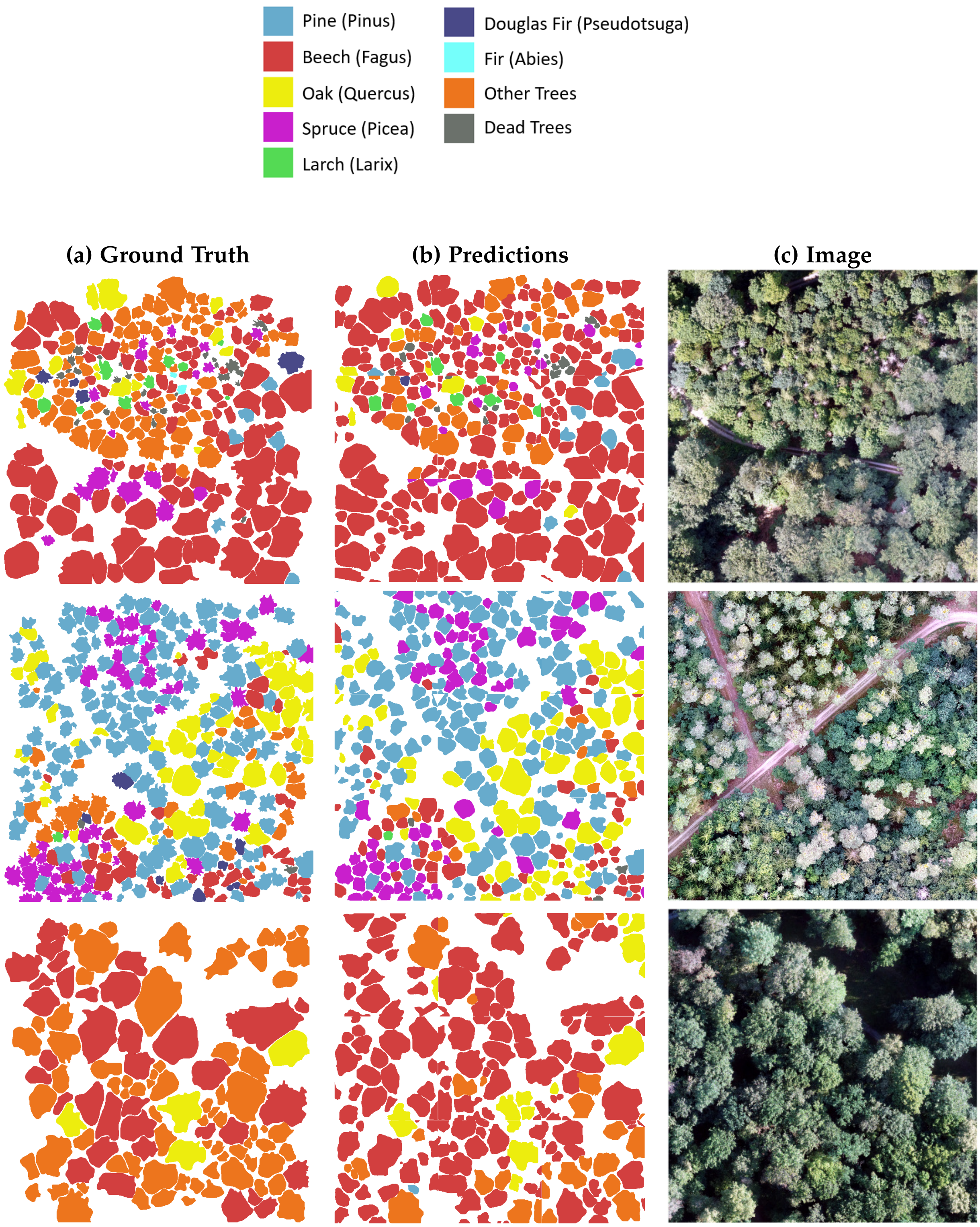


Figure 7.3.: Comparison of ground truth (a), the final output of the postprocessing image (b), and the input image (c) of original input images. Each image is of one area. From top to bottom, first, Tretzendorf, then Stadtwald, and at the bottom, Hain.

visualization helps to identify the strengths and weaknesses of the introduced methods. The results show the first comparison of (c) and (d), the implementation of the ITC loss. The results show that in (c), the ITC loss improves the results. To compare these images better, images (a) and (b) provide a zoom-in of the images (c) and (d). The difference can be seen in that the shape of the image (b) output is more precise, the noise around the ITCs is reduced, and for most species, the decision per shape is more reduced towards one class. The final output of the postprocessing step (e) shows the output of the final method. This step refines and separates the ITC shapes and eliminates any noise or multi-classification of one tree instance. Summarized, the improvement from both methods can be seen from (c) to (d) with better and more accurate classifications and from (d) to (e) with shape correction, single species classification per ITC, and noise removal.

Compared to the ground truth (f), the output only has the limitations of some misclassifications, but overall, the best result. It can also be seen in images (e) and (f) how many trees are not delineated in the ground truth that are classified in the predictions. This is a significant indicator that the metrics of the results are not valid to some extent.

7.4. Computational Efficiency of Experiments

The training time of the model was recorded while ensuring consistent conditions, including the same number of epochs and identical input images for a fair comparison. Although time-dependent training was not prioritized, maintaining computational efficiency remained crucial. Focusing on efficient resource utilization aimed to minimize runtime overhead while delivering optimal performance. This approach allowed for a balanced assessment of model efficiency without sacrificing the consistency of the experimental setup. Table 7.4 shows the significant time difference in model training. The U-Net model is much faster than FCT by double the time. With the high computational costs of the second loss function, adding it to the U-Net model increases the runtime enormously by six times.

Table 7.4.: Comparison of runtime for U-Net, FCT and inclusion of the ITC loss function for completing the same task with similar epochs and parameters.

Training model	Runtime
FCT Model	6h 56m
U-Net model	3h 30m
U-Net model with ITC loss	19h 2m

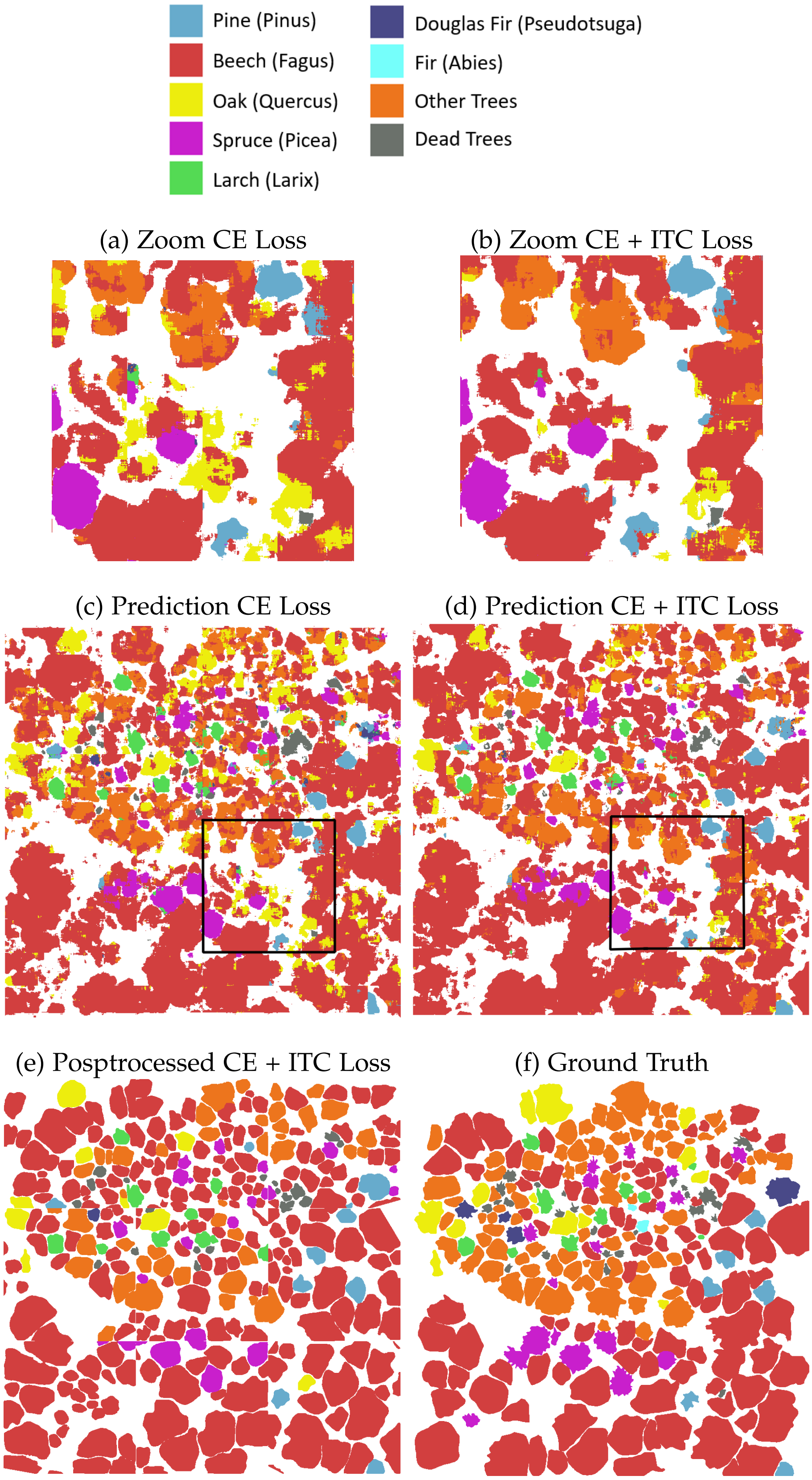


Figure 7.4.: Comparison of the ITC loss and postprocessing output using only the CE loss (c) and using CE and ITC loss (d), with the zoom in (a) and (b). Image (e) shows the final postprocessing and (f) the ground truth.

7.5. Comparison with Previous Work

Since the benchmark dataset "BAMFOREST" is new, there are no other results regarding the semantic segmentation to compare this dataset. Two other papers tried a similar approach to semantic segment TS with different datasets and approaches. The metrics of the results are not on the same basis but will be compared to have a relative comparison.

In Table 7.5, the results are compared. Schiefer et al. (2020) and Ramesh et al. (2024) will be compared to this research output. Unlike this research, they both have smaller labeled data for training, but both classify more species. Also, their multimodal approaches include using DSM or time-series images. What makes the comparison of these results interesting is that the spatial resolution is the same. Furthermore, the dataset used by Schiefer et al. (2020) includes a German forest with similar TS, and all used the same architecture U-Net, with some modifications.

Table 7.5.: Comparison of mIoU and F1-score for this research output, compared to related literature. The values are all from different datasets with different approaches. For this comparison, the results are without postprocessing. This is because of limited testing data usage in postprocessing. Only the training output is compared.

	Schiefer et al. (2020)	Ramesh et al. (2024)	This work
Approach	RGB + DSM	Time-Series	CE + ITC Loss
Architecture	U-Net	U-Net (R-101)	U-Net
F1	0.73	-	0.71
mIoU	-	0.55	0.57
No. classes	14	14	10
Spatial resolution (cm)	2	2	2
Dataset (ha)	51	44	105
Differences	German Forest	Canadian Forest	German Forest Only Test-2

The results show that this research achieves similar metrics for the overall F1-score and the mIoU. This is achieved even though both use a multimodal approach. However, this result is only an indicator of the different factors that change the work. For the output of this work, only Test-2 was used, which is from the same area as the training data. Also, fewer TS were classified, and the dataset is much larger. What is remarkable is that both Ramesh et al. (2024) and this work tried different architectures, also newer ones compared to U-Net, but both achieved the best results with U-Net.

When comparing the output imagery of all work, this research outstands by additionally using the ITC-based instance segmentation of the postprocessing, which enables easy differentiating between individual tree crowns. This eases the comparison of the individual trees.

8. Discussion

The discussion is divided into three sections covering the most critical aspects of this research, starting with impacts related to factors of the environment, like forest traits or data availability. Second, the aspects that were evaluated during the experiments are discussed. Finally, this research examines the behavior of the networks incorporating semantic segmentation and both proposed ITC-based methods.

Due to their distinct structural characteristics, the training proved to be more straightforward for coniferous than deciduous trees. Deciduous trees show a disadvantage because of their even and more homogeneous traits. However, distinguishing closely related TS that appear similar remains a significant challenge, like the Douglas Fir and Fir. This is enriched if the quantity of training data is limited because the differences in the structural characteristics are not trained sufficiently.

While the model's output often mirrors the labeled shapes correctly, inconsistencies arise because not every tree is delineated correctly in the ground truth. The dataset shows some wrong and missing delineations.

Additionally, the classification of "Other Trees" presented substantial difficulties. This class includes trees that experts could not confidently assign to known target species. It includes various textures and traits. Consequently, the model struggles with this class, as its variability leads to confusion with other classes.

Furthermore, the dataset was limited to park and natural forest areas with only natural backgrounds, dirt roads, or meadows, excluding images with infrastructure or background objects like roads, cars, or buildings, which limits its applicability to more diverse environments.

During the experiments, a smaller patch size of 512 x 512 proved to be beneficial, aligning with the findings of Schiefer et al. (2020), as very high-resolution data of two cm GSD is particularly effective when the traits of the trees can be learned in detail, as described in chapter 6. Since the GSD is the highest for to this point known tree datasets, this enables networks to focus on the tree traits rather than just the tree's shape or color.

To tackle the unique needs of the species classification, the baseline networks were adjusted and optimized during the parameter tuning. A weight for the CE loss and data augmentations were tested to balance out the training of each class. Still, they were not used due to poor results, suggesting that these strategies require further refinement for this dataset. For Schiefer et al. (2020), the data augmentation improved the results, whereas applying weights to their loss function showed similar outcomes to this work.

The experiments showed that U-Net demonstrates better results during parameter tuning than FCT, suggesting that it is more suited for the task of TS classification involving the chosen parameters. This is validated by Ramesh et al. (2024), where U-Net was also superior to other baseline CNN and transformer architectures. These findings highlight the importance of model-specific optimization and underscore U-Net's stability in handling complex segmentation tasks.

The U-Net model performed well testing on the test data from the same region as the training data, Test-2, successfully classifying seven classes, including five TS, indicating its capability to generalize within similar datasets. Only Ramesh et al. (2024) and Schiefer et al. (2020) could classify more species in multi-species classification, using more advanced multimodal approaches.

Minority classes, particularly classes six (Douglas Fir) and seven (Fir), were poorly trained due to insufficient representation in the training data. Also, other studies show that the performance divergence of their results compared between the different classes vary a lot (Schiefer et al. 2020; Ramesh et al. 2024). Attempts to address this through weighted loss functions or additional retraining were unsuccessful, highlighting the difficulty of improving performance for underrepresented classes.

The poor performance on Test-1, a dataset with a different class distribution and a different region than the training data, revealed limitations in the model's adaptability to unseen conditions far from the training data.

Integrating the proposed ITC-aware loss function for segmentation significantly improved classification outcomes. However, its effectiveness was limited to classes with sufficient training data and adequate initial performance. Classes with low IoU or F1-scores are further disadvantaged by the ITC loss, as the approach requires minimum class training results to enhance them. The experiments showed that an F1-score above 58 percent is successfully improved. The ITC loss is a new approach that benefits by incorporating tree crown information. It helps the model not only perform pixel-based classification but also perform tree instance-based learning. The ITC loss benefits include incorporating a defined shape and learning that each tree shape is limited to one class.

The postprocessing step, which also incorporated ITCs, introduced another substantial improvement, correcting shapes, separating tree crowns within large homogeneous areas, and reducing noise from non-tree regions. These enhancements benefit all trained classes, demonstrating the value of postprocessing in refining segmentation results. By separating ITCs and reducing each tree shape to one predicted class, this work presents a new approach to using postprocessing for correcting the pixel-based classification with a voting mechanism. However, the approach relies heavily on usable images and prior tree crown segmentation results, which restricts its application to specific data, suiting the results of the instance segmentation.

The novel approach of applying the ITC loss on semantic segmentation in combination with the introduced postprocessing strategy showed a significant improvement. Tackling the problem of dense and overlapping tree crowns in semantic segmentation showed a remarkable effect with being able to separate ITCs, compared to other related work.

9. Conclusion

This study introduced an unexplored approach for TS classification, combining semantic and ITC segmentation to accurately identify ITCs along with the species in a natural German forest. The model demonstrates a strong performance by leveraging an ITC-based loss function and an innovative postprocessing step, especially for coniferous TS.

This research explored the following research questions in TS classification using DL techniques and an end-to-end ITC-based loss function followed by a postprocessing method implementing an ITC segmentation.

How successfully can deep learning be used for tree species classification?

DL has proven to be an effective tool for TS classification in this work when applied to a very high-resolution dataset. Models like U-Net and FCT demonstrate their capability to classify TS by leveraging pixel-based semantic segmentation. U-Net performed most robustly with an F1-score of 71.23 percent and a mIoU of 57.03 percent, handling dense forest areas and differentiating between species with distinct traits and textures. However, challenges remain in accurately classifying closely related species and handling ambiguous or poorly defined classes like "Other Trees" or with little data availability. The success of DL in this context relies heavily on the quality of training data.

What percentage of training data per class is required to achieve robust classification of tree species, measured by IoU and F1-score?

The amount of training data per class significantly impacts the model's performance. This work revealed that classes with plenty and diverse training samples achieve much higher performance, as seen with well-represented species in the dataset. For this work, classes with more than two percent of the overall pixels in the dataset can be successfully classified. Contrarily, minority classes with only 0.35 to 0.81 percent of the pixel-based data showed poor performance due to insufficient representation, even after attempting to address the imbalance through a weighted loss function and retraining. The results suggest that robust classification demands a sufficient number of samples per class and diversity in the data to capture variations in species traits. Additionally, regarding the amount of training data, the TS also need to be considered since some trees share easier learnable traits than others.

To what extent can ITC-based training and postprocessing enhance pixel-based tree species classification?

Incorporating ITC segmentation results into the semantic segmentation significantly improved the pixel-based classification results. Both approaches, the ITC loss function and the postprocessing, improved the F1-score significantly by 1.15 and 3.25 percent. The ITC loss function enhanced the model's training for classes with enough training data. However, the approach was less effective for underrepresented classes with low initial performance, as it relies on sufficiently accurate results for a successful refinement. Based on ITC segmentation, the postprocessing step corrected shapes and refined classifications, resulting in better alignment with ground truth labels across most classes. Since postprocessing is performed after training, it relies heavily on prior training results. If these results are sufficient, outstanding improvements can be achieved.

This study outlines several limitations in using DL for TS classification. The class imbalance issue, particularly for minority classes like Douglas Fir and Fir, limits the model performance and generalizability. Efforts to mitigate this, such as using weighted loss functions and targeted retraining, did not yield significant improvements. Deciduous trees proved to be particularly difficult to classify due to their homogeneous texture and shape. Moreover, the heterogeneous mixture of species of the "Other Trees" category added complexity, encompassing a wide range of textures and traits. Dataset limitations like focusing only on natural forest environments and inaccuracies in tree delineations introduced noise and reduced the reliability of the metric results.

The findings of this study offer practical implications for forest management and ecological research. By enabling detailed TS classification, the proposed approach provides essential insights into forest composition and dynamics, supporting sustainable resource management and long-term planning without being forced to use field surveys or statistical sampling. The ability to analyze species-specific interactions and distributions facilitates a better understanding of competition patterns, biodiversity, and resilience to environmental changes. This framework can also assist in monitoring forest health, identifying stress factors, and conservation strategies. This work shows that DL is an essential tool for addressing challenges in sustainable forestry and climate change mitigation.

Future directions should address limitations discovered during this work by integrating multimodal data sources, such as DSM, LiDAR, and hyperspectral imagery, to enhance the model's ability to differentiate visually similar TS. Multi-temporal datasets capturing seasonal variations, particularly in autumn when leaf colors change, could improve the classification accuracy for deciduous trees and provide additional context for distinguishing species. Addressing class imbalance remains a critical priority, which could be achieved through targeted data augmentations, tailored training strategies for minority classes, or alternative loss functions. Expanding the model's applicability to diverse geographic regions and forest types is essential for broader transferability. Further, exploring other transformer or language model architectures in the context of TS classification could open up new possibilities to improve the training results.

These findings highlight the method's potential to support forest management by enabling detailed TS distribution, interactions, and vitality analysis. This research represents a step forward in leveraging DL for automated TS classification. While challenges remain, such as class imbalance and dataset limitations, the framework lays a foundation for future innovations. This approach can significantly enhance forest management and ecological research by combining semantic segmentation with ITC-based approaches. The insights gained from this work can contribute to biodiversity conservation and forest monitoring efforts, providing a pathway to better understand and manage forest ecosystems and ease sustainable usage.

A. Appendices

A.1. Overview of Selected Literature Research Papers

Table A.1.: State-of-the-art papers dealing with TS classification using tree crown imagery and DL. (Fabry-Perot interferometer (FPI), Hyperspectral (HS), Colour data (RGB), Unmanned aerial images (UAV))

Paper	**Focus**	**Spatial Res.**	**Spectral Res.**	**No. of Species**	**Type of Forest**	**DL Model**	**Other Info**
Ecke et al. (2024)	Multi-species	10 cm	Multi-spectral UAV	5	Bavaria, natural forest	EfficientNet	Time-series dataset
Ferreira et al. (2024)	Multi-species	15 cm	Multi-spectral UAV + LiDAR	6	Brazil, urban forest	ResUnet	
Ramesh et al. (2024)	Multi-species	1.81 cm, 2.02 cm	RGB	14	Quebec, Canada, temperate mixed forest	U-Net, DeepLabv3+, Mask2Former, 3D-UNet, U-TAE, Processor Module	Time-series images
Beloiu et al. (2023)	Single / multi-species	10 cm	RGB	4	Switzerland, natural forest	Faster R-CNN	
Cha et al. (2023)	Forest region	5, 10, 20, 60 m	Hyper-spectral satellite	9	South Korea, natural forest, plantation	U-Net	Different forest regions
Mäyrä et al. (2021)	Multi-species	0.5, 1 m	Hyper-spectral + LiDAR	4	Finland, mixed forest	3D-CNN	

Continued on the next page

Paper	Focus	Spatial Res.	Spectral Res.	No. of Species	Type of Forest	DL Model	Other Info
Pearse et al. (2021)	Single species	10 cm	RGB UAV	1	New Zealand, various sites	ResNet-50	Disease detection of one species
Zhang et al. (2021)	Single species		RGB UAV	10	China, various sites	AlexNet, VGG-16, ResNet-50	Near-infrared (NIR) spectral region used
Cao and Zhang (2020)	Forest region	20 cm	RGB	6	China, forest farm	U-Net, ResNet	
Egli and Höpke (2020)	Forest region	1.6 cm	RGB	4	Germany, research forest	Custom CNN	
Ferreira et al. (2020)	Multi-species	4 cm	RGB	3	Brazil, natural forest	ResNet-18 in DeepLabv3+	Palm species only
Nezami et al. (2020)	Multi-species	2.3 cm (RGB), 8.6 cm (FPI)	RGB + HS	3	Finland, mixed forest	3D-CNN	
Schiefer et al. (2020)	Multi-species	2 cm	RGB	14	Southern Black Forest and Hainich Areas, Germany, temperate mixed forest	U-Net	Use of DSM
Sun et al. (2019)	Forest region	30 m	RGB, LiDAR, DSM (LAS)	17	China, tropical wetland	AlexNet, VGG16, ResNet50	

List of Figures

List of Tables

Acronyms

LiDAR Light Detection and Ranging. 10, 12, 13

LRZ Leibniz Supercomputing Center. 37

mIoU mean intersection over union. 40, 41, 43, 52, 57

ML machine learning. 4, 5

RGB red-green-blue. 2, 3, 8, 10, 12, 13, 19, 60, 65

RS remote sensing. 2, 4, 10, 12

SHP shape file. 17, 19, 20

TIF raster file. 17, 19, 20, 26, 28

TN true negatives. 39

TP true positives. 38–40, 43

TS tree species. 2–6, 8–13, 15–17, 19, 21, 23, 25, 26, 29, 38–40, 42, 43, 47, 52, 54, 55, 57–60, 62, 64, 65

UAV uncrewed aerial vehicle. 2, 4, 8, 10, 15, 60, 65

U-TAE U-Net with Temporal Attention Encoder. 13, 14

ViT vision transformer. 5, 6, 13

wandb Weights & Biases. 38

WCE Weighted Cross-Entropy. 32, 33

Bibliography

Aleissaee, A. A., A. Kumar, R. M. Anwer, S. Khan, H. Cholakkal, G.-S. Xia, and F. S. Khan (Jan. 2023). "Transformers in Remote Sensing: A Survey". In: *Remote Sensing* 15.7. Number: 7 Publisher: Multidisciplinary Digital Publishing Institute, p. 1860. ISSN: 2072-4292. DOI: 10.3390/rs15071860. URL: https://www.mdpi.com/2072-4292/15/7/1860 (visited on 10/25/2024).

Beloiu, M., L. Heinzmann, N. Rehush, A. Gessler, and V. C. Griess (Jan. 2023). "Individual Tree-Crown Detection and Species Identification in Heterogeneous Forests Using Aerial RGB Imagery and Deep Learning". In: *Remote Sensing* 15.5. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute, p. 1463. ISSN: 2072-4292. DOI: 10.3390/rs15051463. URL: https://www.mdpi.com/2072-4292/15/5/1463 (visited on 07/22/2024).

"Der Wald in Deutschland: Ausgewählte Ergebnisse der vierten Bundeswaldinventur" (Oct. 2024). In: ed. by BMEL.

Burslem, D. and M. D. Swaine (Mar. 1, 2003). "Associations between tree growth, soil fertility and water availability at local and regional scales in Ghanaian tropical rain forest". In: *Journal of Tropical Ecology* 19, pp. 109–125. DOI: 10.1017/S0266467403003146.

Cao, K. and X. Zhang (Jan. 2020). "An Improved Res-UNet Model for Tree Species Classification Using Airborne High-Resolution Images". In: *Remote Sensing* 12.7. Number: 7 Publisher: Multidisciplinary Digital Publishing Institute, p. 1128. ISSN: 2072-4292. DOI: 10.3390/rs12071128. URL: https://www.mdpi.com/2072-4292/12/7/1128 (visited on 07/22/2024).

Cha, S., J. Lim, K. Kim, J. Yim, and W.-K. Lee (Aug. 2023). "Deepening the Accuracy of Tree Species Classification: A Deep Learning-Based Methodology". In: *Forests* 14.8. Number: 8 Publisher: Multidisciplinary Digital Publishing Institute, p. 1602. ISSN: 1999-4907. DOI: 10.3390/f14081602. URL: https://www.mdpi.com/1999-4907/14/8/1602 (visited on 07/22/2024).

Chen, J., X. Liang, Z. Liu, W. Gong, Y. Chen, J. Hyyppä, A. Kukko, and Y. Wang (Nov. 1, 2024). "Tree species recognition from close-range sensing: A review". In: *Remote Sensing of Environment* 313, p. 114337. ISSN: 0034-4257. DOI: 10.1016/j.rse.2024.114337. URL:

https://www.sciencedirect.com/science/article/pii/S0034425724003559 (visited on 10/16/2024).

Chen, L.-C., Y. Zhu, G. Papandreou, F. Schroff, and H. Adam (Aug. 22, 2018). *Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation*. Number: arXiv:1802.02611 version: 3. DOI: 10.48550/arXiv.1802.02611. arXiv: 1802.02611. URL: http://arxiv.org/abs/1802.02611 (visited on 10/25/2024).

Cheng, B., I. Misra, A. G. Schwing, A. Kirillov, and R. Girdhar (June 15, 2022). *Masked-attention Mask Transformer for Universal Image Segmentation*. Number: arXiv:2112.01527. DOI: 10.48550/arXiv.2112.01527. arXiv: 2112.01527. URL: http://arxiv.org/abs/2112.01527 (visited on 10/25/2024).

Ecke, S., F. Stehr, J. Frey, D. Tiede, J. Dempewolf, H.-J. Klemmt, E. Endres, and T. Seifert (Apr. 1, 2024). "Towards operational UAV-based forest health monitoring: Species identification and crown condition assessment by means of deep learning". In: *Computers and Electronics in Agriculture* 219, p. 108785. ISSN: 0168-1699. DOI: 10.1016/j.compag.2024.108785. URL: https://www.sciencedirect.com/science/article/pii/S0168169924001765 (visited on 07/22/2024).

Egli, S. and M. Höpke (Jan. 2020). "CNN-Based Tree Species Classification Using High Resolution RGB Image Data from Automated UAV Observations". In: *Remote Sensing* 12.23. Number: 23 Publisher: Multidisciplinary Digital Publishing Institute, p. 3892. ISSN: 2072-4292. DOI: 10.3390/rs12233892. URL: https://www.mdpi.com/2072-4292/12/23/3892 (visited on 07/22/2024).

Fare Garnot, V. S. and L. Landrieu (Oct. 2021). "Panoptic Segmentation of Satellite Image Time Series with Convolutional Temporal Attention Networks". In: *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*. 2021 IEEE/CVF International Conference on Computer Vision (ICCV). Montreal, QC, Canada: IEEE, pp. 4852–4861. ISBN: 978-1-66542-812-5. DOI: 10.1109/ICCV48922.2021.00483. URL: https://ieeexplore.ieee.org/document/9711189/ (visited on 10/25/2024).

Fassnacht, F. E., H. Latifi, K. Stereńczak, A. Modzelewska, M. Lefsky, L. T. Waser, C. Straub, and A. Ghosh (Dec. 1, 2016). "Review of studies on tree species classification from remotely sensed data". In: *Remote Sensing of Environment* 186, pp. 64–87. ISSN: 0034-4257. DOI: 10.1016/j.rse.2016.08.013. URL: https://www.sciencedirect.com/science/article/pii/S0034425716303169 (visited on 08/12/2024).

Ferreira, M. P., D. R. A. d. Almeida, D. d. A. Papa, J. B. S. Minervino, H. F. P. Veras, A. Formighieri, C. A. N. Santos, M. A. D. Ferreira, E. O. Figueiredo, and E. J. L. Ferreira (Nov. 1, 2020). "Individual tree detection and species classification of Amazonian palms using UAV images and deep learning". In: *Forest Ecology and Management* 475, p. 118397. ISSN: 0378-1127. DOI: 10.1016/j.foreco.2020.118397. URL: https://www.sciencedirect.com/science/article/pii/S037811272031166X (visited on 07/22/2024).

Ferreira, M. P., D. R. dos Santos, F. Ferrari, L. C. T. C. Filho, G. B. Martins, and R. Q. Feitosa (Apr. 1, 2024). "Improving urban tree species classification by deep-learning based fusion of digital aerial images and LiDAR". In: *Urban Forestry & Urban Greening* 94, p. 128240. ISSN: 1618-8667. DOI: 10.1016/j.ufug.2024.128240. URL: https://www.sciencedirect.com/science/article/pii/S1618866724000384 (visited on 07/22/2024).

Jadon, S. (Sept. 3, 2020). *A survey of loss functions for semantic segmentation*. Number: arXiv:2006.14822. arXiv: 2006.14822. URL: http://arxiv.org/abs/2006.14822 (visited on 10/31/2024).

Kattenborn, T., J. Eichel, and F. Fassnacht (Nov. 27, 2019). "Convolutional Neural Networks enable efficient, accurate and fine- grained segmentation of plant species and communities from high-resolution UAV imagery". In: *Scientific Reports* 10, p. 17656. DOI: 10.1038/s41598-019-53797-9.

Lin, T.-Y., M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick (2014). "Microsoft COCO: Common Objects in Context". In: *Lecture Notes in Computer Science* 8693, pp. 740–755. DOI: 10.1007/978-3-319-10602-1_48.

Mäyrä, J., S. Keski-Saari, S. Kivinen, T. Tanhuanpää, P. Hurskainen, P. Kullberg, L. Poikolainen, A. Viinikka, S. Tuominen, T. Kumpula, and P. Vihervaara (Apr. 1, 2021). "Tree species classification from airborne hyperspectral and LiDAR data using 3D convolutional neural networks". In: *Remote Sensing of Environment* 256, p. 112322. ISSN: 0034-4257. DOI: 10.1016/j.rse.2021.112322. URL: https://www.sciencedirect.com/science/article/pii/S0034425721000407 (visited on 07/22/2024).

Natesan, S., C. Armenakis, and U. Vepakomma (2020). *Individual tree species identification using Dense Convolutional Network (DenseNet) on multitemporal RGB images from UAV*. URL: https://cdnsciencepub.com/doi/10.1139/juvs-2020-0014 (visited on 08/12/2024).

Nezami, S., E. Khoramshahi, O. Nevalainen, I. Pölönen, and E. Honkavaara (Jan. 2020). "Tree Species Classification of Drone Hyperspectral and RGB Imagery with Deep Learning Convolutional Neural Networks". In: *Remote Sensing* 12.7. Number: 7 Publisher: Multidisciplinary Digital Publishing Institute, p. 1070. ISSN: 2072-4292. DOI: 10.3390/rs12071070. URL: https://www.mdpi.com/2072-4292/12/7/1070 (visited on 07/22/2024).

Pearse, G. D., M. S. Watt, J. Soewarto, and A. Y. S. Tan (Jan. 2021). "Deep Learning and Phenology Enhance Large-Scale Tree Species Classification in Aerial Imagery during a

Biosecurity Response". In: *Remote Sensing* 13.9. Number: 9 Publisher: Multidisciplinary Digital Publishing Institute, p. 1789. ISSN: 2072-4292. DOI: 10.3390/rs13091789. URL: https://www.mdpi.com/2072-4292/13/9/1789 (visited on 07/22/2024).

Ramesh, V., A. Ouaknine, and D. Rolnick (July 17, 2024). *Tree semantic segmentation from aerial image time series*. Number: arXiv:2407.13102. DOI: 10.48550/arXiv.2407.13102. arXiv: 2407.13102[cs]. URL: http://arxiv.org/abs/2407.13102 (visited on 08/22/2024).

Ronneberger, O., P. Fischer, and T. Brox (May 18, 2015). *U-Net: Convolutional Networks for Biomedical Image Segmentation*. Number: arXiv:1505.04597 version: 1. DOI: 10.48550/arXiv.1505.04597. arXiv: 1505.04597[cs]. URL: http://arxiv.org/abs/1505.04597 (visited on 07/22/2024).

Schiefer, F., T. Kattenborn, A. Frick, J. Frey, P. Schall, B. Koch, and S. Schmidtlein (Dec. 1, 2020). "Mapping forest tree species in high resolution UAV-based RGB-imagery by means of convolutional neural networks". In: *ISPRS Journal of Photogrammetry and Remote Sensing* 170, pp. 205–215. ISSN: 0924-2716. DOI: 10.1016/j.isprsjprs.2020.10.015. URL: https://www.sciencedirect.com/science/article/pii/S0924271620302938 (visited on 08/12/2024).

Sun, Y., J. Huang, Z. Ao, D. Lao, and Q. Xin (Nov. 2019). "Deep Learning Approaches for the Mapping of Tree Species Diversity in a Tropical Wetland Using Airborne LiDAR and High-Spatial-Resolution Remote Sensing Images". In: *Forests* 10.11. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute, p. 1047. ISSN: 1999-4907. DOI: 10.3390/f10111047. URL: https://www.mdpi.com/1999-4907/10/11/1047 (visited on 07/22/2024).

Tedrake, R. (2024). *Robotic Manipulation. Perception, Planning, and Control*. URL: http://manipulation.mit.edu.

terrabyte Documentation (2024). URL: https://docs.terrabyte.lrz.de/ (visited on 07/22/2024).

Tian, J., D. Panangian, W. Fan, B. Siegmann, and X. Yuan (2025). "Deep learning based individual tree crown delineation from panchromatic aerial imagery". In: *ISPRS Geospatial Week 2025*. Submitted.

Tragakis, A., C. Kaul, R. Murray-Smith, and D. Husmeier (Jan. 29, 2023). "The Fully Convolutional Transformer for Medical Image Segmentation". In: *2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*. 2023 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV). Waikoloa, HI, USA: IEEE, pp. 3649–3658. ISBN: 978-1-66549-346-8. DOI: 10.1109/WACV56688.2023.00365. URL: https://ieeexplore.ieee.org/document/10030969/ (visited on 08/22/2024).

Troles, J., U. Schmid, W. Fan, and J. Tian (Jan. 2024). "BAMFORESTS: Bamberg Benchmark Forest Dataset of Individual Tree Crowns in Very-High-Resolution UAV Images". In: *Remote*

Sensing 16.11. Number: 11 Publisher: Multidisciplinary Digital Publishing Institute, p. 1935. ISSN: 2072-4292. DOI: 10.3390/rs16111935. URL: https://www.mdpi.com/2072-4292/16/11/1935 (visited on 07/22/2024).

Zhang, C., K. Xia, H. Feng, Y. Yang, and X. Du (Oct. 1, 2021). "Tree species classification using deep learning and RGB optical images obtained by an unmanned aerial vehicle". In: *Journal of Forestry Research* 32.5, pp. 1879–1888. ISSN: 1993-0607. DOI: 10.1007/s11676-020-01245-0. URL: https://doi.org/10.1007/s11676-020-01245-0 (visited on 07/22/2024).

Zhong, L., Z. Dai, P. Fang, Y. Cao, and L. Wang (May 2024). "A Review: Tree Species Classification Based on Remote Sensing Data and Classic Deep Learning-Based Methods". In: *Forests* 15.5. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute, p. 852. ISSN: 1999-4907. DOI: 10.3390/f15050852. URL: https://www.mdpi.com/1999-4907/15/5/852 (visited on 10/09/2024).