



**University of
Zurich**^{UZH}

Tree Species Classification from AVIRIS-NG Hyperspectral Imagery using Convolutional Neural Networks

GEO 511 Master's Thesis

Author

Benjamin Zehnder
17-711-177

Supervised by

Dr. Anna Schweiger

Faculty representative

Prof. Dr. Alexander Damm

30.04.2022

Department of Geography, University of Zurich



**Universität
Zürich** ^{UZH}

GEO 511
Master Thesis
April 30, 2022

Tree Species Classification from AVIRIS-NG Hyperspectral Imagery using Convolutional Neural Networks

Benjamin Zehnder
17-711-177

Supervised by:
Dr. Anna K. Schweiger

Faculty representative:
Prof. Dr. Alexander Damm

Remote Sensing Laboratories
Department of Geography
University of Zurich

Abstract

This study focuses on the automatic classification of tree species using a three-dimensional convolutional neural network (CNN) based on field-sampled ground reference data, a LiDAR point cloud and AVIRIS-NG airborne hyperspectral remote sensing imagery with 2 m spatial resolution acquired on 14 June 2021. I created a tree species map for my 10.4 km² study area which is located in the Jurapark Aargau, a Swiss regional park of national interest. I collected ground reference data for six major tree species present in the study area (*Quercus robur*, *Fagus sylvatica*, *Fraxinus excelsior*, *Pinus sylvestris*, *Tilia platyphyllos*, total n = 331). To match the sampled ground reference to the AVIRIS-NG 425 band hyperspectral imagery, I delineated individual tree crowns (ITCs) from a canopy height model (CHM) based on LiDAR point cloud data. After matching the ground reference data to the hyperspectral imagery, I split the extracted image patches to training, validation, and testing subsets. The amount of training, validation and testing data was increased by applying image augmentation through rotating, flipping, and changing the brightness of the original input data. The classifier is a CNN trained on the first 32 principal components (PC's) extracted from AVIRIS-NG data. The CNN uses image patches of 5 × 5 pixels and consists of two convolutional layers and two fully connected layers. The latter of which is responsible for the final classification using the *softmax* activation function. The results show that the CNN classifier outperforms comparable conventional classification methods. The CNN model is able to predict the correct tree species with an overall accuracy of 70% and an average F1-score of 0.67. A random forest classifier reached an overall accuracy of 67% and an average F1-score of 0.61 while a support-vector machine classified the tree species with an overall accuracy of 66% and an average F1-score of 0.62. This work highlights that CNNs based on imaging spectroscopy data can produce highly accurate high resolution tree species distribution maps based on a relatively small set of training data thanks to the high dimensionality of hyperspectral images and the ability of CNNs to utilize spatial and spectral features of the data. These maps provide valuable input for modelling the distributions of other plant and animal species and ecosystem services. In addition, this work illustrates the importance of direct collaboration with environmental practitioners to ensure user needs are met. This aspect will be evaluated further in future work by assessing how these products are used by environmental practitioners and as input for modelling purposes.

Contents

Abstract	I
List of Figures	VI
List of Tables	IX
Abbreviations	XI
1 Introduction	1
1.1 Background	1
1.2 Research Context	2
1.3 Research Objectives	4
2 Study Area	5
3 Data	7
3.1 Hyperspectral Imagery	7
3.2 LiDAR Data	9
3.3 Ground Reference Data	10
4 Methods	13
4.1 Workflow	13
4.2 Preprocessing Hyperspectral Imagery	14
4.3 CNN Input Data Preparation	15
4.3.1 CHM Computation	15
4.3.2 ITC Delineation	17
4.3.3 Field Data Matching	19
4.3.4 Image Patch Extraction	20
4.3.5 Dimensionality Reduction	22
4.3.6 Data Augmentation	23
4.3.7 Patch Equalization	25
	III

4.4	CNN Model	25
4.4.1	Model Architecture	25
4.4.2	Model Training	29
4.4.3	Full Study Area Classifications	30
4.5	Accuracy Assessment	30
5	Results	31
5.1	Model Training	31
5.2	Classification Results	32
5.3	Full Study Area Classifications	35
6	Discussion	36
6.1	Ground Reference Data	36
6.2	Image Patch Extraction	37
6.3	CNN Model	38
6.4	Classification Accuracy	39
7	Conclusion	44
A	Appendix – Additional Figures	53

List of Figures

FIGURE 1.1. Conceptual visualization of a CNN	3
FIGURE 2.1. Study area.....	5
FIGURE 3.1. AVIRIS-NG spectrometer and its airborne platform	7
FIGURE 3.2. AVIRIS-NG flight strips.....	8
FIGURE 3.3. 100 m × 100 m subset of the LiDAR point cloud.....	9
FIGURE 3.4. Trimble R7 GNSS.....	10
FIGURE 3.5. Study area and location of sampled ground reference data.....	11
FIGURE 4.1. Workflow overview	13
FIGURE 4.2. Image co-registration of AVIRIS-NG imagery and SWISSIMAGE orthophoto	14
FIGURE 4.3. LiDAR point density	16
FIGURE 4.4. Comparison of three CHM algorithms.....	17
FIGURE 4.5. Subset of the CHM showing delineated ITCs.....	19
FIGURE 4.6. Field data matching.....	20
FIGURE 4.7. Conceptual workflow of image patch extraction	21
FIGURE 4.8. Average reflectances per target tree species	22
FIGURE 4.9. CNN architecture	26
FIGURE 4.10. The rectified linear unit function	28
FIGURE 5.1. Accuracy and loss CNN training	31
FIGURE 5.2. Confusion matrix for the trained CNN	32
FIGURE 5.3. Confusion matrices for random forest and support-vector machine.....	34
FIGURE 5.4. Full study area tree species classification map	35
FIGURE 6.1. PLS-DA probability plot.....	40
FIGURE 6.2. Subset of the study area showing forest types and the CNN predictions.....	42
FIGURE 6.3. Long-Term contracts for the preservation of forest in the Canton of Aargau	43
FIGURE A.1. Average reflectance spectra of oak samples	53
FIGURE A.2. Average reflectance spectra of beech samples.....	54
FIGURE A.3. Average reflectance spectra of ash samples.....	54
FIGURE A.4. Average reflectance spectra of pine samples	55

FIGURE A.5. Average reflectance spectra of linden samples	55
FIGURE A.6. Cumulative Beta distribution	56
FIGURE A.7. Confusion matrix for the trained CNN.....	57
FIGURE A.8. Confusion matrix for the trained CNN.....	58
FIGURE A.9. Confusion matrix for the random forest classifie.....	59
FIGURE A.10. Confusion matrix for the random forest classifie.....	60
FIGURE A.11. Confusion matrix for the support-vector machine	61
FIGURE A.12. Confusion matrix for the support-vector machine	62
FIGURE A.13. Raw full study area tree species classification map	63
FIGURE A.14. Full study area prediction confidence	64
FIGURE A.15. Full study area NDVI.....	65
FIGURE A.16. RGB image of the AVIRIS-NG hyperspectral imagery.	66

List of Tables

TABLE 3.1 Number of sampled trees	11
TABLE 4.1. Input parameter values for the ITC delineation algorithm	18
TABLE 4.2. Number of training patches per tree species before and after MixUp.....	24
TABLE 4.3. Number of labeled pixels before and after image augmentation.....	25
TABLE 5.1. Classification report for the trained CNN.....	33
TABLE 5.2. Classification report for RF and SVM.....	34
TABLE 6.1. Mean tree height and mean crown size per species.....	41

Abbreviations

Adam	Adaptive Moment Estimation
ALS	Airborne Laser Scanning
ANN	Artificial Neural Network
AOI	Area of Interest
API	Application Programming Interface
ASPRS	American Society for Photogrammetry and Remote Sensing
AVIRIS-NG	Airborne Visible InfraRed Imaging Spectrometer - Next Generation
BN	Batch Normalization
Cal/Val	Calibration & Validation
CHM	Canopy Height Model
CHIME	Copernicus Hyperspectral Imaging Mission for the Environment
CNN	Convolutional Neural Network
CRS	Coordinate Reference System
DL	Deep Learning
ESA	European Space Agency
GNSS	Global Navigation Satellite System
HSI	Hyperspectral Imaging
ITC	Individual Tree Crown
JPA	Jurapark Aargau
JPL	Jet Propulsion Laboratory
LiDAR	Light Detection and Ranging
ML	Machine Learning
MLC	Maximum Likelihood Classifiers
NASA	National Aeronautics and Space Administration
PCA	Principal Component Analysis
ReLU	Rectified Linear Unit
RF	Random Forest
RTK	Real Time Kinematic
RS	Remote Sensing

RSME	Root Mean Square Error
SBG	Surface Biology and Geology
SNR	Signal-to-Noise Ratio
SVM	Support-Vector Machine
SWIR	Shortwave Infrared
TOC	Top-of-Canopy
UZH	University of Zurich
VNIR	Visible and Near-Infrared

1 Introduction

1.1 Background

Globally among all ecosystem types, forests are of high relative importance for the mitigation of climate change and play a crucial role in biodiversity conservation [1]. Thus, the development of cost and time efficient methods for extracting information about forest ecosystems is needed and can be of great value for forest practitioners. Forest ecologists, land managers, and commercial enterprises could make use of accurate tree species composition assessments to study biodiversity patterns, estimate timber stocks, or improve estimates of forest fire risks [2]. In recent years a lot of literature has been published on the use of remote sensing data for forest applications. This development is facilitated through improvements in operational remote sensing and field sensors which need new classification tools to bridge the gap between data-rich remote sensing imagery and the need for high-resolution information about forests [2]. Due to their high spectral resolution hyperspectral imagery provides valuable information for tree species classification. Hyperspectral data cover a continuous spectral range with hundreds of narrow bands ranging from the visible and near-infrared (VNIR) wavelength region (400–1100 nm), up to the shortwave infrared area (SWIR) at 1100–2500 nm [3]. Specific vegetation characteristics can be related to spectral features in the data and are thus vital for the discrimination and classification of tree [4]. Recent studies using hyperspectral data for tree species classification tasks are for example Fassnacht et al. [5] or Dalponte et al. [6]. However, most studies on the topic of tree species classification utilize LiDAR (Light Detection and Ranging) data in addition to hyperspectral imagery (e.g., [3], [4], [7]–[11]). LiDAR data is a great complementary data source to multispectral or hyperspectral imagery (HSI) as it provides information on the vertical structure of forests and can thus be useful for the description and delineation of forest characteristics related to height [4]. In tree species classification studies, LiDAR point clouds are predominantly used to derive a canopy height model (CHM) which automated algorithms use as an input to delineate individual tree crown (ITC) polygons (e.g., [7], [8]). ITC delineation is crucial in matching the sampled ground reference data to the AVIRIS-NG (Airborne Visible InfraRed Imaging Spectrometer - Next Generation) HSI to extract the needed data for training and validating the classification model. The outcome of the classification approach depends directly on the selected classification method. Machine learning (ML) methods like support-vector machines (SVMs), random forests (RFs), and artificial or convolutional neural networks (ANNs & CNNs) have been especially popular for tree species discrimination from remote sensing imagery [3]. Of those, Fassnacht et al. [12] identified support-vector machines (SVMs) as the most common method for tasks of this type.

In this work a CNN will be implemented for the tree species classification task at hand to contribute to the currently ongoing ValPar.CH project. This project aims at quantifying the ecological infrastructure of multiple Swiss parks of national interest and its value to nature, society, and economy (valpar.ch). One of the inspected sites is the Jurapark Aargau (JPA), a regional park of national importance located between Brugg, Laufenfeld, Rheinfelden, and Aarau (jurapark-aargau.ch). The JPA is also one of two ValPar.CH sites which have been part of this summer's ESA CHIME (Copernicus Hyperspectral Imaging Mission for the Environment) campaign to collect data for various targets across Europe with the AVIRIS-NG (Airborne Visible InfraRed Imaging Spectrometer - Next Generation) sensor owned by NASA (National Aeronautics and Space Administration) and operated by the JPL (Jet Propulsion Laboratory). The hyperspectral imagery on which the tree species classification task will be conducted in this work was acquired over the JPA on June 14, 2021. Creating such a tree species classification map of the main tree species present in the JPA has been defined as the overall aim of this work together with representatives of the JPA. As Fassnacht et al. (2016) state, such direct collaboration with environmental practitioners is a way to ensure that research efforts match the needs of the user. The needed ground reference data was acquired over the course of September 2021 accumulating to a total of 331 trees of which the exact position and their species was captured. I collected Ground reference data sampled across my study area which is a spatial subset of the JPA with a small part of it being located outside the geographic extent of the JPA.

1.2 Research Context

Classification algorithms traditionally used in Remote Sensing can be grouped in two big categories: unsupervised and supervised. Unsupervised classifiers use remotely sensed data and divide it into a number of statistical clusters in multi- or hyperspectral space. This means that these methods do not require information (i.e. labeled data) on the study area prior to the classification [13]. Supervised classifiers on the other hand do need prior information on the ground cover of the study area. Such information could be labeled data samples indicating the ground cover at certain points within the study area. Spectral signatures of sampled data are used to train the classifier which later will be utilized to predict over the whole area of interest (AOI) to generate a final classification image [14].

In early studies of tree classification using RS data the most widely classifiers were unsupervised clustering methods such as K-means or ISODATA and supervised methods such as maximum likelihood classifiers (MLC) [12]. Facilitated by improvements in computational capacity in hardware and software after 1995, non-parametric decision tree based classifiers and neural networks emerged as supervised alternatives to the common algorithms [12]. Since then, such machine learning (ML) classifiers have become a major-focus in the remote sensing literature [15]. Maxwell et al. [15] describe ML algorithms as being able to model complex data signatures based

on a variety of input data without making assumptions about the data distribution (i.e., non-parametric: data does not need to be normally distributed).

While ML methods are still being widely used, a sub-discipline of ML is emerging. Deep learning (DL) methods such as convolutional neural networks (CNNs) have already shown great potential in RS classification with various software packages being made available to implement these algorithms [15]. In DL, classifiers aim at learning data representations from an image input while ML methods search for useful representations of the data within a space of possibilities using guidance from a feedback signal [16]. In the case of a tree species classification task, the feedback signal is the fraction of correctly classified samples in a validation set that are not used for model training. For remote sensing tasks, DL classifiers are often framed within the field of artificial neural networks (ANNs) [17]. The concept of ANNs is inspired by the neural connections in the human brain and the pulses that travel through it to transmit information [18]. The ‘deep’ in deep learning refers to the depth of a model which is the number of hidden layers an ANN consists of. Each hidden layer is composed by a set of “neurons”, that apply weights over the input data and whose responses are filtered, determining the “neural activations” which will be forwarded to the next hidden layer [18].

The hierarchical nature of such DL architectures allows to extract high-level, hierarchical, and abstract features [19]. More specifically, at early stages in the learning process general features are extracted with the final layers being able to learn pieces of knowledge that are more related to the application at hand [18]. In other words, ANNs are a mathematical framework for learning representations from data with the extracted patterns getting more and more complex with each hidden layer.

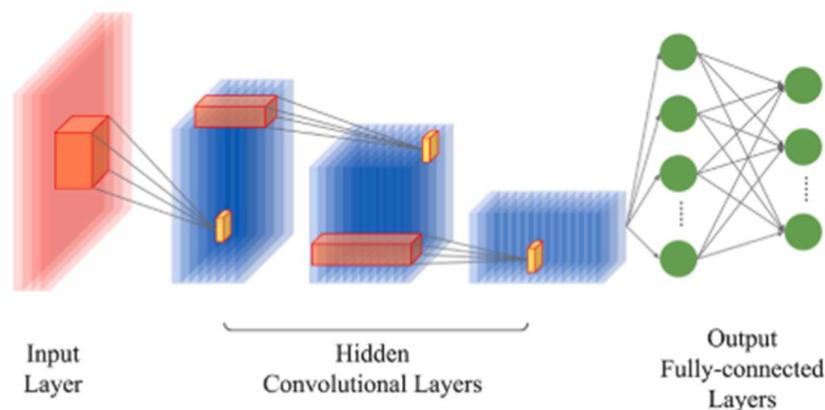


FIGURE 1.1. Conceptual visualization of a CNN [18].

In this thesis, I used a convolutional neural network (CNN) for the tree species classification. The structural unit of CNNs are the convolutional layers composed of several convolutional kernels

which are used to compute different feature maps [18], [20]. A feature map highlights a certain aspect like a shape or a pattern of the input image. As visualized in FIGURE 1.1, in a CNN the convolutional layers are usually followed by a number of fully-connected layers which take all neurons in the previous layers and connect them to every single neuron of the current layer to generate global semantic information, i.e. combining the extracted information to output class probabilities for each input [20]. When using hyperspectral imagery (HSI), the strength of CNNs is their ability to integrate spectral features with spatial-contextual information in a very efficient way [18]. Also, CNNs offer a lot of flexibility regarding the dimensionality of the single layers and their ability to make strong assumptions about the input images [21]. According to Gu et al. [20], these factors have turned CNNs into one of the most successful and popular NN models, making them the current state-of-the-art in DL and extremely popular for HSI data classification [18] including tree species discrimination [2], [3], [22]–[24].

1.3 Research Objectives

The aim of this thesis is to create a pixel-level classification map of the main tree species for our study area within the Jurapark Aargau (JPA). The idea is to exploit the capability of high resolution AVIRIS-NG hyperspectral imagery to discriminate tree species with a CNN and the aid of LiDAR data. The created product should be of such quality that it could be used as input for species distribution and ecosystem service modelling. These tasks which important for the ValPar.ch project aiming at assessing the ecological infrastructure within the park. In addition, I am also aiming to add to the literature and describe how DL methods and especially CNNs can further improve our ability to extract information from remotely sensed data. More formally, the defined research objectives of this thesis are:

1. Creating a tree species distribution map of the main tree species present in my study area within the JPA that can be used as an input for species distribution and ecosystem service modelling.
2. Adding to the understanding of how deep learning can improve our ability to extract information from remotely sensed data.

How and if these objectives are met is presented in the remainder of this work which is divided into five main chapters. In chapter 2 the study area is introduced, a summary of the data sources follows in chapter 3. The methods used in this thesis are presented and explained in chapter 4. Chapter 5 presents the obtained results which are being discussed in chapter 6. Finally, the thesis is rounded off with a summary, the main takeaways, and an outline of potential further research in chapter 7.

2 Study Area

My study area lies in the southern part of the JPA just north of Aarau. The JPA is a regional park of national importance located between Aarau, Brugg, Laufenfeld, and Rheinfelden spanning over the cantons of Aargau and Solothurn (jurapark-aargau.ch). The topography of the JPA is characterized by the hilly Table Jurassic landscapes. The JPA has a total size of 299 km² with roughly half of it being covered by forest as stated on the park's website [25]. The study area has a size of 10.4 km² and consists of multiple forest areas that are not connected to each other. As can be seen in FIGURE 2.1, a small part in the southeast of the study area is outside of the geographic extent of the JPA.

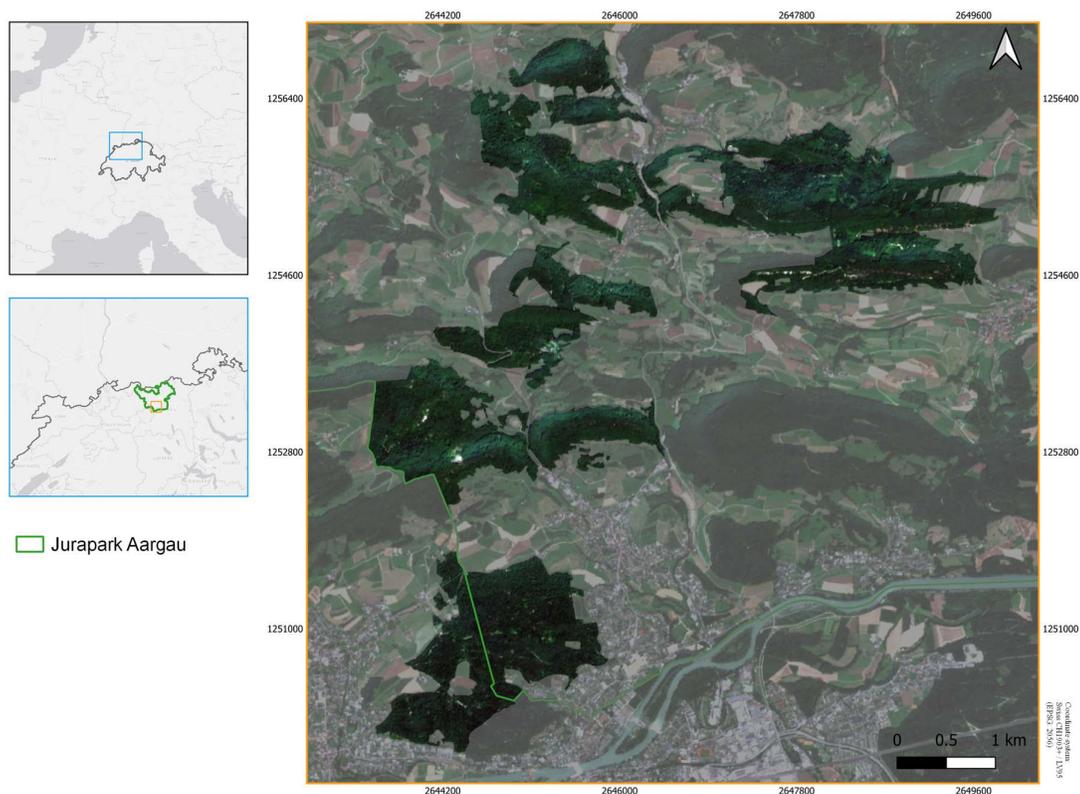


FIGURE 2.1. a) Map of Switzerland highlighting the extent of inset map b) in light blue. b) Extent and location of the JPA highlighted in green. c) Study area highlighted in RGB with semi-transparent background ESA Sentinel-2 remote sensing imagery from July 23rd, 2021.

The Canton of Aargau keeps and publishes information about the cantons forest areas online [26]. The according plant sociological map of the study area shows that the spatially most prominent

forest communities are different sorts of beech forests with the largest part being classified as lime-beech forests. One part of the study area in the south is a larger oak forest reserve. Other types of forest communities present are oak-pine forests, linden-maple forests, and alder-ash forests. Together with a representative of the JPA the main tree species present in the study area were defined. These are the species that will be classified and for which field data has been sampled. The species included are Common oak (*Quercus robur*), beech (*Fagus sylvatica*), ash (*Fraxinus excelsior*), European Red pine (*Pinus sylvestris*), and Largeleaf linden (*Tilia platyphyllos*).

3 Data

3.1 Hyperspectral Imagery

In summer 2021 the ESA (European Space Agency) CHIME (Copernicus Hyperspectral Imaging Mission for the Environment) & SGB (Surface Biology and Geology) Mission took place. This mission is part of calibration and validation (Cal/Val) activities for ESA's and NASA's upcoming CHIME (Copernicus Hyperspectral Imaging Mission for the Environment) and SBG (Surface Biology and Geology) satellites respectively. The goal of these satellite missions is to provide imaging spectroscopy data at global coverage at regular intervals of time with high spatial resolution [27]. Such Cal/Val missions aim at determining the quality and integrity of the data provided by the spectrometers which are to be deployed on the currently being developed satellites [27]. The summer 2021 Cal/Val mission was a joint effort between ESA, NASA (National Aeronautics and Space Administration), JPL (Jet Propulsion Laboratory), and the University of Zurich (UZH). The sensor of choice was the Airborne Visible InfraRed Imaging Spectrometer - Next Generation (AVIRIS-NG) (FIGURE 3.1) which is owned by NASA and operated by the Jet Propulsion Laboratory (JPL). This sensor is a pushbroom spectral mapping system with high signal-to-noise ratio (SNR), designed for high performance spectroscopy [28].



FIGURE 3.1. The AVIRIS-NG spectrometer developed by NASA/JPL (left) and its airborne platform being used; the “Super King Air” research aircraft (right) [29].

The western part of the JPA was one of the target areas of the 2021 ESA CHIME & SBG mission. Data collection took place on Monday June 14, 2021, under clear sky conditions. The data was

collected across a total of ten flight strips of which nine are being used and cover the study area as shown in FIGURE 3.2.

The hyperspectral imagery (HSI) consists of a total of 425 spectral bands in the wavelength range of 380–2510 nm with a spectral sampling interval of 5 nm. The spatial resolution of the data ranges from 1.9–2.1 m depending on flight strip.

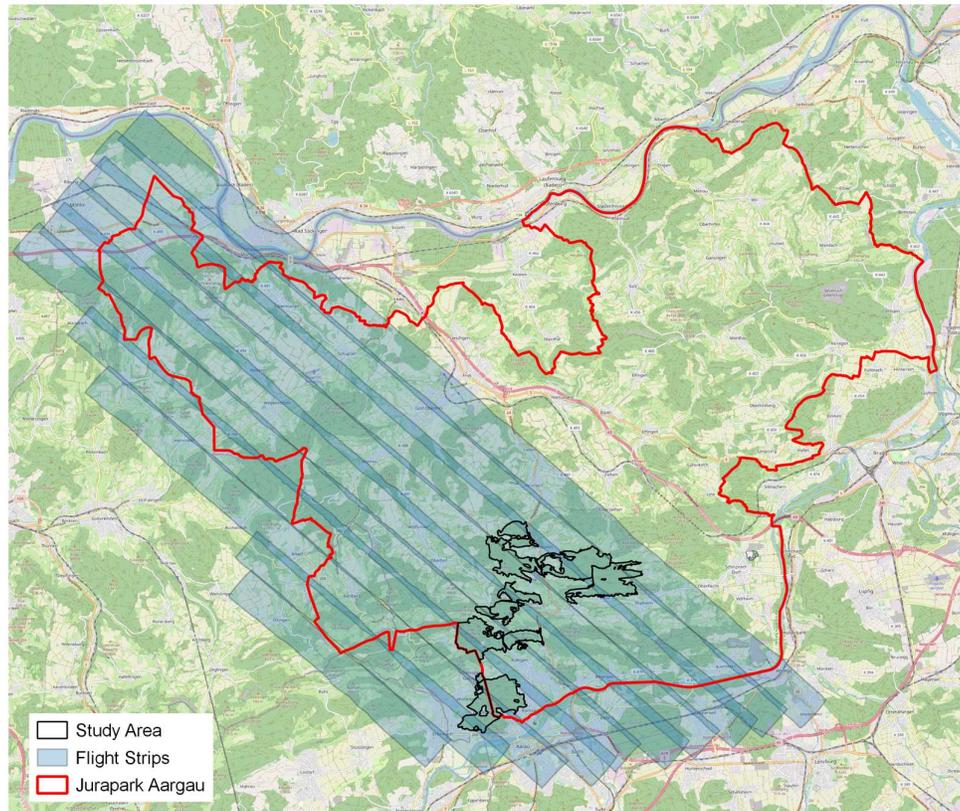


FIGURE 3.2. AVIRIS-NG data strips overlaying the perimeters of the JPA and my study area (Background: OpenStreetMap [30]).

Due to its high spectral resolution AVIRIS-NG HSI has been a popular data source in vegetation discrimination studies (e.g. [2], [19], [31], [32]). Ahmad et al. [33] demonstrated that it can be highly recommended to use AVIRIS-NG data for characterizing, quantifying, modelling, and mapping vegetation. Potential uncertainties in the data could come from natural and irreducible measurement noise or calibration uncertainties arising from systematic optical and electronic imperfections in the instrument [28].

3.2 LiDAR Data

The LiDAR data being used in this work are a subset of the swissSURFACE3D product of the Federal Office of Topography, Swisstopo [34]. swissSURFACE3D is a LiDAR point cloud describing the surface of Switzerland with all its natural and artificial elements. The LiDAR data was being acquired in different stages using an airborne sensor with the goal of finalizing the product until the end of 2023 [34]. The data being used for this project was acquired in 2020. The product is freely accessible for everyone [35] and delivers the x-, y-, and z-coordinate for each measured point in the CH1903+/LV95 (EPSG:2056) coordinate reference system. FIGURE 3.3 shows a $100\text{ m} \times 100\text{ m}$ height normalized subset of the LiDAR point cloud.

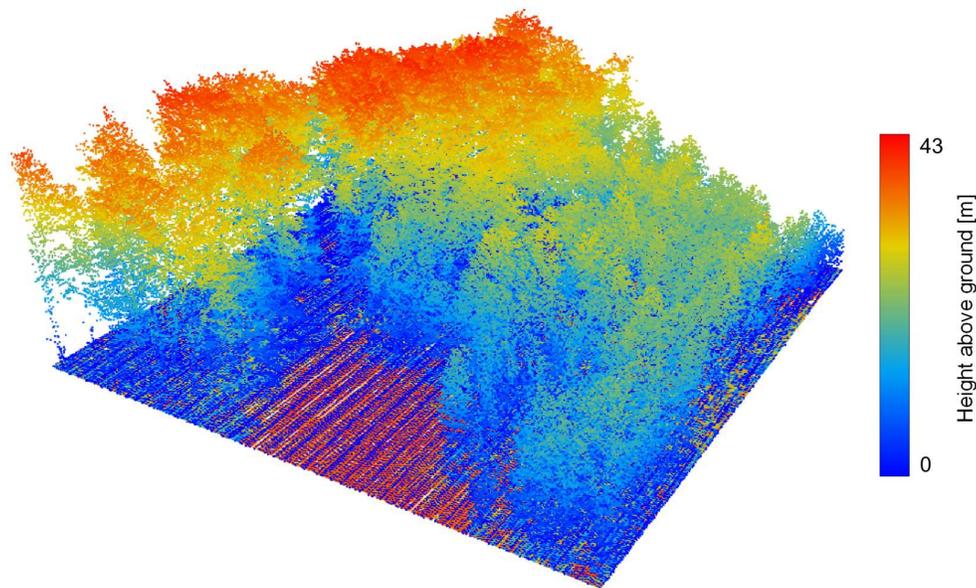


FIGURE 3.3. A $100\text{ m} \times 100\text{ m}$ height normalized subset of the LiDAR point cloud acquired over a part woodruff-beech part maple-ash forest located in the easternmost part of the study area.

The positional accuracy of the product is reported to be $\pm 20\text{ cm}$ horizontally and $\pm 10\text{ cm}$ vertically [34]. In terms of point density the minimum requirement are five points/ m^2 and the average is between 15 and 20 points/ m^2 [34]. However, the mean point density for my study area, is about 31 points/ m^2 . FIGURE 4.3 shows the spatial distribution of said point density. The point density is higher in certain regions which indicate the areas where the flight strips overlap.

3.3 Ground Reference Data

In October 2021, approximately three months after imagery acquisition, ground reference data was collected. I collected coordinate information of mature trees from each target species, to be subsequently matched to the HSI to train and evaluate the classification model. Sampling was conducted with a Trimble R7 GNSS (Global Navigation Satellite System) as depicted in FIGURE 3.4.



FIGURE 3.4. Trimble R7 GNSS used for the collection of ground reference data.

The GPS positions were acquired with the Real Time Kinematic (RTK) method. The principle of this method is that the controlling unit of the GNSS system connects via mobile internet with a reference station which is part of a continuous operating reference station network. This allows the setup of data links between the network server and the GNSS receiver to deliver the differential corrections needed to acquire high precision GPS positions [36]. However, due to a reliable internet connection not always being available some positions had to be acquired using the static measurement principle. In total the position of 331 single trees was collected (TABLE 3.1). There does not seem to be a clear consensus in literature on how many ground reference data points are needed for work of this nature. In studies the number of used data points ranges from 3.5 trees/km² as in Dalponte et al. [7] to around 1000 trees/km² as in Matsuki et al. [9] and Wu & Zhang [11]. The reference data density used in this thesis (34.8 trees/km²) is comparable to Fricker et al. [2] who used about 34 trees/km². In addition to acquiring the x-, y-, and z-coordinate of the trees, I

determined their species. I only sampled top-of-canopy (TOC) trees, i.e., trees with crowns visible from above which can be later identified in the HSI.

TABLE 3.1. Common and scientific name and the number of sampled trees used in this thesis.

Species	Number
Common Oak (<i>Quercus robur</i>)	59
Beech (<i>Fagus sylvatica</i>)	89
Ash (<i>Fraxinus excelsior</i>)	73
European Red pine (<i>Pinus sylvestris</i>)	61
Largeleaf Linden (<i>Tilia platyphyllos</i>)	49
	331

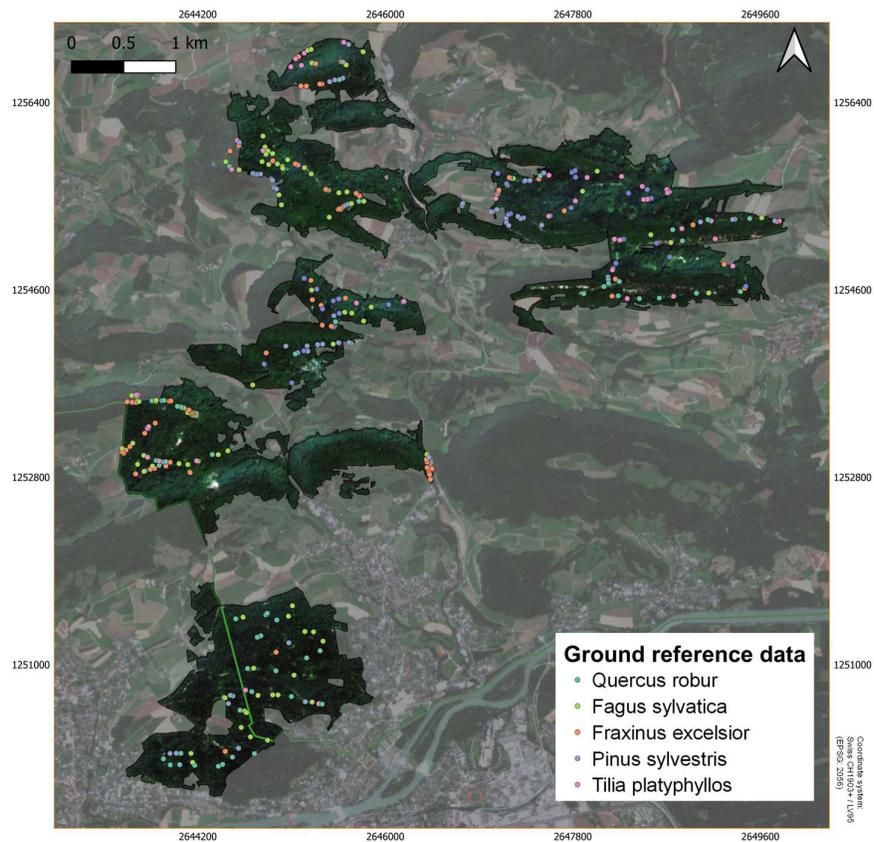


FIGURE 3.5. Study area and location of sampled ground reference data. Each data point represents a single tree. RGB background from ESA Sentinel-2 remote sensing imagery from July 23rd, 2021.

The goal was to have the measured tree positions distributed equally across the study area and across the different species. Both aims were not met optimally as can be seen in FIGURE 3.5. Due to the mostly hilly landscape of the study area reaching measurable trees further away from the paths was close to impossible. Additionally, not all areas of the study area were equally as accessible due to the available paths. Concerning distribution across tree species, the visually assessed relative abundance of the species influenced the number of measured trees per class. Most forest areas accessed were beech- and ash-dominant.

The mean horizontal positional accuracy of the sampled ground reference data is 0.546 m. This accuracy should fall into an acceptable range for the task at hand. Fricker et al. [2] discarded trees with a positional accuracy of over 1 m in their tree species classification study. The positional accuracy in the individual tree species mapping from Engler et al. [37] was reported to be at ± 2 m and Dalponte et al. [4] worked with a final uncertainty of the horizontal position of sample points of 1.70 m. To minimize the potential negative effect of the horizontal positional uncertainty the position of the tree was, if possible, measured directly under the crown and not at the stem. In addition, I only sampled mature trees with a crown diameter greater than approximately five meters to ensure that the measured data points are positioned under the respective tree crown even in the case of a larger positional error. This is important to later be able to match the data point to the correct crown in the aerial imagery. To further increase matching accuracy, I preferably sampled trees standing alone, if that was not possible, trees surrounded by the same species. This strategy should minimize the risk of later extracting pixels with mixed spectral signatures.

4 Methods

4.1 Workflow

FIGURE 4.1 illustrates the overall workflow of this thesis. This chapter explains in detail how each step contributes to the overall objective of creating a tree species classification map for my study area in the JPA. In section 4.2 it is shown how the HSI were prepared for later use with the most important step being the spatial co-registration of individual flight strips. The following section, 4.3, describes the process of preparing the labeled data as input for training, validating, and testing the classification network. In section 4.3 it is shown how to delineate ITCs from the LiDAR point cloud and how these were matched to the ground reference data. Further, computational methods are presented to illustrate how the sample data can be transformed to fit data formats needed for the CNN. Sections 4.4 and 4.5 demonstrate how the CNN is defined, trained, and evaluated with a proper accuracy assessment.

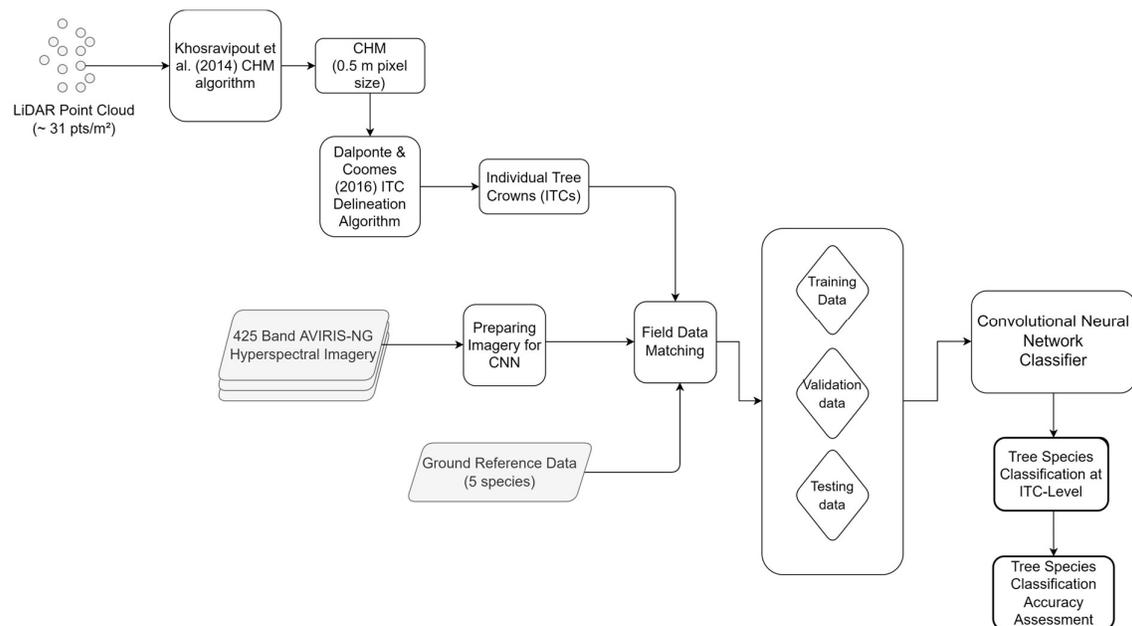


FIGURE 4.1. Workflow overview showing how different analytical methods and various software are used to process each data source to serve as input for the classifying algorithm – a CNN

4.2 Preprocessing Hyperspectral Imagery

The AVIRIS-NG HSI data used in this thesis was distributed in the ENVI [38] data format. It consists of a simple “flat binary file” and an associated ASCII header file containing important information on the nature of the imagery data [39]. This format allows the data to be stored in a rotated grid, meaning that the pixels are not aligned in north-south. That specification enables the observations to be stored according to the direction of the flight path and thus preserve the original values. The data used was rotated by 48 degrees. A positive rotation angle refers to the pixels being rotated to the east [40]. The imagery was georeferenced in the WGS84/UTM zone 32N (EPSG: 32632) coordinate reference system (CRS). To reduce the storage space required by the raw AVIRIS-NG reflectance data, all flight strips were clipped with a bounding box polygon of the study area which first had to be converted from the CH1903+/LV95 CRS. After clipping, the HSI was transformed to the CH1903+/LV95 CRS, as all other data sources used in this work were georeferenced in the Swiss CRS. All these operations were conducted in ENVI [38].



FIGURE 4.2. AVIRIS-NG imagery and SWISSIMAGE orthophoto before (left) and after (right) image co-registration.

Also, the transformation of the imagery to the Swiss CRS was needed to be able to co-register the data to a valid reference as offsets between the flight strips and compared to SWISSIMAGE were visible. SWISSIMAGE is the digital orthophoto mosaic of Switzerland provided by Swisstopo [41]. As geographical reference I used the SWISSIMAGE level 3 product with a pixel size of 0.1 m and a horizontal positional accuracy of 0.1 m. This most recent SWISSIMAGE product was acquired from 2017 to 2019 and published in 2020 [41]. I sampled the orthophoto down to a spatial resolution of 1 m to reduce the storage size of the product and ensure faster computation. Upon visual analysis, the offset between the flight strips was determined to be less than half a pixel, i.e., < 1 m. However, the visible offset to the reference image was larger. FIGURE 4.2 illustrates this offset between the SWISSIMAGE and the AVIRIS-NG HSI before (left) and after (right) image co-registration

conducted in ENVI version 5.6 [38] with the automated image registration workflow. The idea of image registration is to automatically align two images such that the corresponding pixels represent the same objects. This is done by locating and matching so-called tie points that correspond to the same feature in both images [42]. I defined tie points manually by identifying matching, non-movable features like road crossings or house edges in both images. The tie points were optimized until the Root Mean Square Error (RMSE) of the location of all tie points was smaller than half an AVIRIS-NG pixel, i.e., < 1 m. This is in accordance with the Positional Accuracy Standards for Digital Geospatial Data [43] defined by the American Society for Photogrammetry and Remote Sensing (ASPRS) in their Positional Accuracy Standards for Digital Geospatial Data stating that at a pixel size of 2 m the orthoimage RMSE in x- and y-direction must not be larger than one pixel.

4.3 CNN Input Data Preparation

4.3.1 CHM Computation

I used the LiDAR point cloud to compute a Canopy Height Model (CHM) which served as input for a delineation algorithm to identify Individual Tree Crowns (ITCs). The LiDAR data was processed in R version 4.0.4 [44] with the *lidR* package providing tools for the analysis of Airborne Laser Scanning (ALS) data [45], [46]. To compute a CHM from the point cloud, the data had to first be height normalized. For this a classified point cloud like the swissSURFACE3D product as first the height of all points classified as ground is set to zero to subsequently, interpolate the ground returns [47]. After height normalization the elevation of each point (i.e. the z-coordinate) is replaced with its height above the interpolated ground surface [47]. The new point cloud was used to generate a canopy height model which is a digital surface characterizing the vegetation height across the landscape [48]. The pixel size of the resulting raster was selected to be 0.5 m. The pixel size is a key parameter in CHM creation, especially for accurate derivation of individual tree attributes [49] such as ITCs. Pouliot et al. [50] suggest that the crown diameter to pixel size ratio has to be appropriate for defining crown shape [47]. I used a CHM pixel size of 0.5 m which seemed to be sufficient for accurately defining tree crown shape in the imagery with 2 m grid spacing. The point cloud used in this thesis has an average point density of ~ 31 points/m² which should be sufficient for generating a 0.5 m CHM, as other studies successfully have used datasets with lower or comparable point densities for similar tasks. The LiDAR data used by Ballanti et al. (4–8 points/m²) [51], Mäyrä et al. (10.2 points/m²) [3], and Versace et al. (21.5 points/m²) [52] all had lower point densities for their CHM of 0.5 m pixel size. FIGURE 4.3 illustrates the spatial distribution of point density which is highest where the flight strips of the ALS acquisition overlap.

For CHM generation two main categories of algorithms exist: (1) point-to-raster algorithms and (2) triangulation algorithms. Point-to-raster algorithms are more simplistic. They utilize a grid of a certain resolution and attribute the height of the highest pixel to each cell [45]. Point-to-raster

algorithms have been widely used in literature for example by Popescu [48], Dalponte & Coomes [53], Hyypä [54] and Hunter et al. [55]. A potential disadvantage of these algorithms however is that the resulting CHM can contain some empty pixels which have to be filled in post-processing using interpolation [45].

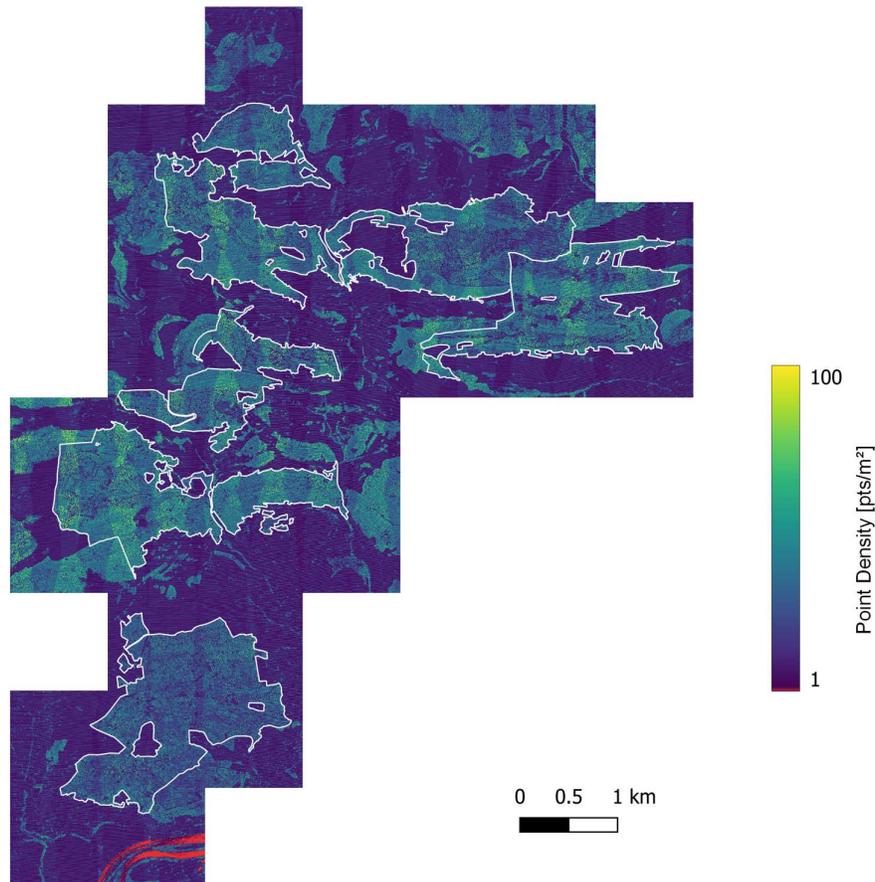


FIGURE 4.3. LiDAR point density of the study area. Pixels with zero LiDAR points are colored in red.

Empty pixels, or so-called ‘pits’ can appear whenever a laser beam is able to penetrate the tree crown deeply before it generates the first return [41]. These ‘pits’ can be avoided with more complex triangulation-based algorithms to interpolate first returns [40].

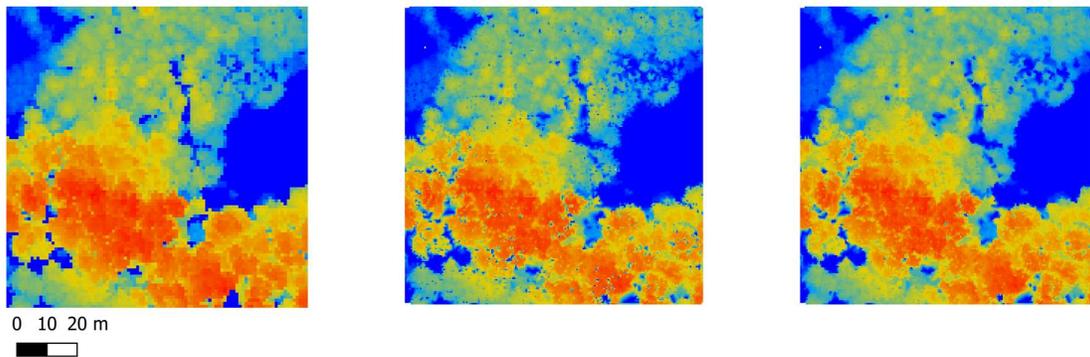


FIGURE 4.4. Comparison of three 100×100 m CHMs created from the same data with three different methods. Left: point-to-raster algorithm (1 m pixel size). Middle: Basic triangulation with rasterization of first returns (0.5 m pixel size). Right: Khosravipour et al. ‘pit-free’ algorithm (0.5 m pixel size). The forest subset is the same as described in FIGURE 3.3.

I used the ‘pit-free’ algorithm from Khosravipour et al. [47] which is implemented in the *lidR* package. This method uses a set of Delaunay triangulations applied sequentially using points with heights above a set of specified thresholds [45]. Khosravipour et al. [47] examined the suitability of their ‘pit-free’ CHM for single tree detection and report that it significantly outperforms conventionally computed CHMs. FIGURE 4.4. shows the comparison between three different methods for CHM computation, one of which is the used ‘pit-free’ algorithm.

4.3.2 ITC Delineation

I used the CHM as input for the delineation of ITCs. This was also done in R version 4.0.4 [44] with the *lidR* package [45], [46]. As Dalponte et al. [8] state, ITC delineation is extremely useful for ecological purposes surrounding trees as it allows researchers to analyze the forest in its primary element, the tree. Similar to CHM algorithms, ITC algorithms can also be split into two categories: (1) the ones using a CHM as input and (2) the ones operating on the whole point cloud [45]. Here, I used the algorithm from Dalponte & Coomes [53] which uses a CHM as input. Other authors using this algorithm for forest related research are Ying et al. [56], Versace et al. [52], Mauri [57], Torresan et al. [58], and Aben et al. [59]. This algorithm is an adapted version from the one Hyypä et al. [60] defined which came out among the best in a benchmarking study by Eysn et al. [61]. The approach first applies a smoothing filter on the CHM to reduce the local maxima which are afterwards located using a circular moving window. Each local maximum serves as an ‘initial region’ around which a tree crown can be located. Neighboring pixels are added to the region if their vertical distance to the local maximum is less than some user-defined fraction of the local maximum height, less than some user-defined maximum difference. The algorithm iterates this process for all neighboring cells included in the region until no more pixels can be added. To finalize the ITC delineation, a 2D convex hull is applied to each region and the height of the first return

within an ITC is extracted to report the height of the resulting ITC shapefiles [53]. Dalponte & Coomes incorporated this method into their R package *itcSegment* [62] but it is also included in *lidR* which is how I applied it. The difference between the two implementations is that in *lidR* the algorithm is written in C++ making it hundreds to millions times faster than the original version in *itcSegment* [46]. Another difference between the two implementations is that in *itcSegment* a 2D convex hull algorithm is applied to each initial growing region while in *lidR* the user can decide between a convex, concave, or bounding box algorithm to delineate the crown shapes. I opted for a concave hull algorithm as it can more precisely capture the geometric boundary of a dataset than a complex hull; this was also recognized by Park & Oh [63] who defined the hull algorithm implemented in *itcSegment*. Following Mauri [57], I buffered the CHM which was clipped to the study area by 10 m to preserve the shape of the trees exceeding the study area boundaries.

With a total of four adaptable parameters, the Dalponte & Coomes ITC algorithm offers a lot of flexibility for the user to achieve the best possible results for the given circumstances (e.g., data source, CHM resolution). After testing parameter combinations from Versace et al. [52], Mauri [57], Torresan et al. [58], and Aben et al. [59] I decided upon visual analysis to use the default parameters as shown and explained in TABLE 4.1. The resulting ITC polygons are shown in FIGURE 4.5 for a small subset of the study area.

TABLE 4.1. Parameter values used as input into the ITC delineation algorithm from Dalponte & Coomes [53] implemented in the *lidR* R package. Descriptions from the package’s documentation from Roussel & Auty [46].

Parameter	Value	Description
th_tree	2	Threshold [m] below which a pixel cannot be a tree.
th_seed	0.45	A pixel is added to a growing region if its height is greater than the tree height multiplied by this value.
th_cr	0.55	A pixel is added to a region if its height is greater than the current mean height of the growing region multiplied by this value.
max_cr	10	Maximum value of the crown diameter of a detected tree in pixels.

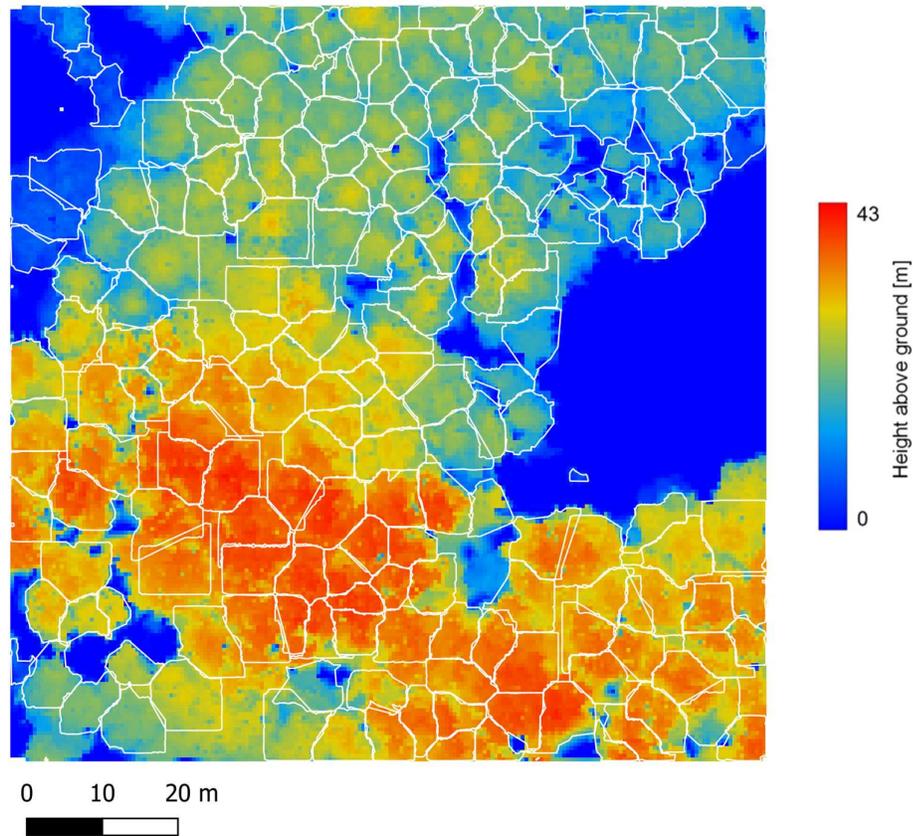


FIGURE 4.5. A 100×100 m subset of the CHM showing delineated ITCs using the Dalponte & Coomes [53] algorithm implemented in *lidR*. The forest subset is the same as described in FIGURE 3.3.

4.3.3 Field Data Matching

After delineating ITCs for the whole study area, I matched the sampled ground reference data (i.e., GPS points) to the tree crown polygons. This was done manually by visual interpretation in QGIS version 3.18 [64]. If an ITC polygon contained a field measured data point, the ITC was assigned a label according to the tree species the data point represents. However, some data points had to be discarded as they could not be clearly assigned to an ITC. This was the case when the data point was either on the border of an ITC or not included in a polygon at all. But still, points showing that behavior were not automatically discarded as I examined each case individually. To better identify the single trees, I used the CHM in addition to the AVIRIS-NG imagery (FIGURE 4.6). That means that to some points more than one ITC could be matched. Also, all selected ITCs smaller than 12 m^2 were discarded. That equals the area of three pixels in the imagery and of 48 pixels in the CHM. This was done to increase the chance of having ‘pure-pixels’ that only represent the crown of one tree. Fassnacht et al. [12] suggest that to ensure at least one ‘pure-pixel’ one should only include

ITCs of at least 3×3 pixels. When a tree crown is described by a square at least the size of 3×3 pixels, at least on pixel (i.e., the center pixel) is ‘pure’ meaning it only includes spectral information on the tree crown and not on any surroundings. But with this method I would have to discard too many tree crowns, as crowns are diverse in shape and not always cover 3×3 pixels. In the end I matched a total of 325 ITC polygons to the 331 sampled GPS points.

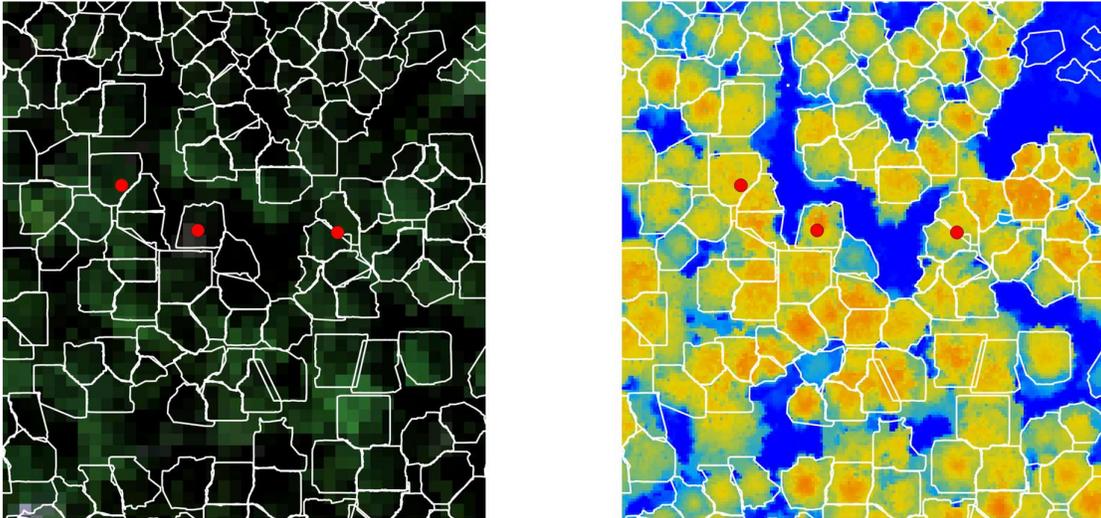


FIGURE 4.6. Small subset of the RGB image of the AVIRIS-NG data and the computed CHM used to match the ground reference data points to the delineated ITCs. The forest subset is the same as described in FIGURE 3.3.

For further analysis I split all ITCs into training, validation, and testing. I randomly assigned 20% of all ITCs per species to the test set ($n = 66$) and 20% of the remaining ITCs per species to the validation set ($n = 52$). All other 207 ITCs were used for training. The validation set will be used to optimize the model and the test data to assess the accuracy of the final model.

4.3.4 Image Patch Extraction

After matching the sampled GPS points to the HSI, the data must be processed to a form that can be utilized by the classifying model. The CNN requires image-like multidimensional input containing spectral as well as spatial information for each pixel [65], [66]. The idea is to extract square patches of a certain size from the imagery around the matched and labeled ITCs. The conceptual design of the image patch extraction is depicted in FIGURE 4.7. The size of the patches affect how much of the image the CNN can “see” when making a prediction [2]. First, each vector dataset containing the matched ITCs gets rasterized to a label raster where each pixel representing an ITC gets labeled according to the tree species and all other pixels are assigned a no-data value.

This label raster is afterwards used for extracting the image patches from the imagery. I extracted a 5×5 patch around each labeled pixel resulting in a total of 4,515 image patches for all trees. This was done for each flight strip individually as I did not create a hyperspectral mosaic to preserve the original reflectance values which would have been altered in the areas where the flight strips overlap. While all patches were in the same CRS (CH1903+/LV95) the pixel sizes vary between 1.9 and 2.1 m according to the flight strip. Patches being extracted from ITCs near the edges which contain no-data values were discarded.

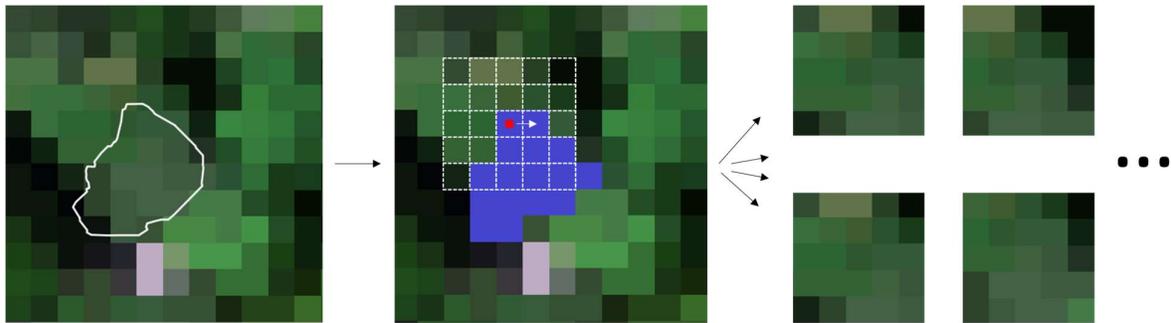


FIGURE 4.7. Conceptual workflow of image patch extraction. a) Polygon of a tree crown delineated from the LiDAR point cloud. b) Rasterized ITC shown in blue with the white grid illustration the extent of the image patch extracted around the upper-left pixel of the rasterized ITC. c) Four example image patches extracted from this crown. The upper-left patch is the one extracted around the example pixel in b).

I chose a patch size of 5×5 to give the CNN enough spatial context for learning how to predict the center pixel of each patch. However, just as reported by Mäyrä et al. [3], the patch size did not have a major impact on classification accuracies. Also, using 5×5 patches instead of 7×7 patches lowered the computation times substantially. Utilizing and learning spatial-spectral characteristics instead of spectral information alone is a big advantage of deep learning methods compared to more traditional machine learning concepts like SVM and RF which are heavily dependent on hand-crafted features [3].

After patch extraction I dropped noisy bands at the beginning and end of the spectrum and water absorption bands, as described in Wang et al. in [67] and [68]. Bands 1–7, 187–217, 281–330, and 408–425 were dropped. The remaining 319 bands were used for all subsequent analyses and covered the spectral regions 412.13–1303.67, 1463.95–1774.49, and 2029.93–2410.59 nm. I extracted the spectra of all labeled pixels (i.e., of all center pixels of the image patches) to visualize the average spectra per tree species in FIGURE 4.8. Due to large overlaps of the standard deviations of the average per-species reflectance these are not illustrated here. In the appendix the average reflectances with the standard deviations are illustrated for each species individually. Patch extraction was done in Python with the *NumPy* [69] and *Rasterio* [70] libraries.

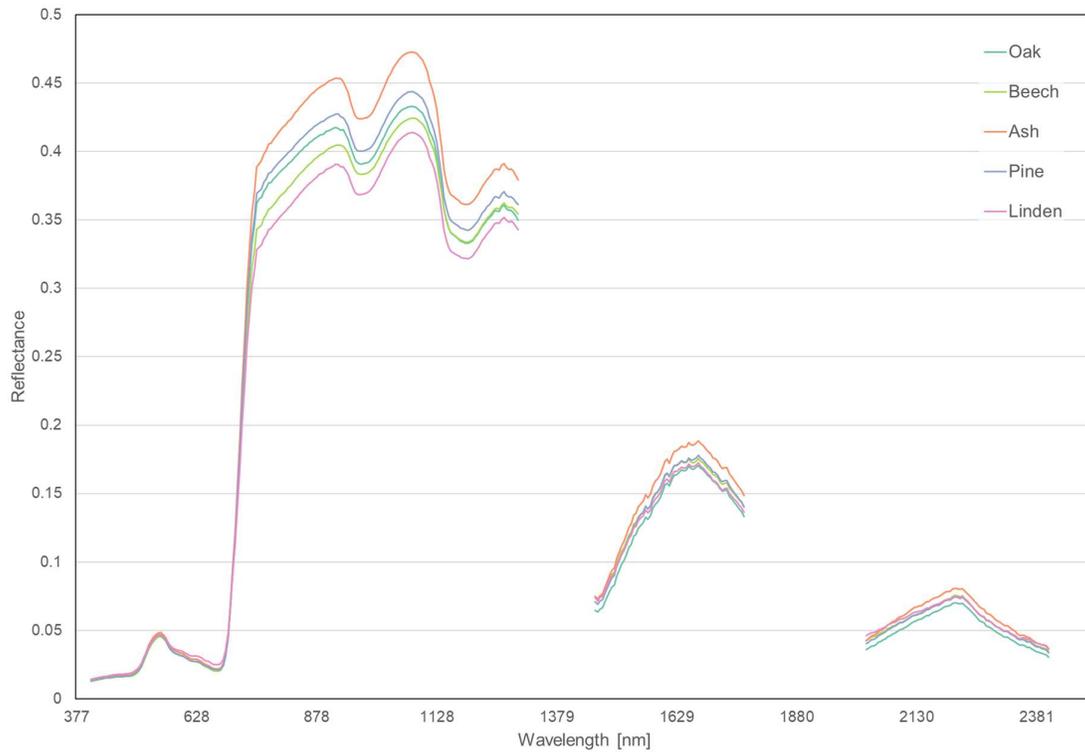


FIGURE 4.8. Average reflectances per target tree species after dropping noisy and water absorption bands.

4.3.5 Dimensionality Reduction

Due to its narrowband nature, it is possible to model, predict and map biophysical as well as biochemical components of vegetation with hyperspectral imagery like AVIRIS-NG [33].

However, for classification the high dimensionality of such data can pose a problem due to the Hughes phenomenon [51]. According to Hughes [71] increasing the number of spectral bands requires an exponential increase in the number of training samples for training a classification model. Thus, classification accuracies can be significantly reduced. The assumption is that more bands in the HSI may cause information redundancy due to neighboring bands being highly correlated [72]. Therefore, to address band collinearity and maximize the between-group variance (as desired in classification tasks) I applied dimensionality reduction [31], [73]. In classifying tasks dimensionality reduction techniques can simultaneously improve accuracies as well as computational efficiency [31]. I applied one of the most common and simple methods of dimensionality reduction for HSI classification which is to extract spectral features with a principal component analysis (PCA) [18], [74]–[76]. PCA for dimensionality reduction of HSI for species

discrimination was also applied in Fricker et al. [2], Mäyrä et al. [3], Patel et al. [19], and Prospere et al. [76].

PCA is a multivariate statistical method that transforms the original data into a set of orthogonal variables (principal components, PCs) by finding a projection matrix that maximizes the data variance in the reduced subspace [77], [78]. The first PC accounts for the maximum proportion of the variance, and subsequent components for the next highest proportion of the variance [77], [79]. I applied the PCA transformation in Python using the *Scikit-learn* module which integrates a wide range of state-of-the-art machine learning algorithms [80]. To apply a PCA transformation in *Scikit-learn* a user-defined number of components first get fit to the original data. Afterwards the learned transformation is applied to new data to transform it to lower-order dimensionality [81]. I first fit 32 PCA components to all patch center pixels of all species ($n = 4,515$) to subsequently apply the transformation to all complete image patches. This means that the transformation is only based on vegetated pixels containing spectral information on spectrally “pure” pixels that only contain information on one of the five species. Thus, in the dimensionality reduced data the variation between the different tree species should be increased.

4.3.6 Data Augmentation

One drawback of such classification tasks collection of ground reference data which is complicated, time consuming, and expensive [3], [18]. State-of-the-art RGB image recognition models have been pretrained with the ImageNet dataset [82] containing around 1.3 million annotated samples across 1,000 classes [3]. My training dataset however only consists of 2,830 labeled image patches and five classes. Having too few training samples can lead to the model overfitting and not being able to classify new data correctly [16]. That is why I applied data augmentation, a common practice to increase the number of training samples utilized in most studies applying CNNs to image data (e.g., [2], [3], [24], [83]–[85]). The basic idea of data augmentation is to generate additional training data from existing training samples via several operations that yield believable-looking images [81]. The goal is that the model will not see the same picture twice as this helps the model generalize better as it is exposed to more aspects of the data [16]. I applied image augmentation in two steps.

First I used a novel technique called MixUp proposed by Zhang et al. [86]. This technique replaces each image patch with a new patch which is a linear combination of two distinct patches that are of different species. More formally:

$$\begin{aligned}\tilde{x} &= \lambda x_i + (1 - \lambda)x_j, & \text{where } x_i, x_j \text{ are image patches} \\ \tilde{y} &= \lambda y_i + (1 - \lambda)y_j, & \text{where } y_i, y_j \text{ are one-hot label encodings}\end{aligned}$$

x_i and x_j are randomly selected image patches from our training data with y_i and y_j being the according one-hot labels [86]. \tilde{x} and \tilde{y} are the image patch and the corresponding label after MixUp.

One-hot means that the labels are not just a single number but rather a vector the length of classes indicating the class with a value between zero and one. Here an oak patch would have the label $[1, 0, 0, 0, 0]$. $\lambda \in [0, 1]$ is a value randomly sampled from a Beta distribution with shape parameters $\alpha = 0.4$ and $\beta = 0.5$. Thus, λ has a high possibility of being near zero or near one indicating weak mixing of the two patches, and low possibility of being near 0.5 indicating strong mixing of two patches. After a weak MixUp a patch which is a combination of oak and beech could have $[0.85, 0.15, 0.0, 0.0, 0.0]$ as its corresponding label. After MixUp with a large λ , however, the dominant label in the one-hot vector of a patch could change. This is why the number of patches per species can be different after applying the MixUp image augmentation technique (TABLE 4.2). MixUp has been shown to improve results for image classification as well as for speech recognition and also adds robustness in case of corrupt labels [3], [86].

TABLE 4.2. Distribution of training patches per tree species before and after MixUp augmentation.

Species	Number of training patches	
	Before MixUp	After MixUp
Oak	581	601
Beech	855	722
Ash	591	615
Pine	512	550
Linden	291	342
TOTAL	2830	2830

The second part of the image augmentation includes changing the brightness, rotating, and adding random noise. These transformations change the appearance of each patch and ultimately increase the available training data significantly, i.e., applying each transformation several times resulted in 96-folding the original training set and a total of 271,680 image patches. The number of patches per species before and after data augmentation is shown in TABLE 4.3.

TABLE 4.3. Number of labeled pixels before and after image augmentation in the training set.

Species	Number of training patches	
	Before Data Augmentation	After Data Augmentation
Oak	601	57,696
Beech	722	69,312
Ash	615	59,040
Pine	550	52,800
Linden	342	32,832
TOTAL	2830	271,680

4.3.7 Patch Equalization

As can be seen in TABLE 4.3, the number of training patches per tree species is uneven, which happens when reference samples are biased towards dominant trees [12]. I collected the most sample points for beech and the least number of points for linden. While this does not indicate anything definite on species dominance it might represent the abundance of the most dominant species. To eliminate bias of the model towards more represented species, I equalized the amount of training data per species following Dalponte et al. [8]. From each species' pool of training data, a random number of image patches was selected equal to the number of training patches of the least represented species. This means that for every species 32,832 image patches were randomly selected from the augmented training set, except for linden as for this species all of the available patches are used. Fassnacht et al. [5] also stated that classification accuracies can be biased by uneven distributions of samples per class and recommend mitigating this issue by adequately sampling underrepresented tree species.

4.4 CNN Model

4.4.1 Model Architecture

There are several ways in which CNNs can be used for hyperspectral image classification [3]. Even for the specific task of tree species classification various approaches have been taken in recent literature. While Fricker et al. [2] tested a fully convolutional network, Mäyrä et al. [3] opted for a

3D-CNN. Schiefer et al. [23] made use of the U-Net architecture originally introduced for the segmentation of biomedical images by Ronneberger et al. [87]. The same approach was taken and extended by Chen et al. [24] who developed and presented a so-called ResU-Net Model. I, however, present a rather simple model architecture which has proven to be the best option for the task at hand. In the following sections the model’s architecture and the used hyperparameters are described. While it is almost impossible to exploit all possible model definitions, I extensively tested a number of CNN configurations which lead to the selection of the parameters presented. The model was written in Python 3.10.2 [88] and implemented using Keras, a deep learning API (application programming interface) [89], and the TensorFlow backend which is an open source machine learning framework [90].

The chosen CNN architecture is rather simple as can be seen in its conceptual visualization in FIGURE 4.9. The architecture consisting of two two-dimensional convolutional (Conv) layers, a flatten and a fully connected layer, performed the best under the given circumstances.

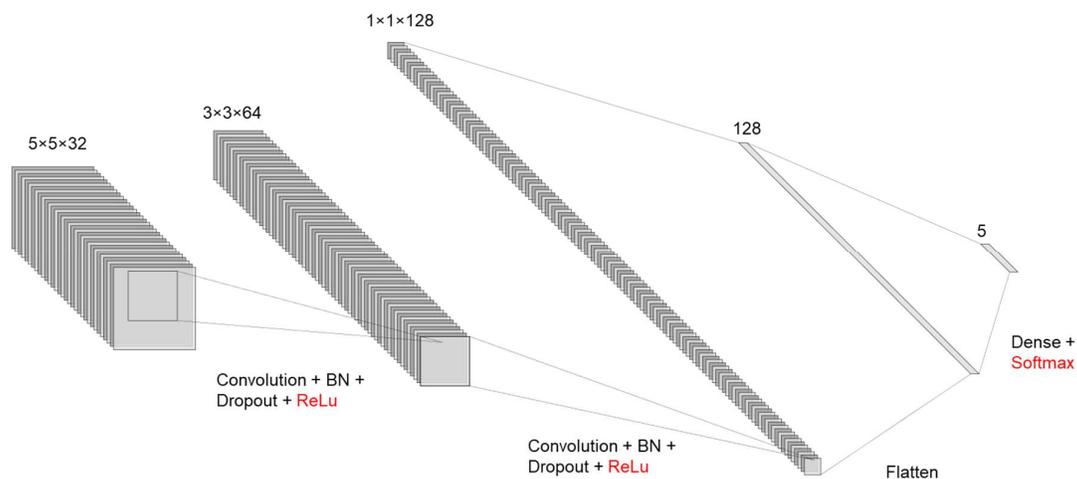


FIGURE 4.9. Architecture of the convolutional neural network used in this study.

The network was trained on patches measuring 5×5 pixels each consisting of 32 channels obtained after dimensionality reduction through PCA. The most significant components of a CNN are the Conv layers consisting of a collection of convolutional filters (so-called kernels). These filters are applied to the input data to generate the output feature map [91]. Mathematically, the kernel is a two-dimensional matrix, with values called kernel weights [91]. A fundamental step in DL is that these weights are initialized randomly and get adapted continuously during training based on a feedback signal. Every single one of those filters, or kernels, is referred to as a “neuron” as it connects to representations of the input data within the CNN with each other. The adjustment is done by an optimizer, which implements a backpropagation algorithm to minimize the loss function [16]. The loss function computes a distance score between the predictions of the network and the

true values of the targets capturing how well the network is performing [16]. That score is called the “loss”. For a Conv layer to produce an output, the kernel slides over the whole input image with a certain step-size (stride). Each repetition of this process produces a new feature image, which represents a single layer in the output stack of a Conv layer. The number of output layers is denoted as the number of filters and is defined by the user. In other words, every entry in the 3D output volume of a Conv layer can be interpreted as an output of a neuron that looks at a small region of the input image and shares parameters with all neighboring neurons since these values stem from the same kernel [92]. The activation function is responsible to determine the neuron output by computing the weighted summation of the neuron input [91]. After the Conv layers, I added a layer to flatten the 3D-patches to a 1D-vector, which serves as an input to the following fully connected (dense) layer. That last layer produces an array of five probability scores (summing to one) each denoting the probability that a patch belongs to one of the five tree species [16].

I chose a kernel size of 3×3 pixels and a stride of one for both Conv layers, with the first having 64 filters and the second having 128 filters. Generally, the convolutional layers cause a reduction of size of the input patches. Padding would add an appropriate number of rows and columns on each side of the input image making it possible to fit the center of the kernel on every pixel in the input patch and retaining the spatial dimensions of the input [16]. By not using padding, I reduced the image patches from 5×5 pixels to 3×3 pixels after the first Conv layer, and to 1×1 pixels after the second. Following Fricker et al. [2] and Mäyrä et al. [3] I did not use padding as it did not improve the results.

After both convolutional layers, I apply batch normalization (BN) and dropout. BN [93] normalizes the obtained feature by subtracting the mean μ and dividing it by the standard deviation σ of each layer [18], [91]. Applying BN before the activation function allows reaching a more independent and high-speed learning [18]. Dropout [94] increases model performance and robustness by forcing neurons to make more strict assumptions [18]. I used relatively high dropout values of 0.5 after the first and 0.7 after the second Conv layer. After BN and dropout, an activation function needs to be applied to map the input of a Conv layer [91]. I used the rectified linear unit (ReLU) [95] activation function, the most commonly used function in the CNN context, which converts all values of the input to positive numbers [91]. ReLU is defined as:

$$f(x)_{ReLU} = \max(0, x)$$

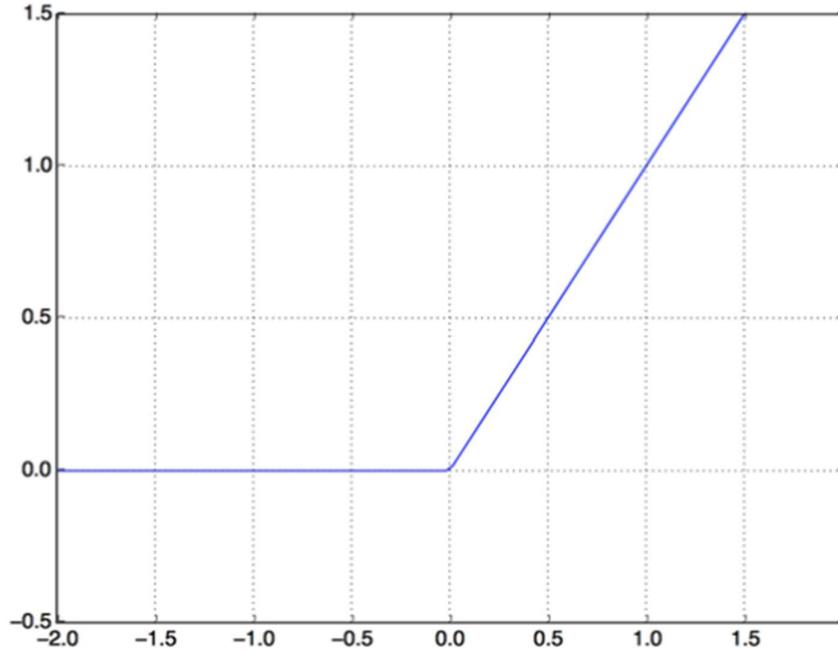


FIGURE 4.10. The rectified linear unit function [16].

By applying a $\max(\cdot)$ function between 0 and the input data x , ReLU equal or smaller than 0 to 0, and to x (i.e., estimated value) otherwise, as depicted in FIGURE 4.10 [18]. After a flattening layer which converts the output volumes of the Conv layers to a 1-D vector, the final probability scores are computed using a fully connected (dense) layer and the *softmax* activation function. In this dense layer, each neuron is connected to all other neurons in the previous layer [91]. To generate the probability scores, the dense layer employs the *softmax* activation function which is commonly applied in multi-class classifications and defined as

$$p_i = \frac{e^{a_i}}{\sum_{k=1}^N e_k^a}$$

where e^{a_i} represents the non-normalized output from the previous layer while N is the number of neurons in the output layer [91]. *Softmax* squeezes the data between 0 and 1 and divides the obtained outputs by their sum giving the probability of the input data belonging to a particular class (p_i) [18]. The CNN model is finalized by selecting an appropriate loss function. The most popular loss function for multi-class classification tasks is the categorical cross-entropy, defined as

$$\sum_{i=1}^n -y_i \log(p_i)$$

where n is the number of classes, p_i is the probability of class i , and y_i is 1 for the correct class and 0 for all other classes [3]. The model tries to minimize the loss function, which in this case is equivalent to maximizing the log-likelihood of the correct class assignment. This, however, can cause the model to overfit due to the loss function assigning 100% to a single class [3]. To avoid this, I applied the method suggested by Szegedy et al. [96] which was also applied for tree species classification by Mäyrä et al. [3]. The loss function called “label smoothing cross-entropy” penalizes the model for overconfidence by modifying the true target y and replacing them with y^* [3].

$$y^* = y_i(1 - \alpha) + \frac{\alpha}{n}$$

I used $\alpha=0.2$ which would replace $y=[0, 1, 0, 0, 0]$ with $y^*=[0.05, 0.8, 0.05, 0.05, 0.05]$. At least for RGB-image classification [96], [97], this method has been shown to improve model robustness [3]. I also applied L2 regularization, which is a popular method to prevent the model from overfitting [16], [18]. This method, which is also called weight decay in the context of neural networks inserts a penalty into the loss function proportional to the square value of the weight coefficients forcing the network to make compromises on its weights’ making it more general [16], [18]. As an optimizer for the model, I chose the Adaptive Moment Estimation (Adam) [98], which has shown excellent results and is thus the most widely used optimizer for deep learning networks [18]. Adam is more memory efficient and needs less computational power than alternatives and works by calculating an adaptive learning rate for each parameter in the model [91]. I used a learning rate of 0.001 with Adam, which is the proportion that weights are updated [91].

I developed code for defining and training my CNN model based on the scripts provided by Fricker et al. [2] and Mäyrä et al. [3].

4.4.2 Model Training

I defined a batch size of 64 and set the number of epochs for the model to 50. One epoch refers to one model training iteration over all training data [16]. However, the model did not need 50 epochs to be trained as I also applied early stopping [99]. Early stopping means that the training of the model is stopped and the results are saved once maximum validation accuracy is reached [16]. This technique can be very effective in training CNNs without significant overfitting [99]. I defined early stopping in a way that training is halted when the validation accuracy does not improve after five epochs.

4.4.3 Full Study Area Classifications

I applied the final model to the full AVIRIS-NG HSI data to compute a pixel-wise tree species classification map. To make a prediction for each pixel in the data, the model needs to be supplied with image patches of the same size the model was trained on. First, I transformed the full image data into the new coordinate system (calculated by PCA from the training data) by multiplying them with the PC loadings. Then, I predicted each pixel by extracting a 5×5 pixel patch around the pixel and feeding it to the CNN to make a prediction based on its' weights. However, the output scene was ultimately reduced in size by a border equal to the patch radius (i.e., 2 pixels). Causing this is the fact that border patches, which contain no-data values outside the area of interest, cannot be used by the CNN for prediction. I applied the model to each of the nine flightstrips separately, due to computational limitations, which means that the resulting predictions rasters have the same pixel size as the corresponding flightstrips (i.e., 1.9–2.1 m).

Finally, I combined the individual predictions per flight strip to one tree species classification map. I first resampled all segments to a 2 m pixel size and then aligned all rasters. Because neighboring flight strips overlap, I defined how the final prediction class was chosen if two different classes were predicted for the same pixel. This was done by assigning the label with the higher prediction probability score.

4.5 Accuracy Assessment

One common accuracy assessment approach is to present the classification of the testing data and the corresponding ground data summarized on a per-class basis in the form of a confusion matrix [100]. Based on this table metrics can be calculated, including the overall accuracy (OA), the user's (UA), and the producer's accuracy (PA). OA expresses the fraction of correctly classified samples, while UA is the number of true positives divided by all positive predictions per class and PA is the number of true positives divided by the actual number of samples per class [101]. Additionally, I report the F1-score, which is a commonly used statistic in DL studies and can be described as the harmonic mean between UA and PA. It is defined as

$$F1 = \frac{2 \times TP}{2 \times TP + FN + FP}$$

where TP are the true positives, FP the false positives, and FN the false negatives [102]. I report the F1-score for each species in addition to the average F1-score of all species and the weighted average F1-score which takes the number of samples per class into account weighing the species with a lot of samples more.

5 Results

5.1 Model Training

With early stopping, I defined that the model training is stopped when the classification accuracy of the validation set does not increase after five epochs. When this criterion is met, the model restores the weights from the best epoch (i.e., the epoch with the highest validation accuracy) and is being saved. Accuracy and loss of the training and validation sets during CNN training are depicted in FIGURE 5.1 showing that the first epoch produced the best results in terms of validation accuracy (accuracy = 0.55, loss = 1.41), while it kept improving in terms of training accuracy.

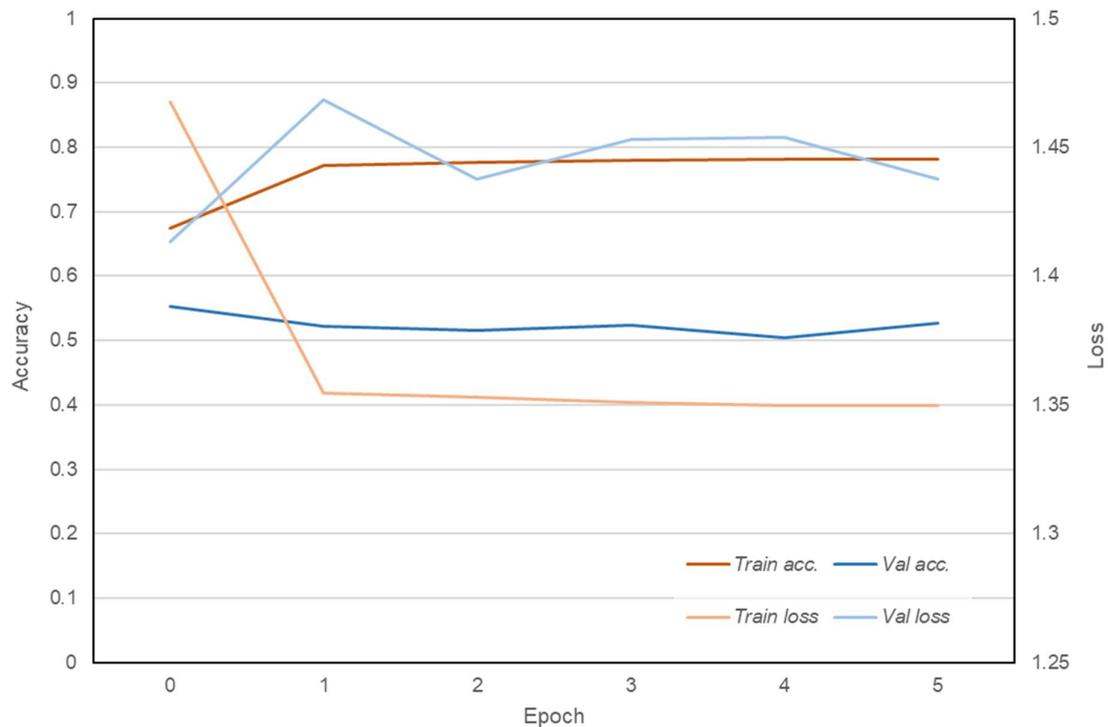


FIGURE 5.1. Accuracy and loss of the training and validation data sets during CNN training.

I chose the weights from the epoch with the best validation metrics and not the best training metrics, because the goal during prediction is classifying data the model has never seen before, similar to the validation data, which is why validation metrics are more informative about model performance.

This behavior of the model indicates that the most important representations are learned by the model after the first epoch with no major improvements in validation metrics afterwards.

5.2 Classification Results

FIGURE 5.2 shows the confusion matrix for the trained CNN comparing the samples of the test set with its' predicted classes. Here a 5×5 pixel patch, as explained in 4.3.4, represents one sample. As the model uses those patches to predict the center pixel this means that the classification metrics apply to all labeled pixels in the test set as a single sample.

The OA of the CNN is 70% with an average F1-score of 0.67 and a weighted average F1-score of 0.70 as stated in the classification report (TABLE 5.1). The weighted average F1-score exceeds the standard average because the species with the most samples in the test set, beech, has the highest accuracy metrics of all species with an UA of 75%, a PA of 86% and a F1-score of 0.80. This, however, is not caused by training the model with an unbalanced training set as I equalized the number of training samples per species before model training.

		Ground truth				
		<i>Oak</i>	<i>Beech</i>	<i>Ash</i>	<i>Pine</i>	<i>Linden</i>
Prediction	<i>Oak</i>	108	1	11	12	0
	<i>Beech</i>	48	231	22	8	0
	<i>Ash</i>	7	27	112	6	11
	<i>Pine</i>	21	8	16	107	11
	<i>Linden</i>	18	2	23	3	41

FIGURE 5.2. Confusion matrix for the trained CNN where rows indicate the predicted species and columns the ground truth.

The highest UA is achieved for oaks with 82%. However, the PA for oaks (53%) is the lowest of all species. This is caused by a large number of oak samples being wrongly classified while the likelihood that a sample that was classified as an oak was actually an oak was high. In absolute numbers, this is the model’s most common misclassification. Other major misclassifications are ash as beech, beech as ash, and oak or ash as pine or linden. The lowest score of any species is the UA of linden of 46% as the model does not seem to be able to learn the most important representations needed for a more robust classification of linden. More confusion matrices that show the UA and PA for each case are given in the appendix.

TABLE 5.1. Classification report for the trained CNN. Support indicates the number of image patches per class in the test set. Support indicates the number of test samples per class.

Species	UA	PA	F1	Support
Oak	0.82	0.53	0.65	202
Beech	0.75	0.86	0.80	269
Ash	0.69	0.61	0.65	184
Pine	0.66	0.79	0.72	136
Linden	0.47	0.65	0.55	63
Overall Accuracy	70%			
Average	0.68	0.69	0.67	
Weighted Average	0.72	0.70	0.70	

While the classification accuracies and metrics of the CNN are not exceptional, the model is able to outperform conventional machine learning methods like a random forest (RF) and a support-vector machine (SVM) classifier (FIGURE 5.3 & TABLE 5.2). The OAs of RF and SVM are 67% and 66% respectively, lower than the 70% OA of the CNN classifier. The average F1-score for RF is 0.61 with a weighted average of 0.65. For SVM, the average F1-score is 0.62 and the weighted average F1-score is the same as for RF. While the overall model metrics are lower, both RF and SVM exceeded the highest PAs for individual species compared to CNN. With 93% the RF PA for beech is the highest metric of all three models. The highest value for the SVM is also the PA for beech at 86%, which is equal to the CNN. In terms of per-species F1-scores all three models are mostly in tune with beech having the highest value followed by pine, ash, oak, and linden. In the CNN case, oak and ash have equal F1-scores. The highest misclassification rate is consistent for all three methods, which is oak classified as beech. However, RF misclassifies ash as beech even slightly more often. Also, RF and SVM perform worse than the CNN when trying to classify linden correctly. Especially RF seems to have trouble, with only 19 correctly classified linden samples, a

PA of 0.30, and a F1-score of 0.37. The linden UA, however, is similar with 49%. More confusion matrices that show the UA and PA for each case for RF and SVM are given in the appendix.

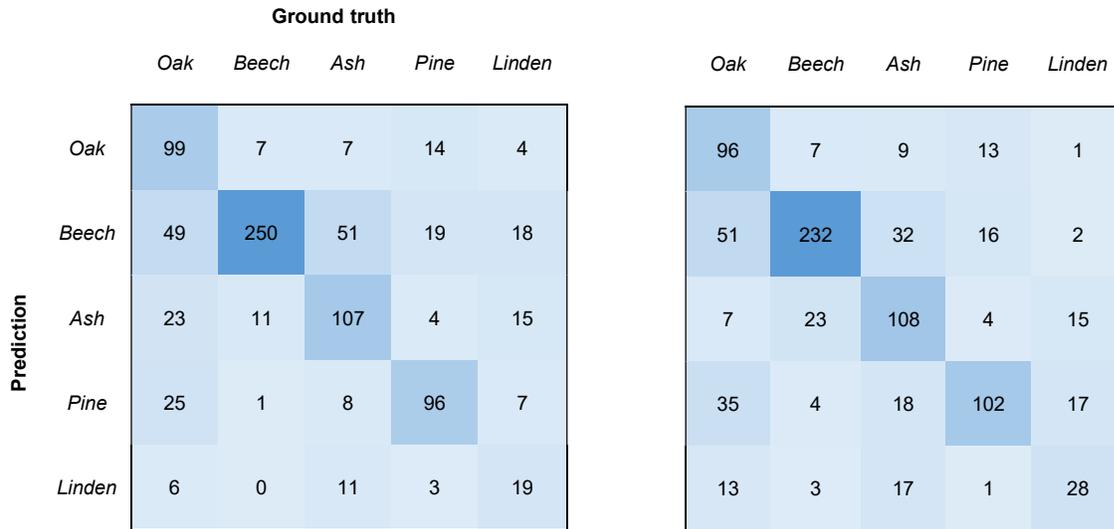


FIGURE 5.3. Confusion matrices for the comparison methods. Left: Random forest (RF). Right: Support-vector machine (SVM).

TABLE 5.2. Classification report for RF and SVM.

Species	RF			SVM		
	UA	PA	F1	UA	PA	F1
Oak	0.76	0.49	0.59	0.76	0.48	0.59
Beech	0.65	0.93	0.76	0.70	0.86	0.77
Ash	0.67	0.58	0.62	0.69	0.59	0.63
Pine	0.70	0.71	0.70	0.58	0.75	0.65
Linden	0.49	0.30	0.37	0.45	0.44	0.45
Overall Accuracy	67%			66%		
Average	0.65	0.60	0.61	0.64	0.62	0.62
Weighted Average	0.67	0.67	0.65	0.67	0.66	0.65

5.3 Full Study Area Classifications

The generated tree species classification map (FIGURE 5.4) includes all pixels with a probability score ≥ 0.4 and a NDVI ≥ 0.75 . The NDVI threshold is applied to only include vegetated pixels in the final classification map. The species with the highest abundance in the classification map is oak which is also the most dominant species. Pixels with a prediction confidence (i.e., probability score) of less than 40% were discarded as the forested area is not completely made up of the five species examined in this thesis. To each pixel the model assigns one probability score per species. This tells us the probability of a pixel belonging to a certain species. All probability scores summed up per pixel are 100%. For example, another prominent species in my study area is Maple (*Acer pseudoplatanus*). The average prediction confidence of all pixels in the final map (i.e., after confidence and NDVI thresholds) is 51.55%. While I would certainly desire this value to be higher it is acceptable. In 6.4 the full study classification map is further discussed.

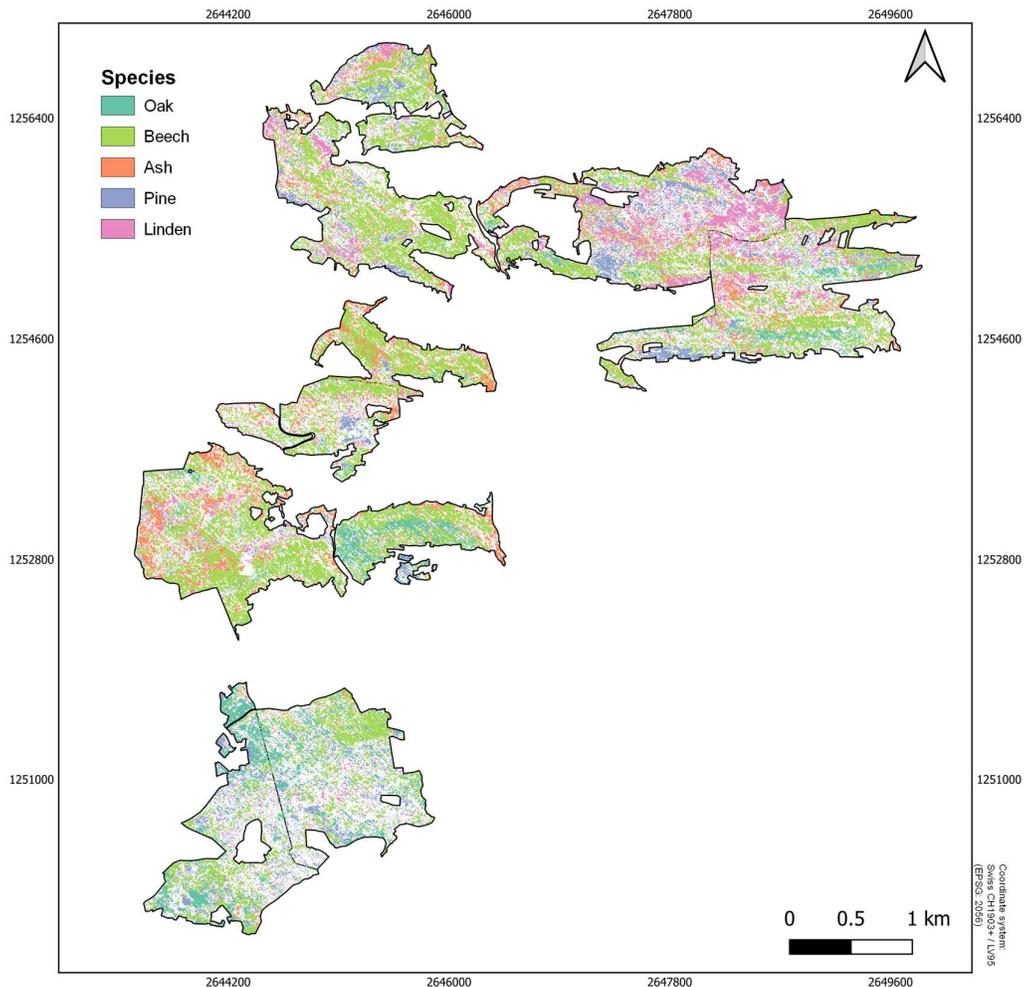


FIGURE 5.4. Full study area tree species classification map showing predictions with a probability score ≥ 0.4 and an NDVI ≥ 0.75 .

6 Discussion

In this thesis, I presented an experimental setup for the pixel-based classification of tree species from hyperspectral imagery with a Convolution Neural Network. This chapter comments on uncertainties, limitations, and possibilities in the experimental setup.

6.1 Ground Reference Data

For a high quality tree species classification, the ground reference data has to fulfill certain criteria and an accurate correspondence between field and remote sensing data needs to be established [2], [12]. Based on extensive literature review, Fassnacht et al. [12] defined criteria for the collection of ground reference data. Two of these criteria are that the data should be representative for the site under investigation and that the spatial scale of the data should match the problem under investigation. In the case of a tree species classification, this means that the collected ground samples must be representative for the true tree species distribution present. While I tried to fulfill these criteria as good as possible, the collected ground reference data is certainly not perfectly balanced. Rather, the number of samples by class matched my perceived relative abundance of species in the study area. There is, however, no easy way to verify this. Reference samples being biased towards dominant tree species could mean that reported accuracies for those species might be representative for a larger part of the wood volume but not necessarily for the larger fraction of individual trees [12]. And even though the model was ultimately trained using the same number of training samples per species, the fraction of artificial samples created with data augmentation was higher for species with less ground reference samples such as linden. However, the collection of a geographically more balanced and dispersed ground reference set was not possible due to the hilly topography of the study area and the dense forest structure. There are some spatial clusters of ground samples and some areas without any samples caused by the inaccessibility of the terrain. Both factors, the uneven distribution across species and across the geographical extent of the study area, are possibly introducing uncertainty into the final classification.

Another possible uncertainty factor is introduced in the matching of the field data to the HSI. Each of the data sources used in that process, the GPS field samples, the HSI, and the LiDAR point cloud have a certain positional inaccuracy. In case of the LiDAR data and the ground reference samples, that accuracy is known but it is not for the AVIRIS-NG HSI, at least not exactly. Generally, we assume a positional uncertainty of AVIRIS-NG HSI of \pm one pixel. To match the imagery to the collected ground samples and the delineated ITCs it first had to be co-registered to a reference image, the SWISSIMAGE orthophoto (4.2). This was critical to ensure the HSI geographically matched with the ITCs delineated from the LiDAR point cloud which was in sync with the

orthophoto. Accurate co-registration of different datasets, especially when working on a single tree level, is an important technical consideration in this field [12]. Also, manually co-registering images is prone to errors and thus positional uncertainties cannot be completely eliminated [42]. Positional inaccuracies and uncertainties can ultimately lead to incorrectly labelled samples used in the training and evaluation of the classification model. The effect of mislabeled pixels on the classification accuracy could be estimated by intentionally mislabeling a fraction of pixels and evaluating the classification accuracy of those as done by Clark et al. [103] and Laybros et al. [104]. In addition, mislabeled samples can be the result of inaccurate ITCs leading to mixed-canopy samples that are not labelled as belonging to several but to a single species. This could also be a reason for the large overlap of the standard deviations of the average per-species reflectance (see appendix). Still, my model, which incorporates the entire spectral information was able to differentiate species to a large degree.

An adequate amount of data for training a neural network is essential as it has an effect on the accuracy and efficiency of the CNN classifier [105]. Unfortunately, collecting a large number of high-quality training samples is not always easy due to limited time, access, or interpretability constraints [15].

6.2 Image Patch Extraction

In many remote sensing studies on tree classification spectra are only extracted from fully illuminated tree crowns. Especially in northern latitudes, shadowed pixels can pose a problem as the solar zenith angle is always quite large and, thus, shadowed pixels are present even in the best acquisition periods (i.e., during mid-summer) [7]. By eliminating the spectra of shaded pixels from the training data some of the intra-species variability of the spectral signatures can be reduced to increase classification accuracies [12]. The spectral properties of shadowed parts differ from sunlit segments of the same tree crown and thus are more often misclassified [51]. That is, however, also the reason I did not restrict training data selection to sunlit pixels. Since I intend to apply the model across the entire study area and thus to cope with any sort of illumination property, I trained the model using any sort of illumination. Also, limiting the data to sunlit pixels would reduce the already quite small amount of data I have. If an entire ITC is shadowed all the pixels within would have to be discarded and even if only a single pixel of an ITC is shadowed, it would not be possible to extract the 5×5 pixel image patches for most of the pixels in the ITC. This is because the model cannot deal with no-data values, which would have been assigned to shadowed pixels.

As for the selected patch size, I also assessed the performance of models using 3×3 and 7×7 pixel patches. The 3×3 pixel patches were able to classify the test patches with an overall accuracy (OA) of 68% and an average F1-score of 0.66. In the 7×7 pixel patch case, the model resulted in an OA of 65% and an average F1-score of 0.64. Both alternatives were thus not able to outperform the 5×5 patches, which classified the image patches of the test set with an OA of 70% and an

average F1-score of 0.67. However, the differences between the different patch sizes are marginal, especially in case of the average F1-score. The idea behind using larger image patches is that they are more likely to contain information from several trees, which can be helpful for classification purpose[3]. An example would be that pines tend to be surrounded by other pines [3]. Using image patches that are too large could, however, lead to the model using spatial and spectral information having no or only little explanatory power for classifying the patch's center pixel.

6.3 CNN Model

Defining the architecture of the CNN and finding the optimal values for the hyperparameters was a long and exhaustive process. I was focused on ultimately generating the best possible classification, meaning that I wanted to find the best model for the given circumstances which were the classification task itself, the quality and resolution of the input data, and the computational capacity. Most techniques and concepts used for the final model are primarily included to prevent overfitting. Overfitting occurs when the gap between the classification accuracy of training and testing data is large. This is ultimately caused by the model learning irrelevant patterns in the training data causing it not being able to generalize, which results in low validation and testing accuracies [16]. Dealing with overfitting means finding the right balance between optimization and generalization. Optimization refers to adjusting the model to get the best result possible, while generalization refers to how well the trained model performs on data it has never seen before [16]. The process of fighting overfitting by finding that balance is called regularization [16]. As Chollet & Allaire [16] state, the best solution to prevent overfitting is to get more training data. I followed that approach by augmenting the training data (see 4.3.6). Using the novel MixUp technique and traditional data augmenting transformations, including changing the brightness, rotating, and adding random noise, I increased the amount of training data 96-fold. Data augmentation not only increases the size of the training set, but also its variance [23]. According to Weinstein et al. [106], a larger training set with increased heterogeneity should further enhance the accuracy and generalization of the CNN. Another fundamental approach I took to prevent overfitting is the utilization of a rather simple model architecture to reduce the computational complexity of the CNN [107]. Simplifying the network and subsequently reducing the number of parameters leads to a lighter CNN model making the training and execution faster [18]. However, not many standards have been defined in that direction for hyperspectral classification problems [18]. In addition to data augmentation and model simplification, which are applied before the actual model execution, I also used regularization techniques applied during the training process. The used L2 regularization adds a penalty to the loss function to minimize the squared magnitude of the weights. This forces the model to reduce its' weights making it more general [18]. An alternative to the L2 regularization is the L1 regularization where the penalty is added proportionally to the absolute value of the weight coefficients (the L1 norm of the weights) [16]. L1 regularization enforces the identification of the most relevant features in a dataset, while L2 pursues a less aggressive regularization that is, however, more efficient in

computational terms [18]. Coding-wise, the L2 regularization is applied with the Conv layers as a so-called kernel regularizer. Applied as individual layers are batch normalization [93] and dropout [108]. Batch normalization tries to accelerate the training of deep neural networks via normalization steps that fix the means and variances of the layer inputs [93]. The original authors of this technique, Ioffe and Szegedy [93], state that batch normalization also has a beneficial effect in the gradient flow through the network by reducing the dependence of gradients on the scale of the parameters or their initial values. Ioffe and Szegedy [93] also mention that using batch normalization can eliminate the need for dropout as it provides similar regularization benefits. I did however use both techniques. The model produced the highest test accuracies when applying dropout directly after batch normalization and before the activation function. Dropout acts by randomly dropping a fraction of all neurons after each training epoch to equally distribute the feature selection power across all neurons [91]. This forces the model to learn different independent features [91], prevents complex co-adaptions, and is thus also a regularization method against overfitting [94]. However, Park and Kwak [109] report that traditional dropout adds random single-pixel noise, which can make it ineffective in 2-D and 3-D models.

In section 5.1 I described the model's behavior during training. The model achieved the highest validation accuracy and the lowest validation loss during the first epoch. Scientific literature on such behavior is sparse as it is most likely often viewed as non-successful. I hypothesize that the non-convergence of accuracy and loss towards a maximum or minimum respectively could be a hint that a fraction of the image patches get classified randomly. As that pattern also appeared with other model architectures and during hyperparameter tuning it could be a sign that the quality of the training data is sub-optimal. As described in 6.1 and 6.2 an array of computational steps is needed to extract the image patches used for model training, validation, and testing. At each step a certain factor of uncertainty is introduced. Depending on how large that factor is this could ultimately lead to misclassified patches or patches that include irrelevant information preventing the model from learning important representations. Still, whilst the classification accuracies are not quite at the desired level, they are acceptable and show that the model is able to learn the most important spectral differences among tree species. These representations are probably learned in the first epoch as the gap between train and validation accuracy widens afterwards, which can be a sign for potential overfitting.

6.4 Classification Accuracy

The adequacy of the training data that spectrally define the classes is of fundamental importance to a classification and a major determinant of the classification accuracy [110], [111]. As mentioned in 6.3 the produced classification accuracies could be limited by the quality and quantity of the hyperspectral input data. To assess that, I conducted a partial least squares discriminant analysis (PLS-DA). Partial least squares (PLS) [112] is a family of statistical methods specifically designed for the large degree of auto-correlation inherent in hyperspectral data [113]. PLS-DA is an effective

technique that addresses the complexities associated with high dimensional data by executing dimension reduction and classification simultaneously [113]. The underlying principle is that a few eigenvectors of spectral matrices are created, which are used to produce scores that explain both the variance of the hyperspectral data as well as the correlation with the dependent variable (i.e., tree species) [113]–[116].

Using R version 4.0.4 [44] and the *caret* [117] package, designed for classification and regression training, I classified all original image patches with 319 spectral bands by applying a PLS-DA. I only included the image patches of the training set as I wanted to assess their suitability for CNN training. The model was executed 50 times to generate a probability plot as seen in FIGURE 6.1. The results show that for each species the probability of a patch being classified correctly is larger than the patch being classified as any other species. However, the chance of correct classification is about only 40% for all species.

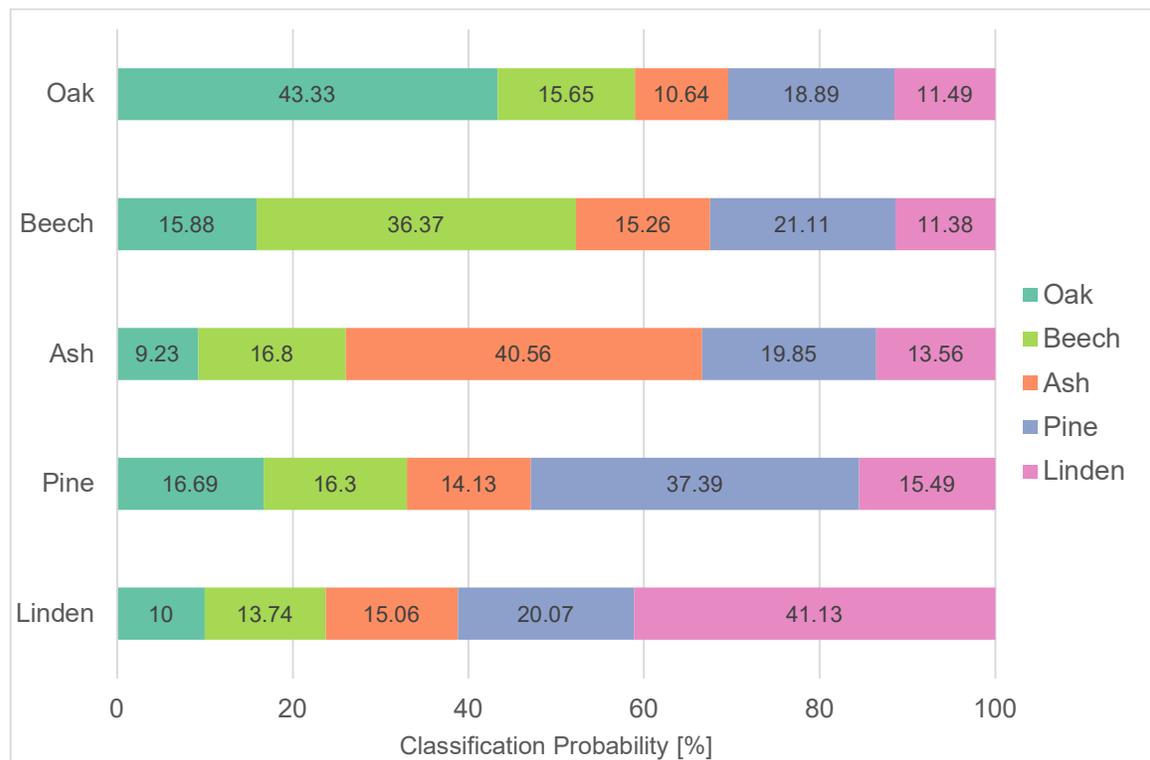


FIGURE 6.1. Probability plot showing the classification probability per tree species after 50 PLS-DA simulations.

The CNN classification model resulted in better classification accuracies than this simple PLS-DA. This could indicate that the different pre-processing steps like dimensionality reduction and data augmentation, as well as the model itself maximize the extraction of information from the training data. The PLS-DA also exhibits a different behavior in terms of misclassifications as compared to

the CNN model. The major misclassifications of the CNN (FIGURE 5.2) like oak as beech, beech as ash, or ash as linden are not visible in the PLS-DA results. In the PLS-DA case the misclassifications are almost equally distributed across all species without any clear patterns. Most misclassified patches are classified as pine, while the species with the least number of misclassified patches is linden. Especially for linden, this is a harsh contrast to the CNN classification results where linden has the lowest user's accuracy (UA) of all species. UA is the number of true positives divided by all positive predictions per class. One reason for the low UA of linden (47%) could be that most of the trees I sampled were relatively small compared to other trees. The average tree height per species, assessed from the LiDAR point cloud, of all sampled trees is summarized in TABLE 6.1. With an average height of 17.82 m the sampled linden are more than six meters smaller than the next taller species. That circumstance is also illustrated in the mean tree crown size per species which is the smallest for linden as well with 41.15 m². As I only found a few really large linden during the ground reference data acquisition this circumstance is no surprise. Most linden I observed (in the accessible terrain) were located in the understory and could thus not be sampled. However, some of the rather small linden were still top of the canopy and thus sampled.

TABLE 6.1. Mean tree height and mean tree crown area per species of all sampled trees with the according standard deviations.

Species	Mean tree height [m]	Tree height standard deviation [m]	Mean tree crown area [m²]	Tree crown area standard deviation [m²]
Oak	25.80	5.80	45.77	14.50
Beech	27.69	6.17	48.48	15.19
Ash	24.28	7.36	45.26	15.92
Pine	24.39	6.41	50.55	15.30
Linden	17.82	9.70	41.15	17.31

The lower average height could lead to a number of uncertainties in the pre-processing and ultimately in the training of the CNN. In hyperspectral data, small trees have a higher possibility of being in shadowed areas, compared to large trees, which could lead to higher noise in the spectral signature of those pixels [7]. In addition, the smaller tree crown diameter of these trees poses multiple problems. Small crowns can be lost in the background signal if image pixels are larger than the crowns [6]. Also it is harder to automatically detect smaller tree crowns as more spatial detail is needed [4]. All these factors could lead to a high uncertainty regarding species ID and crown

location in the extracted image patches for smaller trees, which is most likely the most detrimental for the classification of linden.

Official information on forests from the Canton of Aargau can be used to further evaluate the classification results for the study area. The plant sociological map [118] shows the main forest types present in my study area. The classification of 10 categories was summarized based on the more detailed forest classes defined by Ellenberg & Klötzli [119].

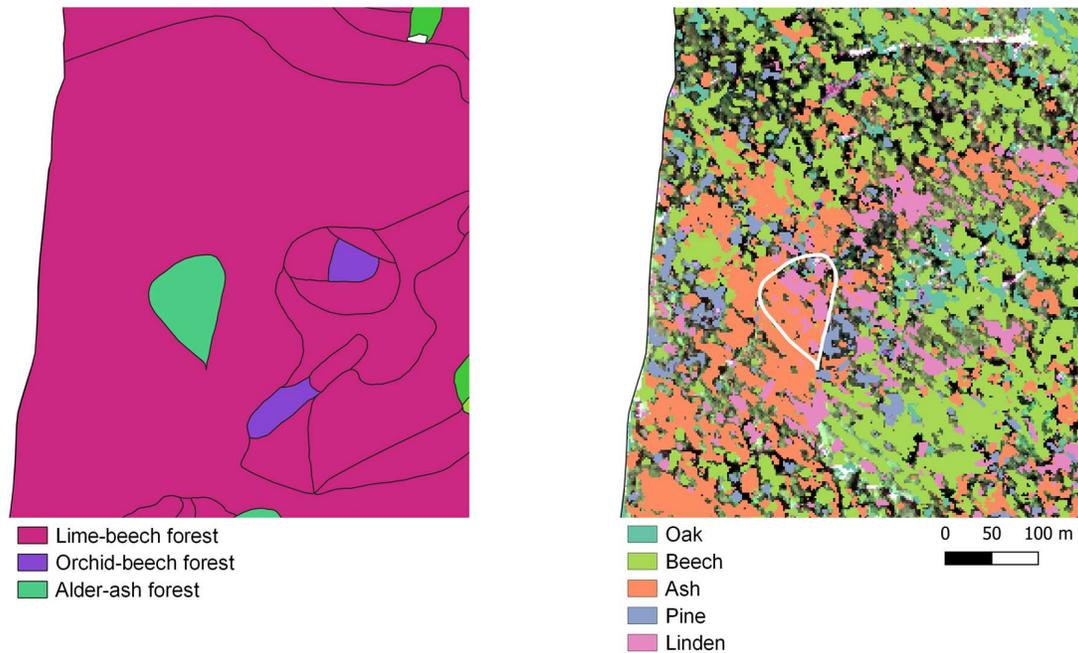


FIGURE 6.2. Subset of the study area showing the forest types (left) and the CNN predictions (right). This subset is located at the eastern border of the study area.

FIGURE 6.2 shows a subset of the study area for comparison of the plant sociological map of the Canton of Aargau with my CNN predictions. This is an example for a good model performance as a high ash abundance is predicted in the location of the reported Alder-ash Forest area. Another larger-scale example for good prediction is the southernmost forest area of my study area. As can be seen in FIGURE 6.3 that area is an oak Forest Reserve. In the full study area classification map (FIGURE 5.4) that area exhibits the highest relative abundance of oak.

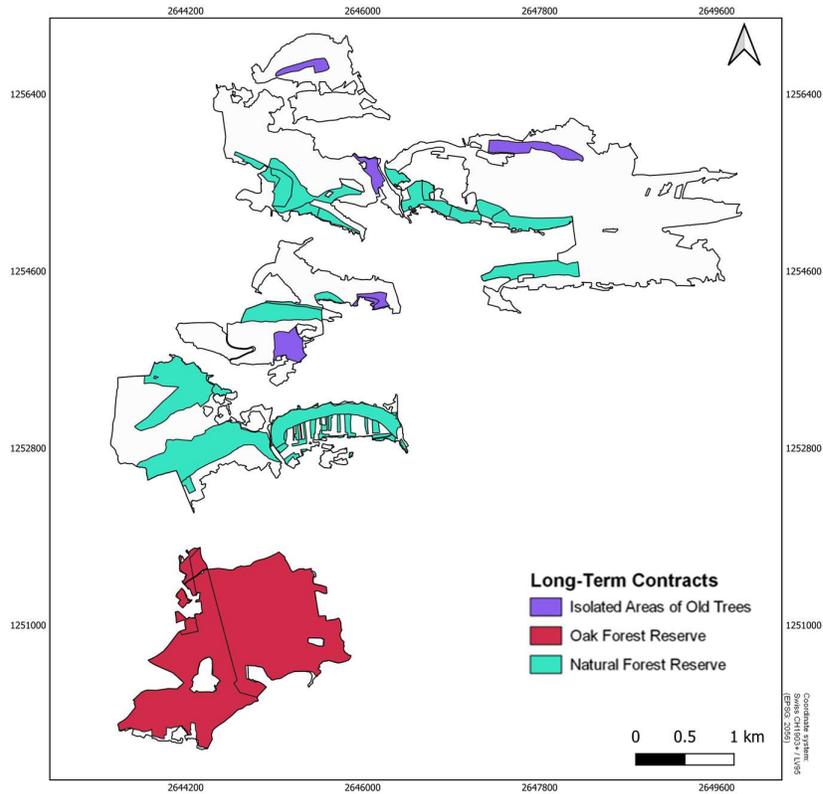


FIGURE 6.3. Long-Term contracts for the preservation of forest in the Canton of Aargau [120].

Such observations show that some ecological circumstances can be, at least to some degree, reproduced by the CNN model. However, more work is needed to exploit the full potential of hyperspectral data for tree species classification and to improve the robustness of the model and the classification accuracies. Especially the optimization of the pre-processing steps, which include image co-registration and field data matching, could have a significant positive effect on the prediction accuracies. Also, deep learning and CNNs offers an almost infinite number of different solutions for such problems. It might as well be that other, more complex, network architectures (e.g. ResU-Net [24], 3D-CNNs [3], Parallel CNNs [19]) are better in exploiting the full information content of hyperspectral imagery from sensors like AVIRIS-NG. Another improvement in classification results could perhaps come from more and better distributed sample data.

7 Conclusion

This thesis shows that a simple CNN architecture can classify tree species from hyperspectral data with an overall accuracy (OA) of 70% and an average F1-score of 0.67. Training, validation, and testing data was derived from ground reference data, a LiDAR point cloud, and AVIRIS-NG imagery to classify five major tree species (oak, beech, ash, pine, linden) for a 10,4 km² forest area in the Jurapark Aargau (JPA). One of the main challenges in training a robust prediction network is to sample and generate an adequate amount of high-quality data. At each processing step between the ground reference data collection and the image patch extraction uncertainties are being introduced. Limiting that level of uncertainty is immensely important in creating training data with large explanatory power. This allows the CNN model to learn the representations needed to discriminate among many tree species. In this thesis, the quantity as well as the quality of the training data was certainly limited. Especially the image co-registration of the AVIRIS-NG flight strips and a reference orthophoto seem to have a large potential for introducing a level of (positional) uncertainty. However, a simple CNN model with two convolutional layers managed to outperform established machine learning methods. A support-vector machine classifier achieved an OA of 66% and an average F1-score of 0.62, and a random forest classifier an OA of 67% and an average F1-score of 0.61. The CNN utilized different regularization methods, including L2 regularization, batch normalization, and dropout to increase its robustness and to prevent overfitting. In addition, MixUp and data augmentation techniques were used to increase the number of training samples synthetically.

This thesis shines a light on the potential of deep learning in the context of high resolution hyperspectral remote sensing imagery. Further work should be focused on the development of more powerful and robust CNNs that can be reliably used in all sorts of species discrimination tasks (e.g., crop discrimination). The goals of this thesis were defined in accordance with the needs of the JPA, highlighting the benefits of cooperation between researchers and forest practitioners.

References

- [1] B. B. N. Strassburg *et al.*, “Global priority areas for ecosystem restoration,” *Nature*, vol. 586, no. 7831, pp. 724–729, 2020, doi: 10.1038/s41586-020-2784-9.
- [2] G. A. Fricker, J. D. Ventura, J. A. Wolf, M. P. North, F. W. Davis, and J. Franklin, “A convolutional neural network classifier identifies tree species in mixed-conifer forest from hyperspectral imagery,” *Remote Sens.*, vol. 11, no. 19, 2019, doi: 10.3390/rs11192326.
- [3] J. Mäyrä *et al.*, “Tree species classification from airborne hyperspectral and LiDAR data using 3D convolutional neural networks,” *Remote Sens. Environ.*, vol. 256, no. February, 2021, doi: 10.1016/j.rse.2021.112322.
- [4] M. Dalponte, L. Bruzzone, and D. Gianelle, “Tree species classification in the Southern Alps based on the fusion of very high geometrical resolution multispectral/hyperspectral images and LiDAR data,” *Remote Sens. Environ.*, vol. 123, pp. 258–270, 2012, doi: 10.1016/j.rse.2012.03.013.
- [5] F. E. Fassnacht *et al.*, “Comparison of Feature Reduction Algorithms for Classifying Tree Species With Hyperspectral Data on Three Central European Test Sites,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 7, no. 6, pp. 2547–2561, 2014.
- [6] M. Dalponte, H. O. Ørka, T. Gobakken, D. Gianelle, and E. Næsset, “Tree species classification in boreal forests with hyperspectral data,” *IEEE Trans. Geosci. Remote Sens.*, vol. 51, no. 5, pp. 2632–2645, 2013, doi: 10.1109/TGRS.2012.2216272.
- [7] M. Dalponte, H. O. Ørka, L. T. Ene, T. Gobakken, and E. Næsset, “Tree crown delineation and tree species classification in boreal forests using hyperspectral and ALS data,” *Remote Sens. Environ.*, vol. 140, pp. 306–317, 2014, doi: 10.1016/j.rse.2013.09.006.
- [8] M. Dalponte, L. Frizzera, and D. Gianelle, “Individual tree crown delineation and tree species classification with hyperspectral and LiDAR data,” *PeerJ*, vol. 2019, no. 1, 2019, doi: 10.7717/peerj.6227.
- [9] T. Matsuki, N. Yokoya, and A. Iwasaki, “Hyperspectral tree species classification of japanese complex mixed forest with the aid of lidar data,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 8, no. 5, pp. 2177–2187, 2015, doi: 10.1109/WHISPERS.2014.8077510.
- [10] R. Piironen, J. Heiskanen, E. Maeda, A. Viinikka, and P. Pellikka, “Classification of tree species in a diverse African Agroforestry landscape using imaging spectroscopy and laser scanning,” *Remote Sens.*, vol. 9, no. 9, pp. 1–20, 2017, doi: 10.3390/rs9090875.
- [11] Y. Wu and X. Zhang, “Object-Based tree species classification using airborne hyperspectral images and LiDAR data,” *Forests*, vol. 11, no. 1, 2020, doi: 10.3390/f11010032.
- [12] F. E. Fassnacht *et al.*, “Review of studies on tree species classification from remotely sensed data,” *Remote Sens. Environ.*, vol. 186, pp. 64–87, 2016, doi: 10.1016/j.rse.2016.08.013.
- [13] M. Hasmadi, H. Z. Pakhriazad, and M. F. Shahrin, “Evaluating supervised and

- unsupervised techniques for land cover mapping using remote sensing data,” *Geogr. Malaysian J. Soc. Sp.*, vol. 5, no. 1, pp. 1–10, 2009.
- [14] K. Jusoff, M. H. Ismail, and N. H. M. Ali, “Spectral separability of tropical forest tree species using airborne hyperspectral imager,” *J. Environ. Sci. Eng.*, vol. 3, no. 1, p. 37, 2009.
- [15] A. E. Maxwell, T. A. Warner, and F. Fang, “Implementation of machine-learning classification in remote sensing: An applied review,” *Int. J. Remote Sens.*, vol. 39, no. 9, pp. 2784–2817, 2018, doi: 10.1080/01431161.2018.1433343.
- [16] F. Chollet and J. J. Allaire, *Deep learning with R*. Shelter Island, NY: Manning Publications, 2019.
- [17] A. Plaza, J. Plaza, A. Paz, and S. Sánchez, “Parallel Hyperspectral Image and Signal Processing,” *IEEE Signal Process. Mag.*, vol. 28, no. 3, pp. 119–126, 2011.
- [18] M. E. Paoletti, J. M. Haut, J. Plaza, and A. Plaza, “Deep learning classifiers for hyperspectral imaging: A review,” *ISPRS J. Photogramm. Remote Sens.*, vol. 158, no. November, pp. 279–317, 2019, doi: 10.1016/j.isprsjprs.2019.09.006.
- [19] H. Patel, N. Bhagia, T. Vyas, B. Bhattacharya, and K. Dave, “Crop Identification and Discrimination Using AVIRIS-NG Hyperspectral Data Based on Deep Learning Techniques,” *Int. Geosci. Remote Sens. Symp.*, pp. 3728–3731, 2019, doi: 10.1109/IGARSS.2019.8897897.
- [20] J. Gu *et al.*, “Recent advances in convolutional neural networks,” *Pattern Recognit.*, vol. 77, pp. 354–377, 2018, doi: 10.1016/j.patcog.2017.10.013.
- [21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Adv. Neural Inf. Process. Syst.*, vol. 25, pp. 1097–1105, 2012, doi: 10.1201/9781420010749.
- [22] C. Sothe *et al.*, “Evaluating a convolutional neural network for feature extraction and tree species classification using uav-hyperspectral images,” *ISPRS Ann. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. 5, no. 3, pp. 193–199, 2020, doi: 10.5194/isprs-Annals-V-3-2020-193-2020.
- [23] F. Schiefer *et al.*, “Mapping forest tree species in high resolution UAV-based RGB-imagery by means of convolutional neural networks,” *ISPRS J. Photogramm. Remote Sens.*, vol. 170, no. July, pp. 205–215, 2020, doi: 10.1016/j.isprsjprs.2020.10.015.
- [24] C. Chen, L. Jing, H. Li, and Y. Tang, “A new individual tree species classification method based on the resu-net model,” *Forests*, vol. 12, no. 9, 2021, doi: 10.3390/f12091202.
- [25] “Wald und innovativer Holzbau im Jurapark Aargau.” 2022, [Online]. Available: <https://jurapark-aargau.ch/wald-innovativer-holzbau.html>.
- [26] AGIS, “Aargauisches Geografisches Informationssystem.” [Online]. Available: <http://www.geoportal.ag.ch>.
- [27] V. Boccia *et al.*, “NASA-ESA Cooperation on the SBG and CHIME Hyperspectral Satellite Missions : a roadmap for the joint Working Group on Cal / Val activities,” *EGU Gen. Assem. 2021*, no. online, 19–30 Apr 2021, 2022.
- [28] J. W. Chapman *et al.*, “Spectral and radiometric calibration of the Next Generation Airborne Visible Infrared Spectrometer (AVIRIS-NG),” *Remote Sens.*, vol. 11, no. 18, 2019, doi: 10.3390/rs11182129.
- [29] Anna K. Schweiger, “Ökologie trifft Fernerkundung - Bildspektrometerflug im Jurapark

- Aargau,” pp. 1–3, 2021, [Online]. Available: <https://www.valpar.ch>.
- [30] OpenStreetMap contributors, “Planet dump retrieved from <https://planet.osm.org> .” 2017.
- [31] K. L. Roth, D. A. Roberts, P. E. Dennison, M. Alonzo, S. H. Peterson, and M. Beland, “Differentiating plant species within and across diverse ecosystems with imaging spectroscopy,” *Remote Sens. Environ.*, vol. 167, pp. 135–151, 2015, doi: 10.1016/j.rse.2015.05.007.
- [32] M. Alonzo, K. Roth, and D. Roberts, “Identifying santa barbara’s urban tree species from AVIRIS imagery using canonical discriminant analysis,” *Remote Sens. Lett.*, vol. 4, no. 5, pp. 513–521, 2013, doi: 10.1080/2150704X.2013.764027.
- [33] S. Ahmad, A. C. Pandey, A. Kumar, and N. V. Lele, “Potential of hyperspectral AVIRIS-NG data for vegetation characterization, species spectral separability, and mapping,” *Appl. Geomatics*, vol. 13, no. 3, pp. 361–372, 2021, doi: 10.1007/s12518-021-00355-6.
- [34] Bundesamt für Landestopografie (Swisstopo), “swissSURFACE3D: Die klassifizierte Punktwolke der Schweiz,” no. April, 2019, [Online]. Available: https://shop.swisstopo.admin.ch/de/products/height_models/surface3d.
- [35] Bundesamt für Landestopografie (Swisstopo), “swissSurface3D.” 2020, [Online]. Available: https://www.swisstopo.admin.ch/de/geodata/height/surface3d.html#341_1554992541029.
- [36] Y. Feng and J. Wang, “GPS RTK Performance Characteristics and Analysis,” *J. Glob. Position. Syst.*, vol. 7, no. 1, pp. 1–8, 2008, doi: 10.5081/jgps.7.1.1.
- [37] R. Engler *et al.*, “Combining ensemble modeling and remote sensing for mapping individual tree species at high spatial resolution,” *For. Ecol. Manage.*, vol. 310, pp. 64–73, 2013, doi: 10.1016/j.foreco.2013.07.059.
- [38] Exelis Visual Information Solutions, “ENVI version 5.6.” Boulder, Colorado, 2021.
- [39] Research Systems, “ENVI User’s Guide,” *Research Systems*. 2001, [Online]. Available: <https://lost-contact.mit.edu/afs/enea.it/software/rsi/envi/docs/userguid.pdf>.
- [40] H. Xie, N. Hicks, G. Randy Keller, H. Huang, and V. Kreinovich, “An IDL/ENVI implementation of the FFT-based algorithm for automatic image registration,” *Comput. Geosci.*, vol. 29, no. 8, pp. 1045–1055, 2003, doi: 10.1016/S0098-3004(03)00104-3.
- [41] Bundesamt für Landestopografie (Swisstopo), “Swissimage - Das digitale Farbphotomosaik der Schweiz,” vol. January, 2022.
- [42] X. Jin, “ENVI Automated Image Registration Solutions.” pp. 1–26, 2018, [Online]. Available: https://www.harrisgeospatial.com/portals/0/pdfs/ENVI_Image_Registration_Whitepaper.pdf.
- [43] American Society for Photogrammetry and Remote Sensing (ASPRS), “ASPRS Positional Accuracy Standards for Digital Geospatial Data,” *Photogramm. Eng. Remote Sens.*, vol. 81, no. 3, pp. 1–26, 2015, doi: 10.14358/pers.81.3.a1-a26.
- [44] R Core Team, “R: A language and environment for statistical computing.” Vienna, Austria, Vienna, Austria, 2021, [Online]. Available: <https://www.r-project.org/>.
- [45] J. R. Roussel *et al.*, “lidR: An R package for analysis of Airborne Laser Scanning (ALS) data,” *Remote Sens. Environ.*, vol. 251, no. September, p. 112061, 2020, doi: 10.1016/j.rse.2020.112061.

- [46] J.-R. Roussel and D. Auty, “Airborne LiDAR Data Manipulation and Visualization for Forestry Applications. R package version 3.2.2.” 2021, [Online]. Available: <https://cran.r-project.org/package=lidR>.
- [47] A. Khosravipour, A. K. Skidmore, M. Isenburg, T. Wang, and Y. A. Hussin, “Generating pit-free canopy height models from airborne lidar,” *Photogramm. Eng. Remote Sensing*, vol. 80, no. 9, pp. 863–872, 2014, doi: 10.14358/PERS.80.9.863.
- [48] S. C. Popescu, “Estimating biomass of individual pine trees using airborne lidar,” *Biomass and Bioenergy*, vol. 31, pp. 646–655, 2007, doi: 10.1016/j.biombioe.2007.06.022.
- [49] Q. Chen, D. Baldocchi, P. Gong, and M. Kelly, “Isolating Individual Trees in a Savanna Woodland using Small Footprint LIDAR data,” *Photogramm. Eng. Remote Sensing*, vol. 72, pp. 923–932, 2006, doi: 10.14358/PERS.72.8.923.
- [50] D. A. Pouliot, D. J. King, F. W. Bell, and D. G. Pitt, “Automated tree crown detection and delineation in high-resolution digital camera imagery of coniferous forest regeneration,” *Remote Sens. Environ.*, vol. 82, no. 2–3, pp. 322–334, 2002.
- [51] L. Ballanti, L. Blesius, E. Hines, and B. Kruse, “Tree species classification using hyperspectral imagery: A comparison of two classifiers,” *Remote Sens.*, vol. 8, no. 6, p. 445, 2016, doi: 10.3390/rs8060445.
- [52] S. Versace, D. Gianelle, L. Frizzera, R. Tognetti, V. Garf, and M. Dalponte, “Prediction of Competition Indices in a Norway Spruce and Silver Fir-Dominated Forest Using Lidar Data,” *Remote Sens.*, vol. 11, no. 23, p. 2734, 2019.
- [53] M. Dalponte and D. A. Coomes, “Tree-centric mapping of forest carbon density from airborne laser scanning and hyperspectral data,” *Methods Ecol. Evol.*, vol. 7, no. 10, pp. 1236–1245, Oct. 2016, doi: 10.1111/2041-210X.12575.
- [54] J. Hyypä, “Detecting and estimating attributes for single trees using laser scanner,” *Photogramm J Finl.*, vol. 16, pp. 27–42, 1999.
- [55] M. O. Hunter, M. Keller, D. Victoria, and D. C. Morton, “Tree height and tropical forest biomass estimation,” *Biogeosciences*, vol. 10, no. 12, pp. 8385–8399, 2013.
- [56] W. Ying, T. Dong, Z. Ding, and X. Zhang, *PointCNN-Based Individual Tree Detection Using LiDAR Point Clouds*. Springer International Publishing, 2021.
- [57] L. Mauri, “Assessment of tree segmentation algorithms based on UAV LiDAR data for Above Ground Biomass estimation,” Wageningen University and Research Centre, 2020.
- [58] C. Torresan, F. Carotenuto, U. Chiavetta, F. Miglietta, A. Zaldei, and B. Gioli, “Individual tree crown segmentation in two-layered dense mixed forests from uav lidar data,” *Drones*, vol. 4, no. 2, pp. 1–20, 2020, doi: 10.3390/drones4020010.
- [59] J. Aben, J. Signer, J. Heiskanen, P. Pellikka, and J. M. J. Travis, “What you see is where you go: visibility influences movement decisions of a forest bird navigating a three-dimensional-structured matrix: What you see is where you go,” *Biol. Lett.*, vol. 17, no. 1, 2021, doi: 10.1098/rsbl.2020.0478rsbl20200478.
- [60] J. Hyypä, O. Kelle, M. Lehtikoinen, and M. Inkinen, “A segmentation-based method to retrieve stem volume estimates from 3-D tree height models produced by laser scanners,” *IEEE Trans. Geosci. Remote Sens.*, vol. 39, no. 5, pp. 969–975, 2001.
- [61] L. Eysn *et al.*, “A benchmark of lidar-based single tree detection methods using heterogeneous forest data from the alpine space,” *Forests*, vol. 6, no. 5, pp. 1721–1747,

2015.

- [62] M. Dalponte, “Package ‘itcSegment,’” *R*, pp. 1–12, 2018.
- [63] J. S. Park and S. J. Oh, “A new concave hull algorithm and concaveness measure for n-dimensional datasets,” *J. Inf. Sci. Eng.*, vol. 29, no. 2, pp. 379–392, 2013.
- [64] QGIS Association, “QGIS version 3.18.” 2021, [Online]. Available: <http://www.qgis.org>.
- [65] K. Makantasis, K. Karantzalos, A. Doulamis, and N. Doulamis, “Deep supervised learning for hyperspectral data classification through convolutional neural networks,” *2015 IEEE Int. Geosci. Remote Sens. Symp.*, pp. 4959–4962, 2015, doi: 10.1109/IGARSS.2015.7326945.
- [66] A. Sharma, X. Liu, X. Yang, and D. Shi, “A patch-based convolutional neural network for remote sensing image classification,” *Neural Networks*, vol. 95, pp. 19–28, 2017, doi: 10.1016/j.neunet.2017.07.017.
- [67] Z. Wang *et al.*, “Mapping foliar functional traits and their uncertainties across three years in a grassland experiment,” *Remote Sens. Environ.*, vol. 221, no. July 2018, pp. 405–416, 2019, doi: 10.1016/j.rse.2018.11.016.
- [68] Z. Wang *et al.*, “Foliar functional traits from imaging spectroscopy across biomes in eastern North America,” *New Phytol.*, vol. 228, no. 2, pp. 494–511, 2020, doi: 10.1111/nph.16711.
- [69] C. R. Harris *et al.*, “Array programming with NumPy,” *Nature*, vol. 585, no. 7825, pp. 357–362, Sep. 2020, doi: 10.1038/s41586-020-2649-2.
- [70] S. Gillies, “Rasterio: geospatial raster I/O for Python programmers.” [Online]. Available: <https://github.com/rasterio/rasterio>.
- [71] G. Hughes, “On the Mean Accuracy of Statistical Pattern Recognizers,” *IEEE Trans. Inf. Theory*, vol. 14, no. 1, pp. 55–63, 1968, doi: 10.1109/TIT.1969.1054314.
- [72] Z. Dabiri and S. Lang, “Comparison of independent component analysis, principal component analysis, and minimum noise fraction transformation for tree species classification using APEX hyperspectral imagery,” *ISPRS Int. J. Geo-Information*, vol. 7, no. 12, p. 488, 2018, doi: 10.3390/ijgi7120488.
- [73] C. H. Yeh and C. H. Spiegelman, “Partial least squares and classification and regression trees,” *Chemom. Intell. Lab. Syst.*, vol. 22, no. 1, pp. 17–23, 1994, doi: [https://doi.org/10.1016/0169-7439\(93\)E0045-6](https://doi.org/10.1016/0169-7439(93)E0045-6).
- [74] I. T. Jolliffe, “Springer series in statistics,” *Princ. Compon. Anal.*, vol. 29, 2002.
- [75] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemom. Intell. Lab. Syst.*, vol. 2, no. 1, pp. 37–52, 1987, doi: [https://doi.org/10.1016/0169-7439\(87\)80084-9](https://doi.org/10.1016/0169-7439(87)80084-9).
- [76] D. Fernandez, C. Gonzalez, D. Mozos, and S. Lopez, “FPGA implementation of the principal component analysis algorithm for dimensionality reduction of hyperspectral images,” *J. Real-Time Image Proc.*, pp. 1–12, 2016.
- [77] K. Prospere, K. McLaren, and B. Wilson, “Plant species discrimination in a tropical wetland using in situ hyperspectral data,” *Remote Sens.*, vol. 6, no. 9, pp. 8494–8523, 2014, doi: 10.3390/rs6098494.
- [78] A. Sellami, M. Farah, I. Riadh Farah, and B. Solaiman, “Hyperspectral imagery classification based on semi-supervised 3-D deep neural network and adaptive band selection,” *Expert Syst. Appl.*, vol. 129, pp. 246–259, 2019, doi:

10.1016/j.eswa.2019.04.006.

- [79] G. Zhao and A. L. Maclean, “A comparison of canonical discriminant analysis and principal component analysis for spectral transformation,” *Photogramm. Eng. Remote Sensing*, vol. 66, no. 7, pp. 841–847, 2000.
- [80] F. Pedregosa *et al.*, “Scikit-learn: Machine Learning in Python,” *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, 2011.
- [81] O. Kramer, “Scikit-Learn,” in *Machine Learning for Evolution Strategies*, 20th ed., Cham: Springer, 2016, pp. 45–53.
- [82] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 248–255, doi: 10.1109/CVPR.2009.5206848.
- [83] H. Li, B. Hu, Q. Li, L. Jing, and O. N. Mj, “CNN-Based Tree Species Classification using Airborne LiDAR Data and High-Resolution Satellite Image,” in *IGARSS 2020-2020 IEEE International Geoscience and Remote Sensing Symposium*, 2020, pp. 2679–2682.
- [84] F. H. Wagner *et al.*, “Using the U-net convolutional network to map forest types and disturbance in the Atlantic rainforest with very high resolution images,” *Remote Sens. Ecol. Conserv.*, vol. 5, no. 4, pp. 360–375, 2019, doi: 10.1002/rse2.111.
- [85] K. Cao and X. Zhang, “An Improved Res-UNet Model for Tree Species Classification Using Airborne High-Resolution Images,” *Remote Sens.*, vol. 12, no. 7, p. 1128, 2020.
- [86] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, “MixUp: Beyond empirical risk minimization,” *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.*, pp. 1–13, 2018.
- [87] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” in *International Conference on Medical image computing and computer-assisted intervention computing and computer-assisted intervention*, Cham: Springer, 2015, pp. 234–241.
- [88] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.
- [89] F. Chollet, “Keras.” 2015, [Online]. Available: <https://keras.io>.
- [90] M. Abadi, A. Agarwal, and P. Barham, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems.” 2015, [Online]. Available: <https://www.tensorflow.org/>.
- [91] L. Alzubaidi *et al.*, *Review of deep learning: concepts, CNN architectures, challenges, applications, future directions*, vol. 8, no. 1. Springer International Publishing, 2021.
- [92] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” *Proc. 2017 Int. Conf. Eng. Technol. ICET 2017*, vol. 2018-Janua, pp. 1–6, 2018, doi: 10.1109/ICEngTechnol.2017.8308186.
- [93] S. Ioffe and C. Szegedy, “Batch Normalization : Accelerating Deep Network Training by Reducing Internal Covariate Shift,” vol. 37, 2015.
- [94] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, “Improving neural networks by preventing co-adaptation of feature detectors,” *arXiv Prepr. arXiv1207.0580*, 2012.
- [95] V. Nair and G. E. Hinton, “Rectified Linear Units Improve Restricted Boltzmann Machines,” *Icml*, no. 3, 2010.
- [96] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, “Rethinking the Inception

- Architecture for Computer Vision,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016, vol. December, pp. 2818–2826, doi: 10.1109/CVPR.2016.308.
- [97] R. Müller, S. Kornblith, and G. Hinton, “When Does Label Smoothing Help?,” *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.
- [98] Z. Zhang, “Improved Adam Optimizer for Deep Neural Networks,” *2018 IEEE/ACM 26th Int. Symp. Qual. Serv.*, pp. 18–19, 2018.
- [99] R. Caruana, S. Lawrence, and L. Giles, “Overfitting in Neural Nets : Backpropagation , Conjugate Gradient , and Early Stopping,” *Adv. Neural Inf. Process. Syst.*, vol. 13, 2000.
- [100] A. E. Maxwell, T. A. Warner, and L. A. Guillén, “Accuracy Assessment in Convolutional Neural Network-Based Deep Learning Remote Sensing Studies—Part 1: Literature Review,” *Remote Sens.*, vol. 13, no. 13, p. 2450, 2021, doi: 10.3390/rs13132591.
- [101] J. Cohen, “A coefficient of agreement for nominal scales,” *Educ. Psychol. Meas.*, vol. 20, no. 1, pp. 37–46, 1960.
- [102] A. E. Maxwell, T. A. Warner, and L. A. Guillén, “Accuracy Assessment in Convolutional Neural Network-Based Deep Learning Remote Sensing Studies—part 2: Recommendations and best practices,” *Remote Sens.*, vol. 13, no. 13, p. 2591, 2021, doi: 10.3390/rs13132591.
- [103] M. L. Clark, D. A. Roberts, and D. B. Clark, “Hyperspectral discrimination of tropical rain forest tree species at leaf to crown scales,” *Remote Sens. Environ.*, vol. 96, no. 3–4, pp. 375–398, 2005, doi: 10.1016/j.rse.2005.03.009.
- [104] A. Laybros *et al.*, “Quantitative airborne inventories in dense tropical forest using imaging spectroscopy,” *Remote Sens.*, vol. 12, no. 10, pp. 1–27, 2020, doi: 10.3390/rs12101577.
- [105] X. Zhuang, B. A. Engel, D. F. Lozano-Garcia, R. N. Fernández, and C. J. Johannsen, “Optimization of training data required for Neuro-classification,” *Int. J. Remote Sens.*, vol. 15, no. 16, pp. 3271–3277, 1994, doi: 10.1080/01431169408954326.
- [106] B. G. Weinstein, S. Marconi, S. A. Bohlman, A. Zare, and E. P. White, “Ecological Informatics Cross-site learning in deep learning RGB tree crown detection,” *Ecol. Inform.*, vol. 56, p. 101061, 2020, doi: 10.1016/j.ecoinf.2020.101061.
- [107] P. Maji and R. Mullins, “On the reduction of computational complexity of deep convolutional neural networks,” *Entropy*, vol. 20, no. 4, 2018, doi: 10.3390/e20040305.
- [108] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting,” *2Journal Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014, doi: 10.1016/0370-2693(93)90272-J.
- [109] S. Park and N. Kwak, “Analysis on the dropout effect in convolutional neural networks,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 10112 LNCS, pp. 189–204, 2017, doi: 10.1007/978-3-319-54184-6_12.
- [110] T. Lillesand, R. W. Kiefer, and J. Chipman, *Remote sensing and image interpretation*. John Wiley and Sons Ltd, 2015.
- [111] G. M. Foody and A. Mathur, “The use of small training sets containing mixed pixels for accurate hard image classification: Training on mixed spectral responses for classification by a SVM,” *Remote Sens. Environ.*, vol. 103, no. 2, pp. 179–189, 2006,

doi: 10.1016/j.rse.2006.04.001.

- [112] S. Wold, M. Sjöström, and L. Eriksson, “PLS-regression: A Basic Tool of Chemometrics,” *Chemom. Intell. Lab. Syst.*, vol. 58, pp. 109–130, Oct. 2001, doi: 10.1016/S0169-7439(01)00155-1.
- [113] K. Y. Peerbhay, O. Mutanga, and R. Ismail, “Commercial tree species discrimination using airborne AISA Eagle hyperspectral imagery and partial least squares discriminant analysis (PLS-DA) in KwaZulu-Natal, South Africa,” *ISPRS J. Photogramm. Remote Sens.*, vol. 79, pp. 19–28, 2013, doi: 10.1016/j.isprsjprs.2013.01.013.
- [114] P. T. Wolter, P. A. Townsend, B. R. Sturtevant, and C. C. Kingdon, “Remote Sensing of Environment Remote sensing of the distribution and abundance of host species for spruce budworm in Northern Minnesota and Ontario,” *Remote Sens. Environ.*, vol. 112, pp. 3971–3982, 2008, doi: 10.1016/j.rse.2008.07.005.
- [115] L. Li, S. L. Ustin, S. Member, and D. Riaño, “Using Genetic Algorithm Partial Least Squares (GA-PLS) Modeling,” *IEEE Geosci. Remote Sens. Lett.*, vol. 4, no. 2, pp. 216–220, 2007.
- [116] L. Li, Y. Cheng, S. Ustin, X. Hu, and D. Rian, “Retrieval of vegetation equivalent water thickness from reflectance using genetic algorithm (GA) -partial least squares (PLS) regression,” *Adv. Sp. Res.*, vol. 41, pp. 1755–1763, 2008, doi: 10.1016/j.asr.2008.02.015.
- [117] M. Kuhn *et al.*, “Classification and Regression Training. R package version 6.0-91.,” 2022.
- [118] Aargauisches Geografisches Informationssystem (AGIS), “Pflanzensoziologische Kartierung.” 2010, [Online]. Available: https://www.ag.ch/de/verwaltung/dfr/geoportal/geodaten_agis/geodatenliste/geodatenliste.jsp?rewriteRemoteUrl=%2Fgeoportal%2Fdownloads%2Fdetails%2FAGIS.aw_stao%3Fsearchcontext%3Dsozio.
- [119] H. Ellenberg and F. Klötzli, “Waldgesellschaften und Waldstandorte der Schweiz,” *Mitt. Eidgenöss. Forsch.anst. Wald Schnee Landschaft.*, vol. 48, no. 4, pp. 587–930, 1972.
- [120] Aargauisches Geografisches Informationssystem (AGIS), “NPW: Langfristige Verträge.” 2021, [Online]. Available: https://www.ag.ch/de/verwaltung/dfr/geoportal/geodaten/geodatenliste?rewriteRemoteUrl=/details/AGIS.aw_lvab?searchcontext%3Dlangfri.

A Appendix – Additional Figures

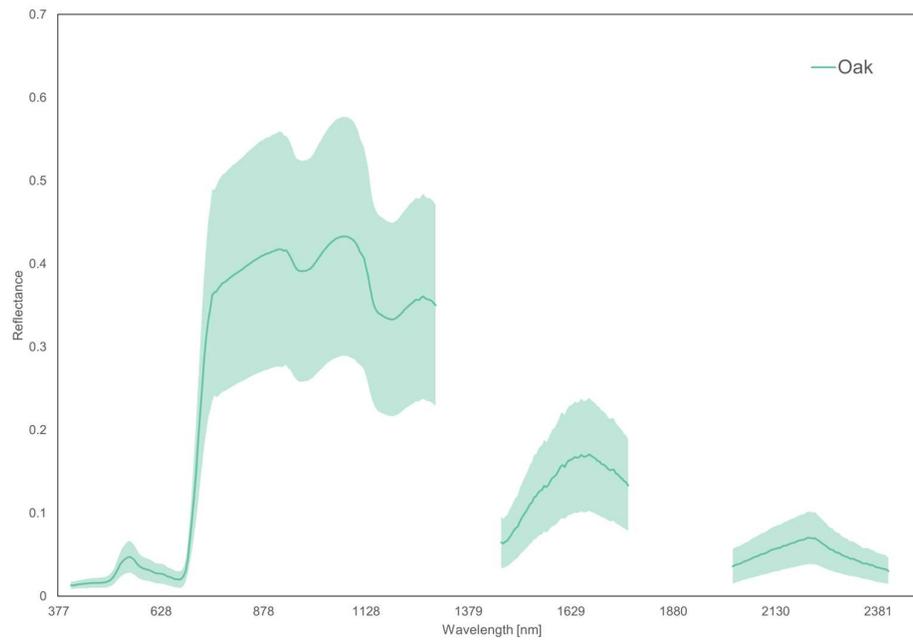


FIGURE A.1. Average reflectance spectra of oak samples with the shaded area illustrating the standard deviation.

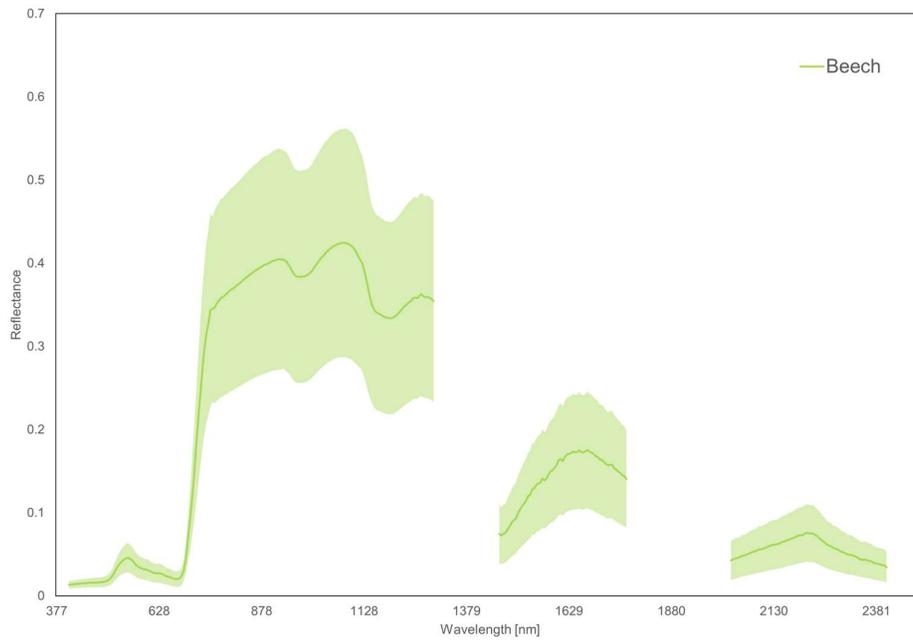


FIGURE A.2. Average reflectance spectra of beech samples with the shaded area illustrating the standard deviation.

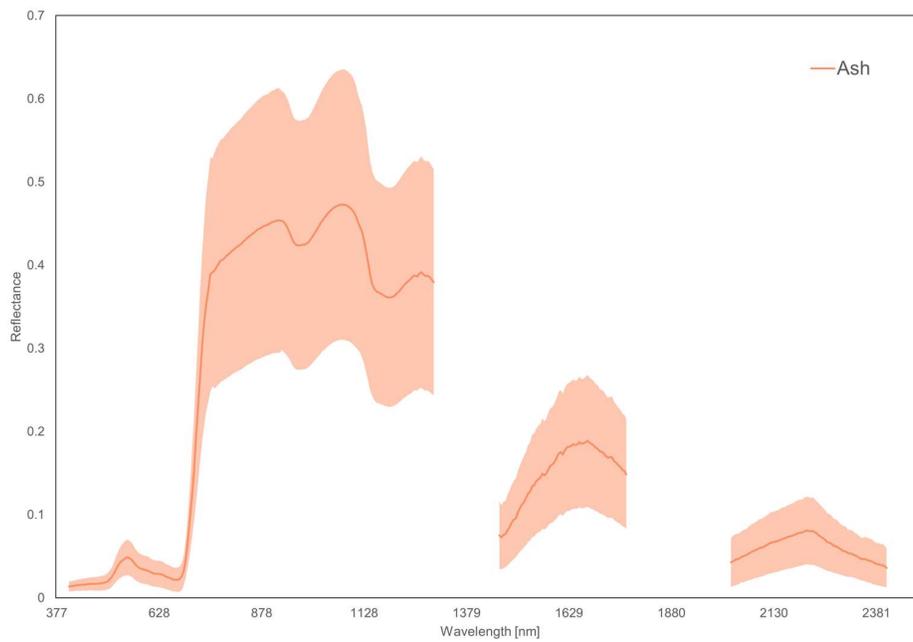


FIGURE A.3. Average reflectance spectra of ash samples with the shaded area illustrating the standard deviation.

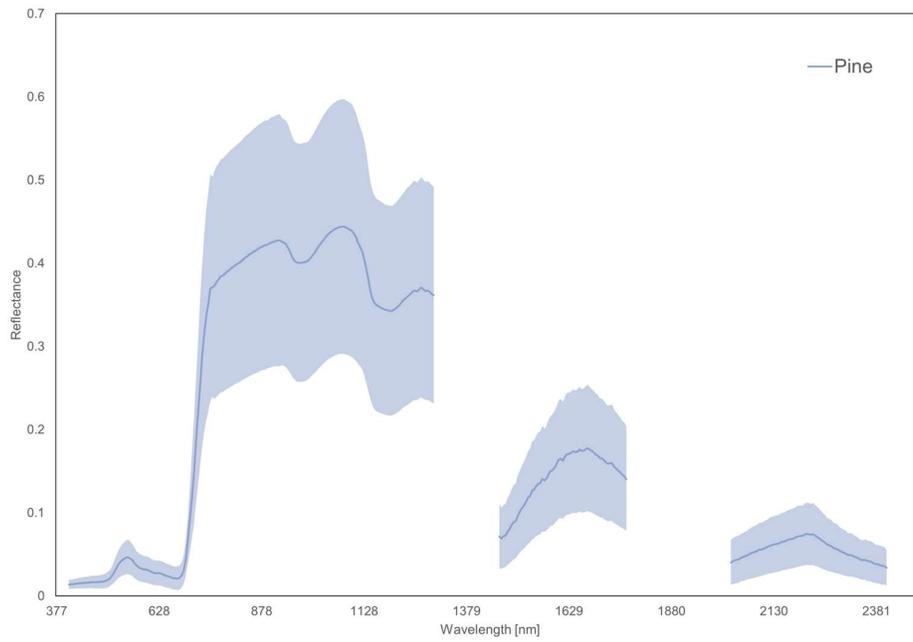


FIGURE A.4. Average reflectance spectra of pine samples with the shaded area illustrating the standard deviation.

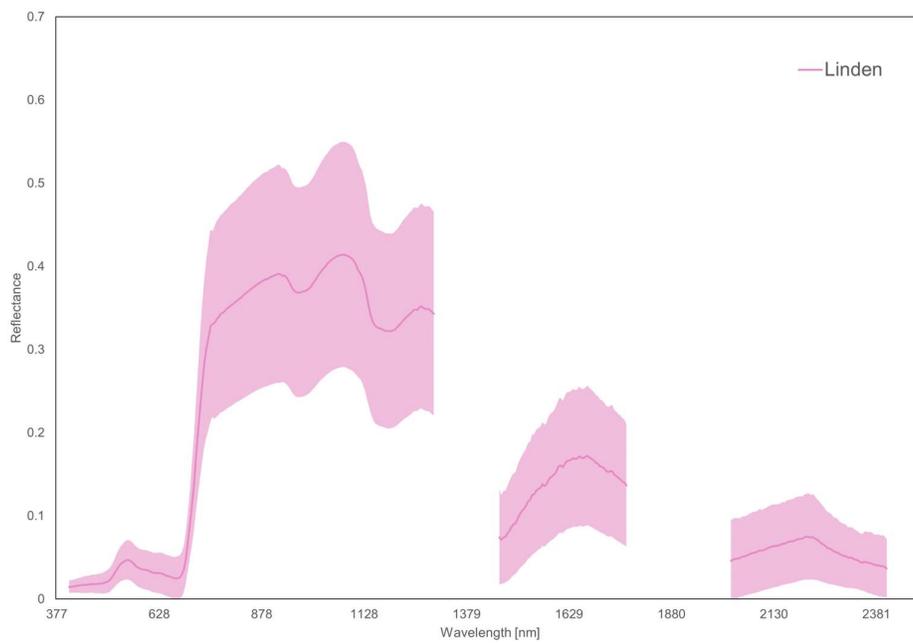


FIGURE A.5. Average reflectance spectra of linden samples with the shaded area illustrating the standard deviation.

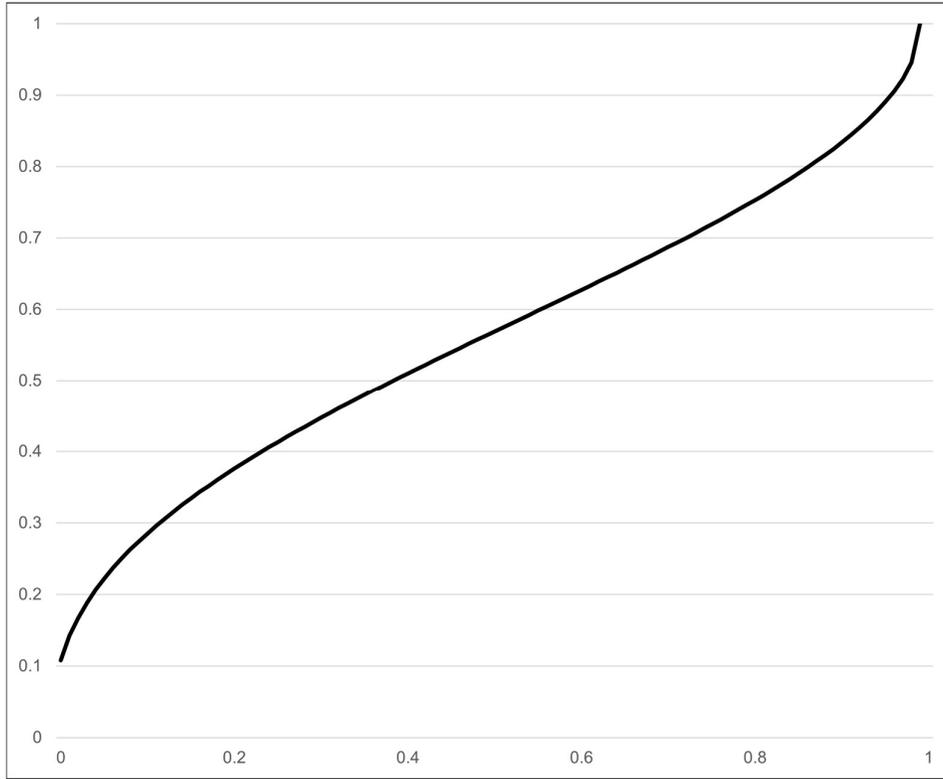


FIGURE A.6. Cumulative Beta distribution with shape parameters $\alpha = 0.4$ and $\beta = 0.5$.

		Ground truth				
		Oak	Beech	Ash	Pine	Linden
Prediction	Oak	82%	1%	8%	9%	0%
	Beech	16%	75%	7%	3%	0%
	Ash	4%	17%	69%	4%	7%
	Pine	13%	5%	10%	66%	7%
	Linden	21%	2%	26%	3%	47%

FIGURE A.7. Confusion matrix for the trained CNN showing the user's accuracy per species (i.e., the number of correctly classified samples divided by all positive predictions per class). Rows indicate the predicted species and columns the ground truth.

		Ground truth				
		Oak	Beech	Ash	Pine	Linden
Prediction	Oak	53%	0%	6%	9%	0%
	Beech	24%	86%	12%	6%	0%
	Ash	3%	10%	61%	4%	17%
	Pine	10%	3%	9%	79%	17%
	Linden	9%	1%	13%	2%	65%

FIGURE A.8. Confusion matrix for the trained CNN showing the producer's accuracy per species (i.e., the number of correctly classified samples divided by the actual number of samples per class). Rows indicate the predicted species and columns the ground truth.

		Ground truth				
		Oak	Beech	Ash	Pine	Linden
Prediction	Oak	76%	5%	5%	11%	3%
	Beech	13%	65%	13%	5%	5%
	Ash	14%	7%	67%	3%	9%
	Pine	18%	1%	6%	70%	5%
	Linden	15%	0%	28%	8%	49%

FIGURE A.9. Confusion matrix for the random forest classifier showing the user's accuracy per species (i.e., the number of correctly classified samples divided by all positive predictions per class). Rows indicate the predicted species and columns the ground truth.

		Ground truth				
		Oak	Beech	Ash	Pine	Linden
Prediction	Oak	49%	3%	4%	10%	6%
	Beech	24%	93%	28%	14%	29%
	Ash	11%	4%	58%	3%	24%
	Pine	12%	0%	4%	71%	11%
	Linden	3%	0%	6%	2%	30%

FIGURE A.10. Confusion matrix for the random forest classifier showing the producer's accuracy per species (i.e., the number of correctly classified samples divided by the actual number of samples per class). Rows indicate the predicted species and columns the ground truth.

		Ground truth				
		Oak	Beech	Ash	Pine	Linden
Prediction	Oak	76%	6%	7%	10%	1%
	Beech	15%	70%	10%	5%	1%
	Ash	4%	15%	69%	3%	10%
	Pine	20%	2%	10%	58%	10%
	Linden	21%	5%	27%	2%	45%

FIGURE A.11. Confusion matrix for the support-vector machine showing the user's accuracy per species (i.e., the number of correctly classified samples divided by all positive predictions per class). Rows indicate the predicted species and columns the ground truth.

		Ground truth				
		Oak	Beech	Ash	Pine	Linden
Prediction	Oak	48%	3%	5%	10%	2%
	Beech	25%	86%	17%	12%	3%
	Ash	3%	9%	59%	3%	24%
	Pine	17%	1%	10%	75%	27%
	Linden	6%	1%	9%	1%	44%

FIGURE A.12. Confusion matrix for the support-vector machine showing the producer's accuracy per species (i.e., the number of correctly classified samples divided by the actual number of samples per class). Rows indicate the predicted species and columns the ground truth.

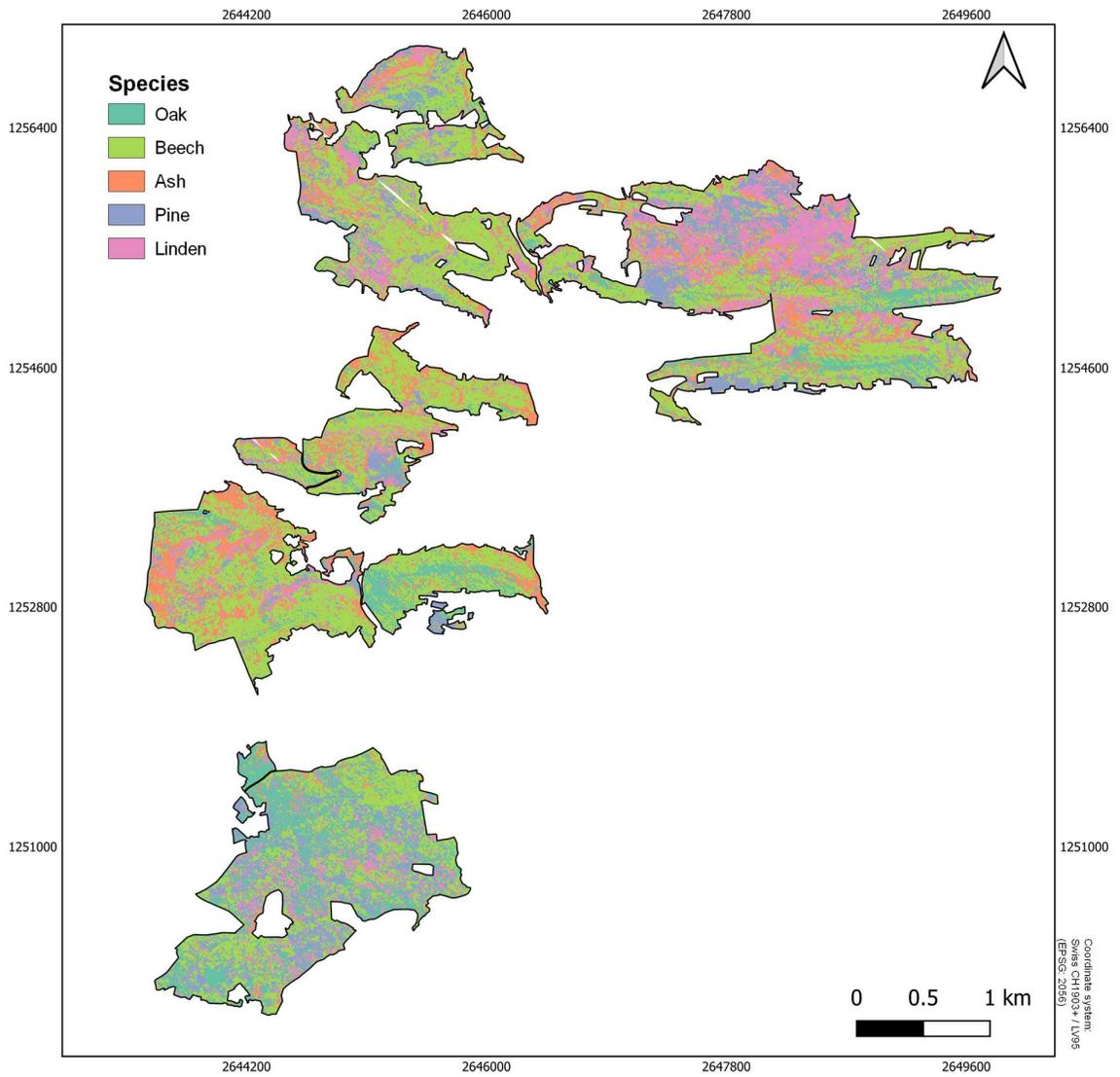


FIGURE A.13. Raw full study area tree species classification map showing all predictions without any thresholds applied.

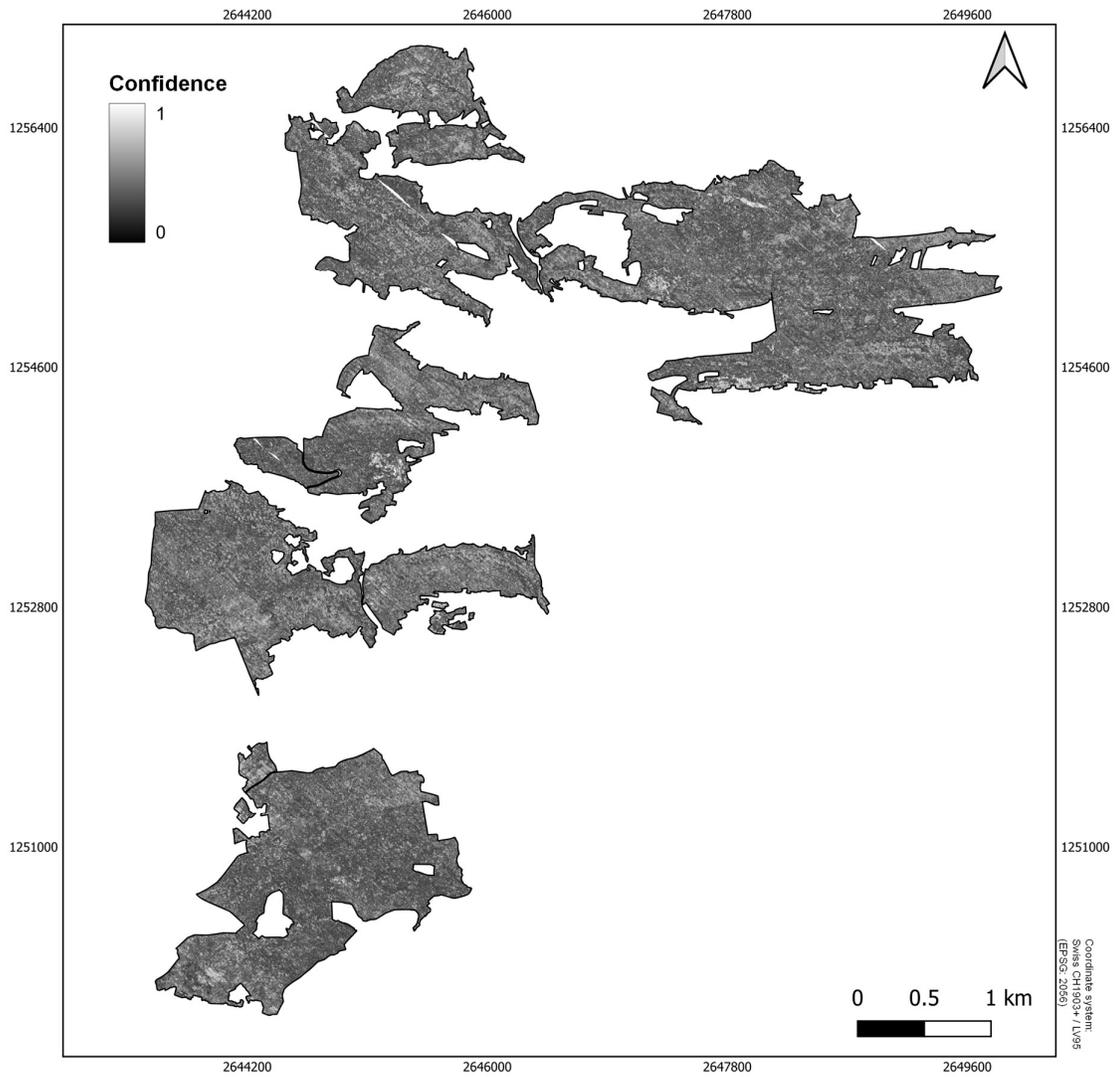


FIGURE A.14. Full study area prediction confidence. This shows the probability score with which the model assigned a label to each pixel.

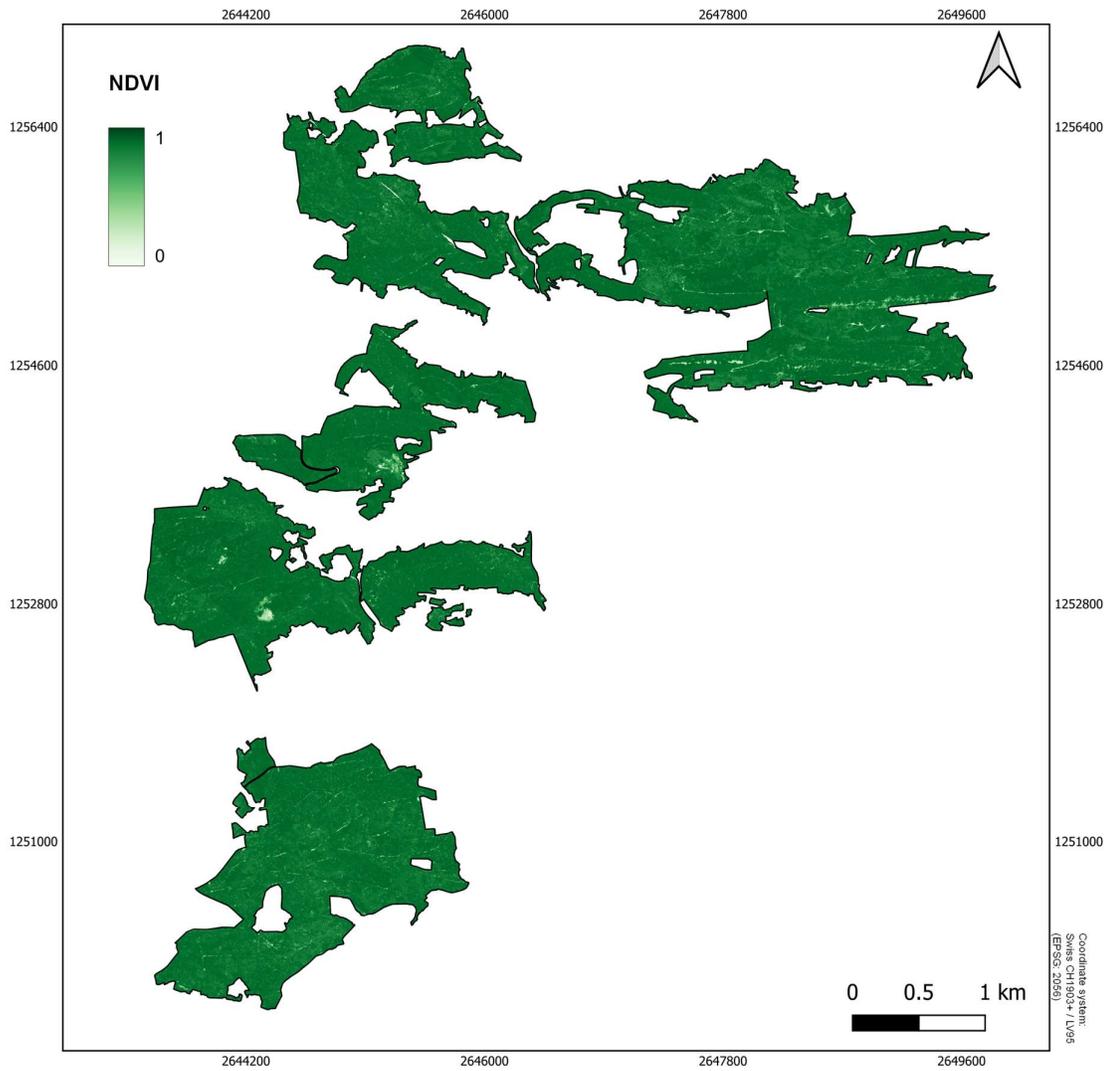


FIGURE A.15. Full study area NDVI. Calculated from AVIRIS-NG hyperspectral imagery.

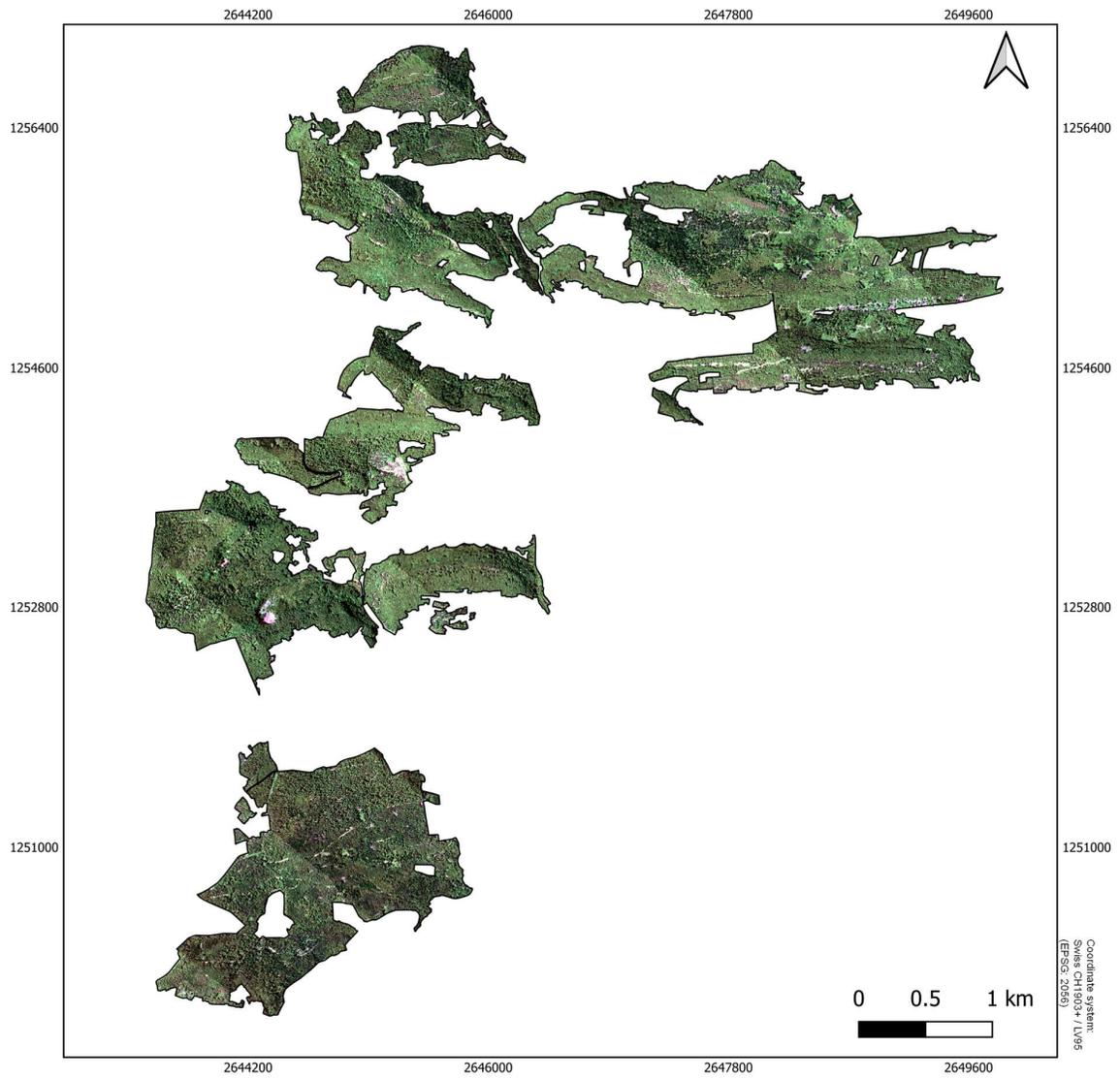


FIGURE A.16. Raw RGB image of the AVIRIS-NG hyperspectral imagery.

Acknowledgements

First, I would like to sincerely thank my supervisor Dr. Anna K. Schweiger for her constant support and advice during this thesis. I would also like to thank her for the opportunity of completing this thesis under the given circumstances with the latest data and in cooperation with the Jurapark Aargau. Speaking of, I would also like to thank the Jurapark Aargau, especially Anja Trachsel for the uncomplicated cooperation and the help in refining the objectives of this thesis. Furthermore, I really appreciated Dr. Mathias Kneubühler and Prof. Dr. Alexander Damm for their discussions and advice on the conceptual design of this thesis.

Personal Declaration

I hereby declare that the submitted Thesis is the result of my own, independent work. All external sources are explicitly acknowledged in the Thesis.

Date: April 30, 2022

Signature

Benjamin Zehnder

A handwritten signature in black ink, appearing to read 'B. Zehnder', is written over a horizontal line. The signature is stylized and cursive.