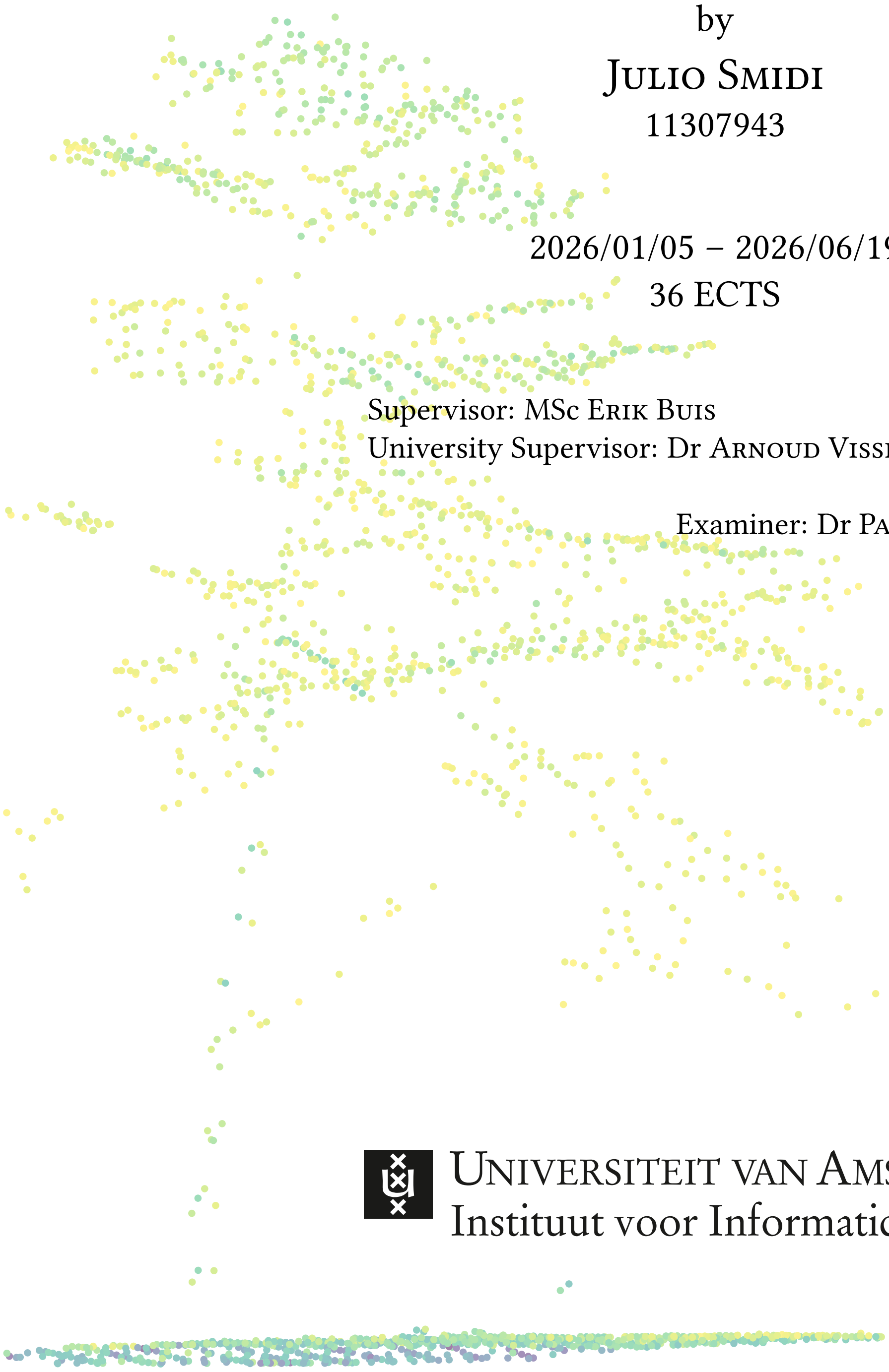
MSc Artificial Intelligence
Master Thesis

Taxonomy-Informed Classification of Tree Species Using Multimodal Remote Sensing Data

Or How to Recognise Different Types of Trees From Quite a Long Way Away



by
Julio Smidi
11307943

2026/01/05 – 2026/06/19
36 ECTS

Supervisor: MSc Erik Buis
University Supervisor: Dr Arnoud Visser
Examiner: Dr Pascal Mettes

Universiteit van Amsterdam
Instituut voor Informatica

Abstract

Current studies on classifying individual tree species using remote sensing data have mostly been conducted on a small, relatively uniform selection of trees and tree species. To investigate whether such an implementation is possible on a larger scale, a model was trained and evaluated on a dataset containing over 900 different tree species across the country of the Netherlands. Labelled data was collected from multiple Dutch municipalities, and open-source aerial imagery and LiDAR height maps were used to extract individual trees. By combining data from multiple modalities, the proposed method can learn from both spectral and structural features that are distinctive for species and reaches an accuracy of 80% and balanced accuracy of 32% across all included municipalities. Furthermore, we test multiple methods of integrating taxonomic information during training, and at inference an option to gracefully coarsen the predictions to a higher-level label in the taxonomy is added based on the prediction confidence, which improved classification accuracy to 95% and balanced accuracy to 51%. Lastly, we apply logit adjustment to correct for the effects of dataset imbalance. Results are evaluated both quantitatively and qualitatively at multiple levels of the taxonomy, and generalisation to out-of-distribution data was tested as well. While the current method successfully improves performance on a country-wide basis by using multimodal data, it does not entirely solve the dataset imbalance issue, and the results of integrating taxonomic information are mixed. Other implications and possible further research directions are discussed at the end.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Relevance

Accurate and recent spatial information on the diversity of tree species is of great use for ecosystem management and conservationism, both in rural and urban areas. Trees in urban regions have been shown to be important for the environment and health for many reasons. In and near cities, trees can reduce heating by providing shade and transpirational cooling [70, 55, 1]. This is also beneficial for energy usage, as less energy is needed to cool buildings and they can absorb greenhouse gas emissions and reduce noise from nearby industry [55]. In cities and other regions with high flooding risk, trees can reduce water runoff and mitigate flooding damage [7]. Trees also provide living and breeding space for many birds and insects in places where their natural habitat is dwindling [66]. Furthermore, there is evidence that trees are able to capture pollutants [6, 25]. Trees and natural scenery have also been shown to have positive effects on both mental and physical health of people [63, 88, 36, 45].

The effects that trees have on the environment and their chances of survival can however differ between species of trees. Different species have differing nutrient and water needs [73, 23], and tolerance levels to natural phenomena such as droughts and heavy rainfall vary between species [23]. Differences between species regarding carbon and nitrogen uptake and storage in the soil have also been found [83, 84, 26, 41]. How well trees manage air quality [5] and can mitigate flood damage also depends on the species. Being able to accurately identify species is therefore of high importance for urban planning and parks, as well as to monitor vegetation biodiversity. Having accurate species information can be used as feedback for planting locations and types, for monitoring invasive species, as well as to know which interventions worked and which have not, especially over longer periods of time.

1.2 Background

1.2.1 Remote Sensing for Tree Species Classification

Ecosystem monitoring can be done at multiple scales. Traditionally, this was only done through *in situ* observations which, although often very precise, are costly both in time and labor and can only have limited spatial coverage. Since the start of the 20^{th} century attempts have been made to monitor the earth's surface from the sky with the use of cameras attached to planes. Over the last decades, remote sensing has evolved to sophisticated imaging modalities that are used on planes and satellites and unmanned aerial vehicles (UAVs), which can be used to collect data more frequently, and over much larger areas. As a consequence, availability of data on the status of ecosystems has grown tremendously. For this reason, artificial intelligence (AI) is now commonly used to analyse remote sensing data, as the vast amount of data required to train machine learning (ML) and deep learning (DL) algorithms are more readily available.

Figure 1.1: An example of a multispectral aerial image, consisting of the RGB bands, collected from the PDOK winter aerial imagery [65].

Passive imaging

Passive sensors capture radiation that is reflected back from the objects on Earth's surface, similar to how a commercial camera would work. Whereas commercial cameras will generally only capture the range of radiation bands that are in the visible light spectrum, remote sensing sensors often capture light both in- and outside of the visible range, such as infrared light. Within passive imaging sources, a distinction is made between multispectral and hyperspectral imaging: multispectral imaging sensors only capture light at a few discrete bands of light, whereas hyperspectral imaging sensors capture a range of much narrower bands that approximate a continuous spectrum, thus providing a much more detailed view of the spectral composition of objects.

Trees and other vegetation reflect light in many spectral bands invisible to humans [92], which is why NIR and higher spectral bands are often used to analyse vegetation. Chlorophyll, for example, a green pigment present in plants, strongly reflects NIR light. Different species of trees can also show differences in spectral signatures, which can be used to distinguish between them [40]. Including infrared information in our aerial images can therefore provide us with additional information that visible light alone can not give. An example of aerial images from which the data will be collected are shown in Figure 1.1 and 1.2 for the RGB and NIR images respectively.

Active imaging

Active sensors, such as radar or LiDAR emit an energy signal (e.g. light), that when reflected is captured again and provides information on the physical structures by tracking among other things the intensity of the reflected signal and the number of times the signal has reflected against something. LiDAR point cloud data is obtained by emitting laser pulses against objects and then capturing the signal that is scattered back, that enables us to create a structural point cloud of objects that contain depth information. LiDAR obtained point clouds are often used in remote sensing for vegetation, as they can provide structural information where passive imaging alone is unable to. LiDAR information is most often either collected from terrestrial level scans

Figure 1.2: An example of a multispectral aerial image, consisting of the false colour infrared bands, collected from the PDOK winter aerial imagery [64].

(TLS), e.g. by backpack scanners or scanners mounted on cars, or from aerial level scans (ALS), e.g. mounted on planes, or on UAVs. An example of ALS collected LiDAR data can be seen in Figure 1.3. TLS LiDAR data is, compared to ALS obtained data, often of higher density and can be collected from different viewing angles, but is often less readily available as data acquisition is much more time- and labour-consuming.

1.2.2 Fusion of Imaging Modalities

A combination of LiDAR and multispectral data is consistently found to improve performance over using either modality alone both for segmentation [35] and classification [27, 2, 17]. When combining data of different modalities, such as multispectral aerial images and LiDAR point clouds, it matters at what point in the analytical process the data is combined. Roughly speaking, data can be fused at one of three moments [3]. Fusion can be done prior to any model training (early fusion), meaning that the different data modalities are overlaid such that they form a single mode of data. In the case of aerial images and point cloud data, one of such approaches would be to colour the points in the point cloud according to the colour values in the aerial images, essentially "colouring-in" the points. Data modalities can also be fused after training (late fusion). This usually entails combining the output features or classifications of separate model modules, each trained on a separate modality, by aggregating the resulting weights and then passing the aggregate through a regression or MLP classifier to obtain a prediction based on both modalities.

Early fusion has as an advantage that it allows for learning of interactions between the data modalities (maybe two species can only be separated based on both branch structure and leaf colour), but can be computationally more expensive, and might also introduce unwanted noise (e.g. because the point cloud density and aerial image resolution do not match). On the other hand, with late fusion, one can more easily determine the impact of each added modality (the trained point cloud and aerial image modules can be used to infer separately), however, there can be little interaction between the spectral and structural features as these are represented in disjoint latent spaces. A third option is middle fusion [85, 95, 76], where interaction is possible not on the level of raw data, but between learned features. Examples of these methods include implementing cross-attention modules that work on unpooled features, or a contrastive loss can be included to allow data

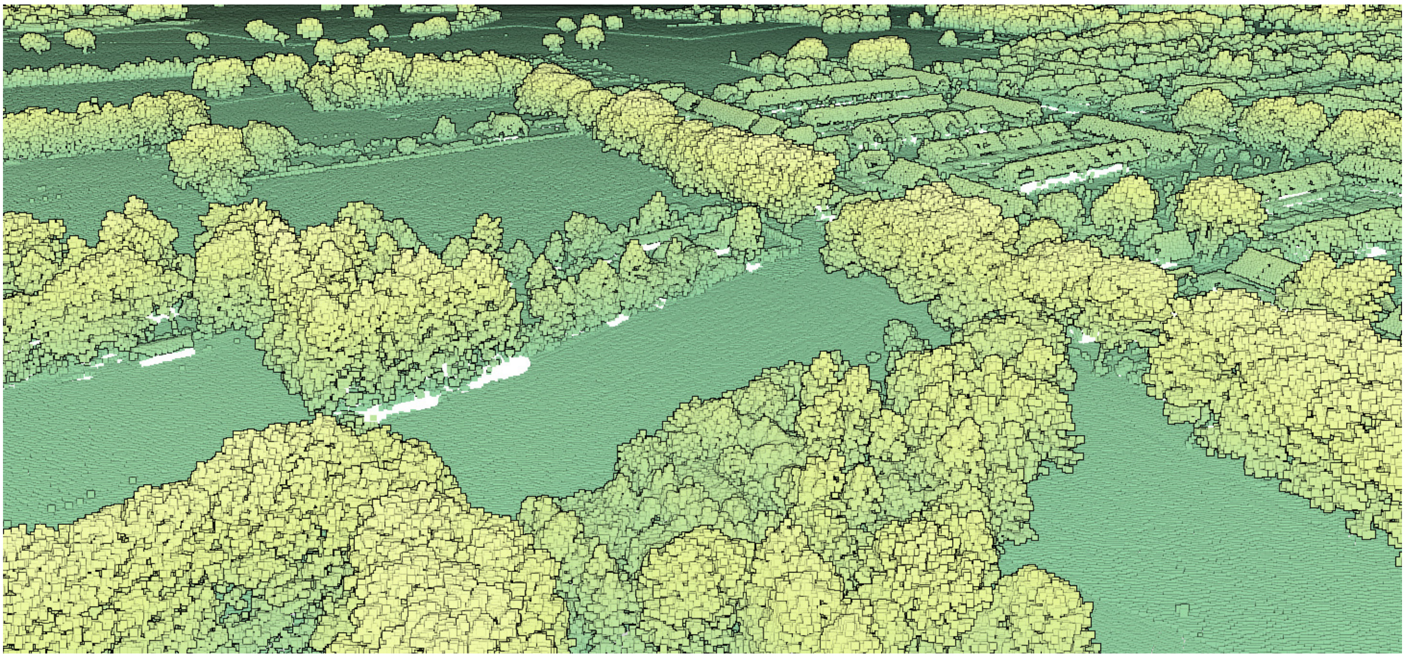


Figure 1.3: An example view of ALS LiDAR point cloud data, collected from AHN6 [78].

of different modalities to be projected into the same latent space. While this is usually computationally more expensive, this allows for interactions of qualities of the data from different modalities to be learned, (e.g. a combination of a certain shade of green leaves and a certain branch structure can be typical for a single species), that would not be learned if data is fused only at classification level.

1.3 Context of Current Investigation

The current project is commissioned by, and done in collaboration with NEO, a Dutch company focusing on earth observation applications for infrastructure, agriculture and sustainability. As of writing, accurate tree species classification over a large amount of classes is still an open problem. Currently, NEO curates a tree database containing information on more than 125 million trees in the Netherlands. This information is valuable to many municipalities, as they often do not have the resources to create and maintain their own tree catalogues. These municipalities have expressed a need for accurate identification of tree species, which is currently only available through manual annotations in some municipalities. Being able to provide nation-wide up-to-date information on tree species would therefore be highly useful for many instances in the Netherlands, and is potentially useful on a global scale as well. NEO currently has an existing tree species classification model [74], but it suffers from class imbalance and accuracy is relatively low for fine-grained species classification.

The current implementation of NEO is based solely on point cloud cut-outs of individual trees. A lot of noise can be present in a single tree point cloud sample: there can be differences in terrain, scanning angles, or flight height [101]. The planting context of a tree, in what geographical area and climate it is planted, can lead to visible intraspecies differences, introducing extra difficulties when trying to classify trees within a species [16, 75]. To reduce the impact of these sources of noise, we propose to include aerial images of trees to the learning pipeline, which add spectral information to the individual tree cut-outs and can provide planting context, assuming some of the area surrounding the tree is included in the images.

Past efforts in classifying tree species with remote sensing have shown positive results, however, the observed areas are usually small and often include a relatively small number of tree species. To the best of our knowledge, no remote sensing-based method that is publicly available exists as of yet that can accurately classify a similarly large multitude of tree species on a country-wide basis.

1.4 Research Objectives

Following the above problem statement, the main research objective for the current thesis will be to improve the existing tree species classification model of NEO. To accomplish this, we have proposed the following subquestions:

- Can we scale the performance of the model to be able to classify more species over a larger area compared to the current method?
- Does combining structural information with spectral information (by fusing point cloud data with multispectral aerial images) improve model performance?
  - Will data fusion using cross-modality feature interaction be more effective than straight feature concatenation?
- Can we improve classification on the tail section of the species distribution?
- Can we use taxonomic information to improve the classifier performance further, especially for the rare cases?
  - Will it be able to classify among multiple levels of the taxonomy (e.g. based on confidence)?
  - Can errors in classifying rare species be mitigated by using information from their more common siblings?

The remainder of this thesis is organised as follows: Chapter 2 reviews related work in tree species classification using remote sensing data and deep learning. Chapter 3 describes the dataset and methods used for the experiments in the current work. Chapter 4 contains the experiments and their results. In Chapter 5, the results and implications are discussed, and Chapter 6 concludes the thesis.

Chapter 2

Related work

2.1 Applications of Artificial Intelligence for Tree Species Classification

Machine learning has become a common and useful tool to analyse remote sensing data. The rapid growth of available remote sensing data has allowed data-hungry machine and deep learning methods to be applied to automate a broad number of observation tasks, such as object detection and classification, scene segmentation and vegetation monitoring. As the objective of the current investigation is to solve a classification problem, we will focus primarily on reviewing other classification research in the field of remote sensing for tree species. In this chapter we will give an overview of AI applications for tree species classification by remote sensing. We consider the usage of both passive and active imaging sources, as well as research on combining multiple imaging modalities.

2.2 Conventional Machine Learning Approaches

The first applications of ML on tree species classification (and remote sensing in general) consisted of conventional ML methods based on explicitly created features. Often used examples are support vector machines (SVMs) and linear discriminant analysis (LDA) and ensemble methods such as Random Forest (RF) or Gradient Boosting (e.g. XGBoost) algorithms [40]. These methods require relatively few resources and are generally more interpretable than modern DL approaches. However, they do require hand-crafted features, of which extensive examples can be found in the existing literature both for structural LiDAR features [80], and spectral aerial images [96] and for a combination of both [40, 20].

Having to manually select features limits the ability of feature-based ML methods to leverage information that is not explicitly fed to the models. These features also often have to be selected by human experts, which can be labour-intensive and difficult. Generalising to other domains is therefore usually not possible for these methods, and although RF classifiers and SVMs remain commonly used in tree species classification, in recent years their popularity shifted in favour of more powerful DL approaches such as convolutional neural networks (CNNs), vision transformers (ViTs) and foundation models (FMs), as these allow for better generalisation towards other unseen features through fine-tuning on new data.

2.3 Deep Learning Approaches

2.3.1 Deep Learning for Spectral Data

The use of deep neural networks (DNNs) has eliminated the need for hand-crafting features, as these networks can learn to extract useful features through many learning iterations. This allows DNNs to learn many interactions in the data that we are unable to recognise ourselves. Early applications of DNNs on classifying tree species made use of CNNs that are able to learn information on crown shape, texture and colour through hierarchical local spatial receptive fields automatically. Although many CNNs, like ResNets [33], have historically performed well in classification tasks, they struggle to capture global structures due to locality of their receptive fields.

ViTs [22] use self-attention mechanisms that do enable them to capture global structures. In the context of tree species classification, this allows these networks to more easily distinguish between trees that appear similar in local structures (such as leaf textures), but differ globally (e.g. having a different crown structure).

The self-attention mechanism present in Transformers, has in the recent years allowed for the development of FMs, models that are pretrained on enormous amounts of data in a self-supervised fashion, often on multiple modalities of data. One recent example is Seasonal Contrast [54] which provides a pipeline pretrained on remote sensing data, that was more effective for transfer learning for remote sensing purposes compared to models trained on ImageNet. Through contrastive learning, it is also able to pair images of the same location in different seasons. Clay [18] is another recent open-source foundation model trained on remote sensing data, that can be used to monitor many aspects on the surface of the Earth.

2.3.2 Deep Learning for Structural Data

In contrast to spectral (image) data, it is not possible to feed raw, unstructured point clouds to conventional computer vision models such as CNNs and ViTs, as these models expect regularly structured two- or three-dimensional input data. Several methods have been developed to use point cloud data for training, which we will discuss in this section.

Voxelisation methods

Early attempts to process point clouds consisted of rasterising the data into 3D grids, where a voxel is one such grid unit (analogous to a 2D pixel). Voxelisation allows models to learn information on the 3D structure of the point clouds. There are however, a few non-trivial problems with voxelising trees (and 3D structures in general). First, tree point clouds contain a lot of empty space, as the canopy is much denser than the area around the base of a tree stem. By voxelisation, information on the point density per region is also lost, as for example stems with high point density and more sparse edges of a canopy will have the same voxel size afterwards. On top of this, time complexity of 3D convolutions is in $O(n^3)$ and the increase in performance over multiple 2D convolutions is often marginal at best. For these reasons, voxelisation models will not be considered in the current research.

Point cloud projection

As a faster alternative to using 3D convolutions, some authors propose to project the point clouds to a 2D plane, such that 2D convolutions can be applied. This approach benefits from the fact that many computing architectures are highly optimised towards 2D convolutions training and it greatly reduces the time complexity to $O(n^2)$. However, a major disadvantage of using projected images, is that a lot of structural and all depth information of the point cloud is lost. To counteract the loss of depth information, some authors propose using a multi-view CNN (MVCNN), in which multiple viewing angles of the same object are input to a CNN.

The current best performing tree species classifier of NEO uses a MVCNN architecture trained on 64 side-views of orthogonally projected individual tree point clouds [74]. This model will be used as a comparative baseline to our implementations.

Native point cloud methods

By projecting or voxelising point clouds, relationships between individual points are not kept. The PointNet [67] architecture was created to work directly on the unordered set of point thus avoiding to need to map the points to a grid. Since its inception, many other methods have been developed to natively work on point cloud data without having to transform it. A notable example is PointNet++ [68], which enables the learning of local features using iterative farthest point sampling (FPS), which creates receptive field coverage analogous to CNNs. PointNet++ was found to be able to successfully classify individual tree species from ALS LiDAR data [48].

PointNet++ however does require more training and inference time compared to CNNs and did not perform better than the MVCNN for the tree registry of NEO [74]. Other native point cloud methods, such as PointConv [90] and Dynamic Graph CNN (DGCNN) [87] have shown improvement over PointNet++ in classifying individual tree species [44]. Recently, other lighter and faster point networks have also been published (e.g. LitePT [97], PointViG [104]), as well as versions using Transformer backbones (Point Transformer [102], Point Transformer v3 [91], including versions trained on ALS LiDAR tree point clouds (PCTreeS [46]). The reduction in computational demands and/or performance increase of these architectures validate their potential as alternatives to the currently used MVCNN.

Another development in this regard is the recently developed Erwin architecture [103]. Erwin makes use of ball tree partitioning, to allow for linear-time attention over unstructured point clouds. These ball trees are thereafter iteratively coarsened and refined, to increase the receptive field of the network, so it can learn both local and global features. It has shown a decrease in runtime over Point Transformer v3 and PointNet++, while retaining the expressivity of self-attention mechanisms. Because of these advantages, we will include an Erwin-based method when comparing model architectures.

A significant part of the success of native point cloud methods in classifying tree species is dependent on the point cloud density [15]. While aerial level scans can be collected more easily on a larger scale, it is less dense than terrestial level scans. It also suffers more from occlusion problems and larger variations in density, due to sometimes being unable to penetrate dense canopies [30, 98].

2.3.3 Deep Learning Fusion of Spectral and Structural Data

An example of an early fusion method is found in Reisi et al. [71], where features extracted from point clouds are combined with aerial image patches of concatenated RGB and near-infrared (NIR) bands as input to multiple neural networks. Another approach, Silvi-Net [10], instead makes use of a dual CNN setup, where one CNN is trained on 2D projections of individual tree point clouds, and the other on cropped out aerial images of these same trees. The resulting pairs of feature vectors are then fed through a single multilayer perceptron (MLP) classifier, to learn from both modalities. TSCMDL [47] uses a similar approach as Silvi-Net, but instead does not project the point clouds, but uses native point cloud methods. A downside of the previous methods, is that they only use concatenation of the data modalities after feature extraction. This means that interactions between data modalities are harder to learn, and that each modality is weighed as equally important, while for some species one of the modalities might be much more distinctive.

MMTSCNet [81] combines multiple modalities of just tree point clouds, namely the resampled point cloud, frontal and top-down depth image projections and numerical features extracted from the point clouds, by concatenating and weighting them prior to passing them through a classification head. By using dynamic weighting between the different modalities, the model is able to learn when to choose which modality to value more for the classification task at hand.

Allowing for more integration between data modalities, cross-attention modules have become more popular. Ge et al. [29] Mohla et al. [57] and Zhang et al. [99] implement a cross-attention-based convolutional fusion network for pixel-wise land cover classification of hyperspectral and LiDAR data. These groups of authors make use of cross-attention to leverage global dependencies within the data, and allow the different data modalities to enhance features from each other. However, these methods are all geared towards land cover classification rather than individual tree species classification, and use hyperspectral instead of multispectral imagery, the latter of which we will use.

Our approach will consist of using both cut-out tree patches from the aerial imagery, and for the point clouds we will use native point cloud methods to keep the structure and non-Euclidean nature of the data. A limitation of using concatenation-based fusion, is that both modalities are weighted equally, and there is no room for cross-modal interaction, as the features remain in disjoint latent spaces. To address this, we implement a cross-attention fusion module that lets spectral and structural features interact directly, and we compare it against plain feature concatenation as a baseline. In addition, we apply a contrastive loss that projects the two modalities into a shared latent space, providing a further means of cross-modal interaction during training. The details of these implementations will be described in Section 3.3.3.

2.4 Taxonomy-Based Classification

Tree species do not exist in isolation, but rather are ordered hierarchically in a taxonomy. The purpose of a taxonomy is to categorise similar species in close categories, where children in a parent taxon often share characteristics. These taxa are ordered along the following levels, from high to low in the taxonomic tree: kingdom, phylum, class, category, order, family, subfamily, genus and species; class and subfamily are optional levels. All trees fall under the kingdom of plants (*Plantae*), and start separating from the phylum level. Species within the same genus tend to be visually similar to each other, a phenomenon described by the phylogenetic signal [59], including for tree species [9] and this similarity decreases monotonically up the taxonomy [14]. Taxonomic structure can therefore be exploited in three broad ways: by shaping the *architecture* of the model (separate heads or conditional paths for each level), by shaping the *loss* (penalising distant mistakes more than close ones), or by shaping *inference* (coarsening predictions when species-level confidence is low).

Use in model architecture

One recent proposal on the architectural side is BioCLIP [77], which is a CLIP based foundation model that has learned to match images of biological species to their taxon, including the earlier taxa in the hierarchy, such that the model learns more fine-grained hierarchical representations. Another foundation model based on hierarchical plant taxonomy is HiTree [38] which makes use of a cascading multi-level classifier to classify trees based on hyperspectral imagery by transferring information across taxonomy levels, allowing it to refine predictions from coarse to fine categories. Switcher-HNet [50] contains a switcher module for needle-leaf and broadleaf classifiers that can switch between these foliage types when one of these two classifiers predicts a higher likelihood for the other foliage type. A separate classifier head then predicts the species within that foliage type. This approach is similar to multi-head classification approaches, where separate classification heads are trained for predicting on family, genus and species level [37]. Some other approaches include using a coarse-to-fine classification approach, where coarse and fine label classifiers can interchange learned information [51, 13].

Use for loss and inference

A different approach to use taxonomic information for classification is used by Park and Kwon [62]. Instead of improving the distance of the mistake made, the authors apply a graceful coarsening based on a prediction certainty threshold. If the classifier is too unsure about its prediction on the species level, it instead makes a prediction on a higher level of the taxonomy, until the threshold is surpassed, or the top of the taxonomy is reached. In conjunction, they use a taxonomy graph with learnable embeddings using GraphSAGE [32] and a multi-level loss to improve taxonomic coherency and to create better separating margins for the different taxa. Rather than modifying the loss to penalise distant mistakes, an alternative is to keep standard cross-entropy and modify the targets themselves. Hierarchical label smoothing spreads part of the probability mass from the true species across other species in proportion to their taxonomic similarity [8, 58]: close relatives – same-genus siblings – absorb most of the residual mass, while more distant taxa receive proportionally less. The authors of MycoAI [72] successfully apply multi-level hierarchical label-smoothing for classifying fungal sequences in their multi-head prediction architecture, which is trained to make a prediction for each of the taxonomic levels, using smoothed labels on each level.

Use for imbalanced data

The above approaches treat taxonomy as a structural prior on the classifier or the loss under the implicit assumption of a balanced class distribution. When the species distribution is itself heavily imbalanced, the taxonomy additionally becomes useful for transferring signal from frequent head species to their rare siblings, an idea made explicit by the Deep Realistic Taxonomic Classifier of Wu et al. [89], who introduce a Correctly Predicted Bits metric that acknowledges the relative correctness of predictions that are taxonomically close even when not exactly right. This connection between hierarchy and imbalance motivates the next section.

2.5 Learning Long-Tailed Distributions

The distribution of tree species in the Netherlands is severely imbalanced: a handful of common native and ornamental species dominate the dataset, while hundreds of rarer species are represented by only a few examples each. Such long-tailed distributions, in which class frequencies decay roughly according to a power law, are common in fine-grained recognition tasks and pose two specific problems for cross-entropy training. First, the per-step gradient is dominated by head classes simply because they appear vastly more often in each mini-batch, leaving the tail with a faint training signal. Second, even when the representation generalises reasonably well, classifier weight norms tend to grow disproportionately for head classes over training, biasing the decision boundary away from rare classes [39].

Long-tailed image recognition has been studied extensively outside of remote sensing, as many naturally occurring phenomena follow such a distribution. Most studies make use of imbalanced versions of popular classification benchmarks such as CIFAR-10-LT, CIFAR-100-LT and ImageNet-LT [52]. One particularly interesting analogue to the current investigation is the iNaturalist dataset [82], as it contains thousands of species categories whose frequency of appearance has a skew of two to three orders of magnitude between the head and tail classes. Approaches to long-tailed image recognition are commonly grouped into three categories [100]: data-level methods that rebalance which samples the model sees during training (over- and under-sampling, data augmentation), loss-level methods that rebalance the per-sample or per-class gradient contribution, and decision-level methods that adjust the classifier or its outputs after representation learning. Representative methods from each category are described below.

The simplest interventions act on the data or on the loss to counteract this bias. Inverse-frequency sampling and class-balanced sampling oversample rare classes so that each class contributes roughly equal expected gradient mass per epoch, while class-balanced loss reweighting based on the effective number of samples [19] downweights majority classes in the cross-entropy loss objective. A common practical solution to these techniques is to truncate the tail: classes that fall below a minimum sample threshold are excluded from training because they provide too weak a signal for the model to learn a meaningful decision boundary, and their misclassification rate would otherwise dominate balanced evaluation metrics. While this might be a crude measure, this is standard practice in fine-grained species benchmarks and is the strategy we adopt in the current investigation.

A different approach is to modify the classifier's decisions rather than the training loss. Logit adjustment [56] shows that many of the above methods can be interpreted as approximations of a single correction: subtracting the log of the class prior from the model's logits at inference, or equivalently adding it during training as a per-class additive bias. This recovers the Bayes-optimal classifier under a balanced test distribution without requiring any change to the representation. Similarly, decoupled training [39] demonstrates that strong long-tailed performance can be obtained by first training the feature extractor with standard instance-balanced sampling and only afterwards re-training the classifier with a class-balanced sampler, an idea the authors call classifier re-training (cRT). The idea to learn good features on the natural distribution, then correct the decision boundary has since been extended: bilateral-branch networks [105] combine a conventional branch and a rebalancing branch through a cumulative learning schedule, and ensemble-of-experts approaches have been proposed such as RIDE [86] in which multiple classifiers specialise on different parts of the class distribution.

For fine-grained recognition in particular, tail classes are often visually similar to head classes, so methods that transfer information between related classes are particularly important. In our case, the hierarchical taxonomic relationships described in the previous section provide this mechanism, through both target-side smoothing and a taxonomy-graph prior on the classifier, letting majority-class signals support the learning of rare species.

Chapter 3

Methods

3.1 Data

Multiple data sources were used to create the dataset for training and evaluation of the models, all of which will be described below.

3.1.1 Boombasis

The Boombasis [60], the database of segmented tree crowns curated by NEO, contains the location coordinates of the stems of 2.1 million trees over 35 municipalities in the Netherlands [12]. Of these municipalities, 33 include the tree species in the dataset for at least a section of the annotated trees (see Figure 3.1), resulting in 1,666,101 labelled trees. Individually delineated tree crown segments were also generated for the trees corresponding to the located stems of the trees, which will be used for cutting out the individual trees. These individual tree crowns were segmented using an inverse watershed method, that finds local maxima in the height map that represent tree tops, which expand downwards to delineate a single crown, until another tree is reached or a rapid decrease in height occurs. As beneath a single crown canopy there may exist multiple trees, a polygon splitting algorithm is thereafter applied to the tree crowns, to segment individual trees. This splitting is done by first detecting the number of tree stems present in a crown, and then using a Voronoi splitting algorithm on the polygon to create disjoint sub-polygons for each tree that each represents a single tree [12]. This last tree splitting step was not applied in the previous tree species model of NEO [74], influencing the ease of direct comparison.

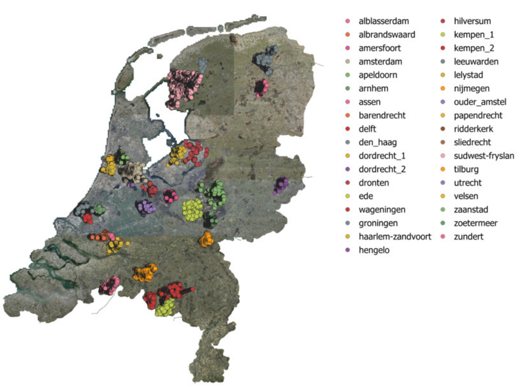
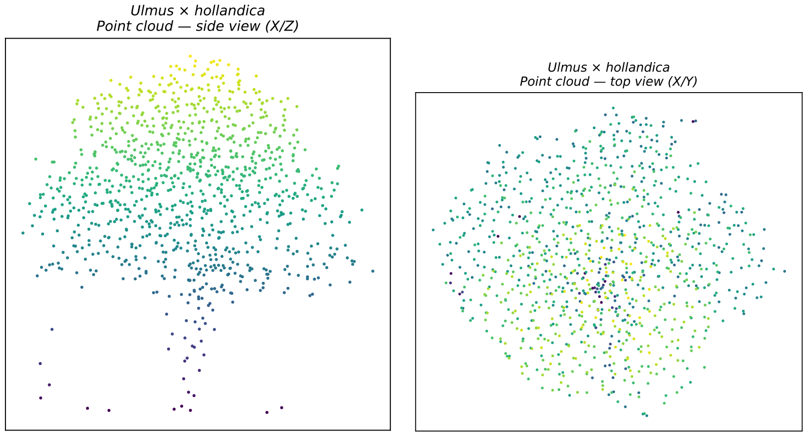


Figure 3.1: A visual overview of the municipalities from which the labelled data is obtained. Note that Dordrecht and Kempen have been collected at 2 separate occasions over 2 different sections of their municipalities, and are thus considered separate sources.



(a) Side view (X/Z) (b) Top view (X/Y)

Figure 3.2: Side and top view of a LiDAR derived point cloud of a *Hollandse iep* (Dutch elm), standardised and coloured by height. On the side view, the stem can be distinguished from the foliage.

3.1.2 LiDAR Data

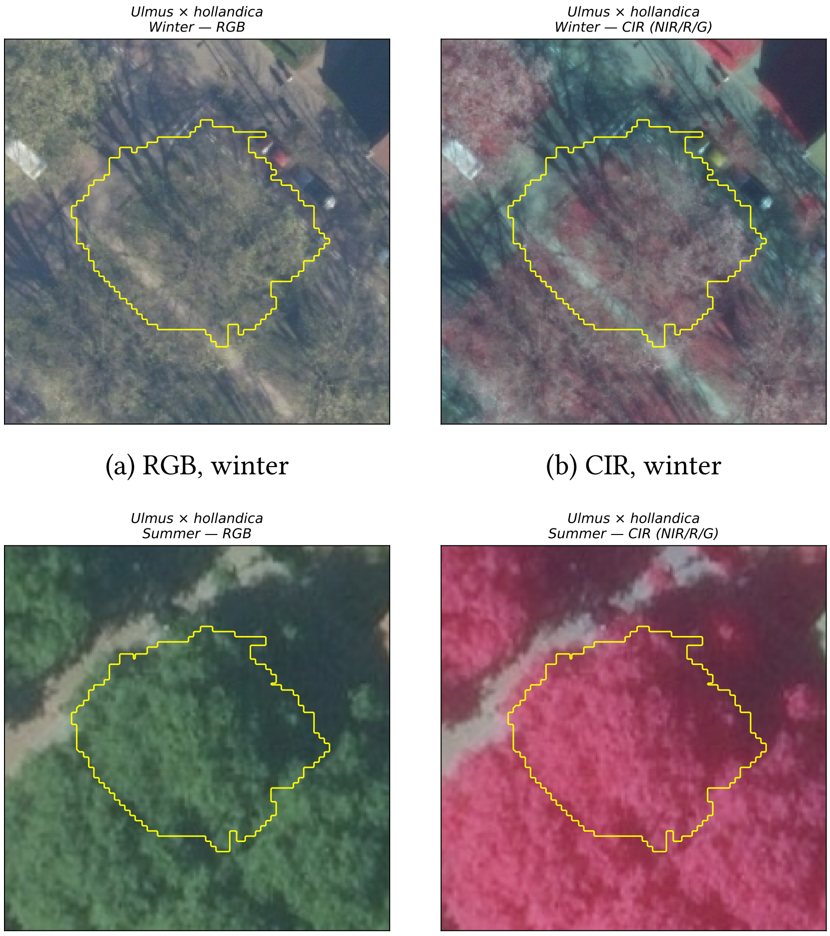
The LiDAR point cloud data of individual trees was collected from the *Actueel Hoogtebestand Nederland* (AHN) [78]. The AHN provides open-source ALS LiDAR height measurements of the Netherlands, collected over multiple years. As of writing, AHN6 is in process of collection, with only the east provinces of the country being currently available. For the remaining provinces, we use the previously published AHN5, and for very small sections of the country that are not covered by either, AHN4 is used. A more detailed visualisation of the area covered by each version can be found in Appendix K.

Using the tree crown segments of the *Boombasis*, we are able to cut out the point clouds of individual trees from the AHN. These individual trees are standardised and preprocessed and are the input for the point cloud module of our model. In the AHN, each point has an object classification (e.g. water, ground, building) that can be used to filter. To reduce compute and prevent any additional information to overfit on, we remove all but the *miscellaneous* class of points, in which trees belong, including the ground below a tree. Other standardisation steps were taken from the previous tree species classifier implementation of NEO [74, 12]. These include translating on the X- and Y-axes of the point cloud such that its centroid is at the origin and setting the original ground points to have an average height of 0 on the Z-axis. The point clouds are then normalised to a unit sphere for training stability. An example point cloud can be seen in Figure 3.2, and more in Appendix A.

On top of the point coordinates, we used two more attributes: the intensity values of the points, standardised to the range of $[0, 1]$ using a linear transformation and the relative return, obtained by dividing the return number by the number of returns for each point. Intensity values provide information not only on the location of the points, but also on the reflectivity of the surface, the angle at which the surface is hit, and on the distance to the receiver. The relative return can help a model distinguish between more dense, woody structures such as the trunk and branches, as these points will have higher intensity value than more permeable structures such as leaves. Different types of trees will also have different branch and leaf density. These measurements are complementary to the position of the points, and are thus informative above the point coordinates alone.

3.1.3 Aerial Imagery

For the aerial imagery, we used RGB [65] and colour-infrared (CIR) [64] orthophoto mosaics of the Netherlands that are provided by PDOK. PDOK collects and publishes aerial images twice per year, and the latest



(a) RGB, winter

(b) CIR, winter

(c) RGB, summer

(d) CIR, summer

Figure 3.3: Aerial imagery of the same *Hollandse iep* (Dutch elm) across two imaging bands; RGB and false-colour infrared (CIR), and two seasons; winter and summer. Note that in the winter, there are no leaves on this tree, while in the summer there are. The chlorophyll contents of the tree, coloured red, that are visible on the infrared bands are much more pronounced in the summer compared to the winter, but still present for both seasons.

2025 data consists of orthophotos with a resolution of 5 by 5 to 8 by 8 centimetres for the winter photos and 25 by 25 centimetres for the summer photos, both for RGB and CIR. We overlayed these aerial images with the tree bounding polygons from the Boombasis, and cut out individual tree "chips" to use as input for the aerial module of our model. The cut-out tree chips have the tree stem as their centroid and the bounding box is a square with a side length 1.5 times the longest side of the tree polygon. All chips are thereafter resized to a 224×224 pixel image as to standardise the size for neural network input. Finally, we concatenate the infrared band of the CIR images to the RGB images, and add an optional fifth masking band to distinguish the tree from the background where pixels within the tree polygon are set to 1 and all others to 0. Example RGB and CIR images can be seen in Figure 3.3 and Appendix B and C. Note that the bounding boxes are based on the winter imagery, so they will not bound the summer images nicely. Regardless, there will be less background included in the summer images, due to the foliage present.

3.1.4 Taxonomic Data

Data on the taxonomy of tree species was collected from the EPPO Global Database [24]. We used EPPO codes to normalise the labels given to the trees from the divergent labelling methods used by each municipality. Using EPPO codes also allowed for easy tracking of the taxonomic lineage of each tree. For plant species, EPPO codes consist of five letters, the first three of which characterise the genus, while the latter two characterise the species. All higher orders in the taxonomy start with "1", followed by three letters of the corresponding taxon, and closed by the letter corresponding to the taxonomic level (e.g. "G" for genus). A number of trees found in the dataset were local cultivars of one or more other trees that did not have a corresponding EPPO code and thus could only be reliably labelled by their genus. For those tree species, a replacement species code was made, such that all trees have a species label, and aggregation by level can be done equally for all trees. Example EPPO codes for two trees and their lineages are shown in Table 3.1.

Level	**Taxon (Apple)**	**EPPO**	**Taxon (Larch)**	**EPPO**
Kingdom	*Plantae*	1PLAK	*Plantae*	1PLAK
Phylum	*Magnoliophyta*	1MAGP	*Pinophyta*	1PINP
Class	*Angiospermae*	1ANGSC	*Pinopsida*	1PINC
Category	*Fabids*	1FABD	-	-
Order	*Rosales*	1ROSO	*Pinales*	1CONO
Family	*Rosaceae*	1ROSF	*Pinaceae*	1PINF
Subfamily	*Spiraeoideae*	1SPVS	-	-
Genus	*Malus*	1MABG	*Larix*	1LAXG
Species	*M. sylvestris*	MABSY	*L. decidua*	LAXDE

Table 3.1: Example lineage based on EPPO codes of the wild apple tree and the larch with corresponding EPPO codes and scientific names. Note the missing category and subfamily taxon for the larch; some lineages have no subfamily or category taxa and are instead directly absorbed by the taxon above.

A complete distribution of all taxonomic classes is shown in Appendix F. We can see that the distribution of species is quite unbalanced; already at the phylum level most trees fall under one phylum, while the other four phyla and their corresponding species present are relatively underrepresented. The frequency distribution of all species can be found in Figure 3.4 to more clearly show the class imbalance problem.

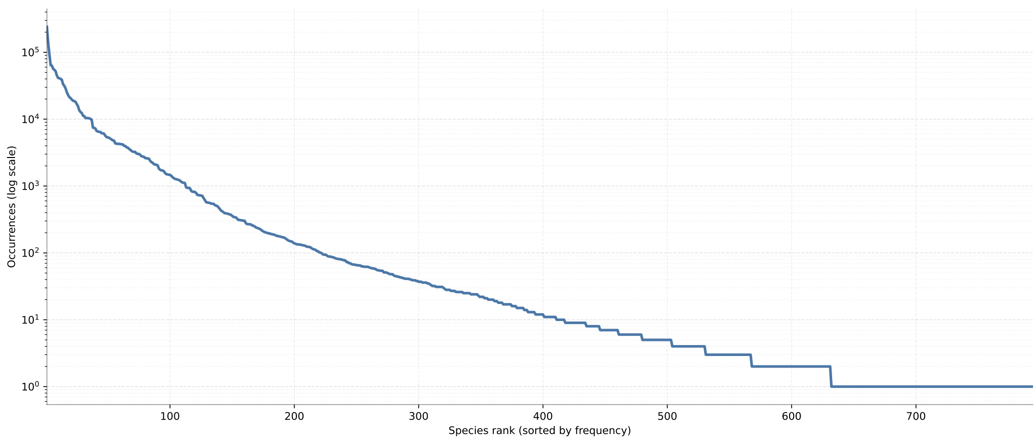


Figure 3.4: Frequency counts per species, scaled logarithmically.

3.2 Data Preprocessing

The eventually used data was subject to a few selection criteria. First, any point clouds that had less than 1,024 points after filtering out the ground were discarded. Such point clouds often did not contain a tree, either due to noisy measurements, an incorrect estimation of the trees position from the polygon segmentation or from the tree being removed in between the moment of labelling and the newer AHN measurements. For consistency reasons, aerial images of these same trees were removed as well, even for the models trained only on aerial images.

Since we are dealing with a large number of species following a long-tailed distribution, a floor value of 10 for species frequency was also implemented. This value was chosen as we used 10 cross-validation folds for training, and this guarantees a tree to be present at least once in each fold. Species that were too rare and from which the model could not learn well enough without intensive upsampling, were removed. Of the original 1,666,101 trees from 906 different species, the floor value and point filtering left 979,337 trees over 345 species to use for training and evaluating the model.

3.3 Model Architecture

We distinguish three model families: a unimodal aerial model, a unimodal point-cloud model, and a multimodal fusion model that combines the two. All three share the same encoders, projection heads and classification head, so that differences in performance can be attributed to the fusion mechanism rather than to incidental architectural changes. The fusion-through-concatenation diagram is shown in Figure 3.5. The unimodal variants and the other two multimodal variants are shown in Appendix D and Appendix E respectively.

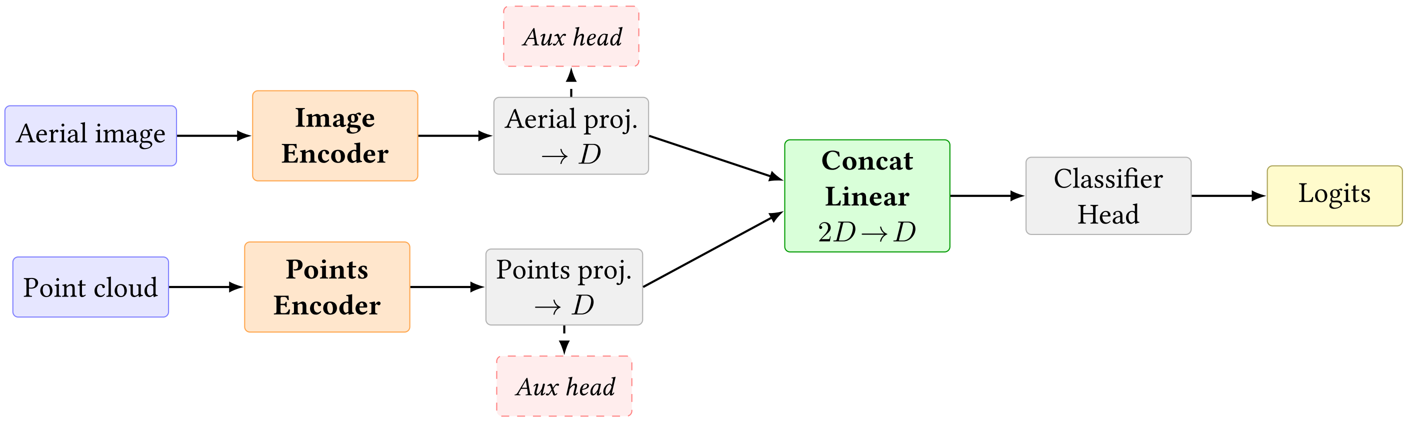


Figure 3.5: Fusion model — concatenation variant. Both modalities are encoded in parallel, each projected to the shared dimension D, and the two vectors are concatenated and projected back to D before classification. Auxiliary per-modality heads (dashed) provide direct supervision to each branch during training.

3.3.1 Encoders

The aerial branch uses a ResNet-18 encoder [33]. The first convolutional layer is widened from three to 4 or 5 input channels to accept the stacked RGB, near-infrared and optional foreground-mask bands; when pretrained weights are used, the original RGB filters are retained, the infrared filter is initialised as the mean of the RGB filters, and any further channels are initialised randomly. The encoder produces a global feature vector of dimension d_a after average pooling, and a spatial feature map of $d_a \times 7 \times 7$ before pooling (for a 224×224 input), which the cross-attention variant uses directly.

The point-cloud branch uses a PointNet++ encoder [68] operating on the standardised tree point cloud together with its per-point intensity and relative-return features. Three set-abstraction layers progressively subsample and aggregate local neighbourhoods: the first samples 512 centroids (radius $= 0.2$), the second 128 centroids ($r = 0.4$), and the third pools all remaining points into a single global descriptor of dimension $d_p = 1024$. The second layer additionally exposes 128 per-group tokens of 256 dimensions, each with an associated 3-D centroid, which the cross-attention variant uses as structural tokens.

3.3.2 Projection and Classification Heads

Each encoder output is mapped into a shared D-dimensional embedding space by a two-layer projection head (linear, layer normalisation, ReLU, linear, layer normalisation). The shared classification head maps the D-dimensional fused representation through a bottleneck (linear to 512, layer normalisation, GELU, dropout) to the C output logits. The bottleneck keeps the representational width above the number of classes while reducing the tendency to memorise observed in wider heads.

3.3.3 Fusion Strategies

All fusion variants encode both modalities in parallel and project them to the shared space, yielding aerial and point embeddings $a, p \in \mathbb{R}^D$. They differ only in how these are combined.

Concatenation

The two embeddings are concatenated and projected back to D dimensions (linear, layer normalisation, ReLU) before classification. This is the simplest fusion and serves as our baseline.

Contrastive alignment

A small gating network predicts a scalar weight $\alpha = \sigma(W[a;p]) \in (0,1)$ from the concatenated embeddings, and the fused representation is the convex combination $\alpha\, a + (1-\alpha)\, p$. This lets the model weight the more informative modality per sample. In addition, an optional cross-modal contrastive loss (Section 3.4.1) is applied to a separate pair of projection heads, encouraging the two modalities to share a latent space.

Spatial cross-attention

Rather than fusing globally pooled vectors, this variant fuses the *spatial* features of the encoders. The aerial feature map is pooled to a fixed 7×7 grid and flattened into 49 patch tokens; the point branch contributes its 128 ball group tokens. Both sets are linearly projected to D dimensions, and each token receives a positional embedding (a learned grid embedding for aerial tokens, an MLP of the 3-D centroid for point tokens) and a learned modality-type embedding to indicate its source modality. A single learned `[CLS]` token is prepended to serve as a learned pooling vector.

This combined sequence (`[CLS]` token, 49 aerial tokens, 128 point tokens) is processed by a Transformer encoder of L layers with h heads. Within each layer every token is updated as an attention-weighted sum of all the others: it matches its query against every token's key and blends their values accordingly. Because the sequence interleaves both modalities, a token attends both to its own modality (self-attention) and to the other (cross-attention) in a single operation, so an aerial patch can draw on the point groups covering the same region and vice versa; stacking L such layers lets these within- and cross-modal interactions compound, leading to the actual fusion. The fused representation is then read out by concatenating three pooled views of the Transformer output (the `[CLS]` token, the mean over the modality tokens, and their element-wise maximum) and projecting back to D. This multi-pool readout avoids the gradient bottleneck of a `[CLS]`-only summary.

3.3.4 Auxiliary Supervision and Warm-up

Because the backbones are trained from scratch, the fused path alone provides a weak gradient to each encoder. We therefore attach an auxiliary linear classifier to each modality and add its cross-entropy to the main objective, weighted by λ_a^{aux} and λ_p^{aux} for aerial images and point clouds respectively. For the concatenation and contrastive variants these heads operate on the global embeddings a and p; for the cross-attention variant they operate on the mean of each modality its projected spatial tokens, so the auxiliary gradient lands on the same representation the Transformer consumes. For an optional initial warm-up period the fusion module is bypassed and the two branches are trained independently through their averaged logits, after which the selected fusion mechanism is enabled.

3.4 Learning Objectives

3.4.1 Loss Functions

Cross-entropy loss

The main loss function that is used for all models is a cross-entropy loss function, a useful and commonly used criterion for multiclass classification. For C classes, this loss is given by:

$$\mathcal{L}_{CE} = -\sum_{c=1}^{C} y_c \log(\hat{y}_c)$$

with y_c the binary indicator indicating if class c is the correct classification for the observation and $\hat{y}_c$ the predicted probability that the observation is assigned to class c, obtained by applying a softmax function to the raw logits output by the network. As the true label y is one-hot encoded, this loss simplifies to:

$$\mathcal{L}_{CE} = -\log(\hat{y}_c) \tag{3.1}$$

By minimising this loss, the model is penalised heavily when it assigns a low probability to the correct class, encouraging the network's predictions to align more closely with the true labels.

Cross-modal contrastive loss

To reduce the domain gap between the 2D image and 3D point-cloud representations, an optional cross-modal contrastive loss is applied, inspired by symmetric InfoNCE [61]. It pulls the two embeddings of the same tree together and pushes apart embeddings of different trees, encouraging both encoders to project into a shared latent space. The intent of this addition is twofold. First, it provides a means of cross-modal interaction during training without adding a dedicated fusion module: by aligning each tree's aerial and point-cloud embeddings, both encoders are pushed towards a shared, modality-invariant representation in which the two modalities can reinforce each other. Second, the loss is supervised purely by the image-point pairing and never sees a species label, so it contributes a gradient to every tree regardless of how many examples its species has. This is potentially beneficial to species in the long tail, as the regular cross-entropy term is dominated by the majority classes. This hypothesis is supported by the finding that self-supervised and contrastive objectives yield representations that are more robust under heavy class imbalance than purely supervised ones [94, 49].

Given a batch N of synchronised samples, we let the L_2-normalised feature embeddings be A_i for the aerial image of the i-th tree and P_j for the point cloud of the j-th tree. Due to the L_2-normalisation, the cosine similarity $s_{i,j}$ between any A_i and P_j, scaled by inverse temperature γ, is then given by their dot product:

$$s_{i,j} = \gamma(A_i \cdot P_j)$$

The objective is to maximise the similarity of the positive pair $s_{i,i}$ – the image and point cloud of the same tree – while minimising the similarity of all negative pairs $s_{i,j}$ – images and point clouds of different trees. This loss is then symmetrically calculated for both directions:

$$\mathcal{L}_{a\to p} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(s_{i,i})}{\sum_{j=1}^{N}\exp(s_{i,j})}$$
$$\mathcal{L}_{p\to a} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(s_{i,i})}{\sum_{j=1}^{N}\exp(s_{j,i})}$$
$$\mathcal{L}_{\text{align}} = \frac{1}{2}(\mathcal{L}_{a\to p} + \mathcal{L}_{p\to a})$$

3.5 Taxonomy-Aware Learning and Inference

3.5.1 Multi-Level Auxiliary Classifier

A complementary way to inject taxonomic structure is to supervise the shared representation at several taxonomic ranks simultaneously. In addition to the species head, we attach a lightweight linear classifier to each of a set of coarser levels $\mathcal{L}$ (by default genus, family and order). Crucially, every auxiliary head reads the same fused embedding $\mathbf{f} \in \mathbb{R}^D$ that the species head consumes, so the auxiliary gradients pull the encoder towards features that are simultaneously discriminative at multiple ranks rather than introducing a separate representation per level.

For a sample of species y, let $a_\ell(y)$ denote its ancestor taxon at level ℓ (undefined for species whose lineage lacks that level, which are ignored at that level). Each head ℓ produces logits $\mathbf{z}_\ell = \mathbf{W}_\ell\mathbf{f}$, and the auxiliary objective is the mean cross-entropy across levels,

$$\mathcal{L}_{\text{hier}} = \frac{1}{|\mathcal{L}|}\sum_{\ell\in\mathcal{L}}\text{CE}\big(\mathbf{z}_\ell,\, a_\ell(y)\big), \tag{3.2}$$

which is added to the main species classification loss with weight λ_{hier},

$$\mathcal{L} = \mathcal{L}_{\text{species}} + \lambda_{\text{hier}}\,\mathcal{L}_{\text{hier}}. \tag{3.3}$$

The auxiliary heads are used only during training and are discarded at inference, where the species head remains the sole predictor. Because a coarse head pools many species into one target – a family head, for instance, sees the entire co-family pool – rare species receive a stronger and more frequent gradient signal through their abundant relatives than they would from the species head alone [8].

3.5.2 Taxonomic Label Smoothing

This mechanism requires an explicit notion of how close two species are. Ordering the taxonomic levels from finest to coarsest (genus, subfamily, family, order, category, class, phylum, kingdom), we define the distance between two species as the depth of their lowest common ancestor: two species in the same genus have distance 1, two in the same subfamily distance 2, and so on up to distance 8 for species sharing only the kingdom; species with no shared ancestor are assigned the maximum distance. We collect these into a symmetric matrix $\mathbf{D} \in [0,1]^{C\times C}$, normalised by the maximum distance, and define the corresponding similarity as $\mathbf{S} = 1 - \mathbf{D}$.

Rather than adding a taxonomic distance-based penalty to the loss, we incorporate the taxonomy on the target side. For a sample of true class y, the one-hot target is replaced by a soft target that retains $(1-\alpha)$ of its mass on y and redistributes the remaining α across all classes in proportion to their taxonomic similarity to y:

$$\tilde{y}_c = (1-\alpha)\nVdash[c = y] + \alpha\,\frac{\mathbf{S}_{y,c}}{\sum_{c'=1}^{C}\mathbf{S}_{y,c'}}. \tag{3.4}$$

The loss is the cross-entropy between this soft target and the predicted distribution:

$$\mathcal{L}_{\text{tax}} = -\sum_{c=1}^{C}\tilde{y}_c \log \hat{y}_c, \tag{3.5}$$

Because $\mathbf{S}$ decreases with taxonomic distance, same-genus siblings absorb most of the redistributed mass while progressively more distant taxa receive proportionally less. Setting $\alpha = 0$ recovers standard cross-entropy. This is a pure target-side operation: it requires no additional classifier heads and introduces no gradient conflict between taxonomic levels, while still penalising distant mistakes more heavily than close ones.

3.5.3 Taxonomy Lineage Embeddings

As a second, representation-side mechanism, we regularise the classifier weights towards a fixed embedding of the taxonomy itself. We build a graph whose nodes are the species together with their genus, subfamily, family and order ancestors, with edges connecting each taxon to its parent. Each node is assigned a unique fixed unit vector, and the embedding of a species is the L_2-normalised sum of the vectors along its lineage – the species node together with all of its ancestors. Because the node codes are near-orthogonal in high dimension, the inner product between two species embeddings grows with the number of ancestors they share: it is largest for species of the same genus, smaller for species that meet only at the family or order, and approximately zero across different orders. The embedding is thus a projection encoding of the multi-hot taxonomic lineage whose geometry reflects taxonomic proximity, and it is held fixed throughout training. Let $\mathbf{g}_c$ be this L_2-normalised taxonomic embedding of species c and $\mathbf{w}_c$ the L_2-normalised weight vector of the final classification layer for class c. We add a cosine-alignment penalty

$$\mathcal{L}_{\text{prior}} = \frac{1}{C}\sum_{c=1}^{C}\big(1 - \mathbf{w}_c \cdot \mathbf{g}_c\big), \tag{3.6}$$

weighted by λ_{tax}, which pulls each class prototype towards its fixed taxonomic embedding. Because $\mathbf{g}_c$ carries no learnable parameters it cannot adapt to the classifier weights; the penalty can only be reduced by moving the prototypes, so the prototypes of taxonomically close species are drawn together and structure is shared between head and tail classes.

3.5.4 Graceful Coarsening at Inference

Following Park and Kwon [62], we additionally allow predictions to be coarsened at inference. Given a confidence threshold θ, a sample is assigned its species-level prediction if the softmax confidence is at least θ; otherwise the prediction is degraded to the parent taxon, whose confidence is obtained by summing the probabilities of all species beneath it. This is repeated up the taxonomy until the confidence exceeds θ or the root is reached. Coarsening trades specificity for correctness, reducing the severity of mistakes on uncertain samples while leaving confident species-level predictions untouched.

3.6 Optimisation, Scheduling and Constraints

All models were trained using an AdamW optimiser [53]. An initial learning rate of 10^{-3} was selected, with a cosine annealing learning rate scheduler with a minimum value of 10^{-6}. Batch size was set to the maximum size that fit in GPU memory for the most demanding architecture, which was 128. The models were constrained using L_2-weighted regularisation to reduce the tendency of the models to overfit on the training data, with a factor of 0.01. All models were trained for 30 epochs, unless mentioned otherwise. For each training run, the data was split into a 0.8/0.1/0.1 split for training, validation and testing data respectively, regardless of the size of the used subset.

3.7 Hardware and Software

All major experiments were run on a high-performance computing cluster using a single NVIDIA A100 GPU and Intel Xeon Platinum 8360Y CPUs. Initial testing of the models and smaller experiments were run on a machine with a NVIDIA GeForce GTX 1080 Ti GPU and an Intel Core i7-8700K CPU. All code is written and executed in Python 3.11.14, using PyTorch 2.9.1 and PyTorch Lightning 2.6.1, running CUDA 12.6.80.

3.8 Performance Metrics

The performance of our classifier is evaluated using the following metrics, all expressed in terms of a common set of quantities. Let N be the total number of test samples, C the number of classes, $y_i \in \{1, \dots, C\}$ the ground-truth label of sample i and $\hat{y}_i$ the model's prediction. For each class c, TP_c, FP_c and FN_c denote the number of true positives, false positives and false negatives, and $\mathbb{1}[\cdot]$ is the indicator function.

- **Overall accuracy** (OA): the fraction of samples whose predicted class matches the truth.

$$\mathrm{OA} = \frac{1}{N} \sum_{c=1}^{C} TP_c$$

- **Balanced accuracy** (BA): the arithmetic mean of per-class recall, which up-weights performance on minority classes.

$$\mathrm{BA} = \frac{1}{C} \sum_{c=1}^{C} \frac{TP_c}{TP_c + FN_c}$$

- **Matthews Correlation Coefficient** (MCC): a balanced correlation measure between predictions and labels that remains informative under class imbalance. With $T = \sum_{c=1}^{C} TP_c$ the total number of correct predictions, $t_c = \sum_{i=1}^{N} \mathbb{1}[y_i = c]$ the support of class c and $p_c = \sum_{i=1}^{N} \mathbb{1}[\hat{y}_i = c]$ the number of times class c was predicted.

$$\mathrm{MCC} = \frac{T \cdot N - \sum_{c=1}^{C} t_c \cdot p_c}{\sqrt{\left(N^2 - \sum_{c=1}^{C} t_c^2\right)\left(N^2 - \sum_{c=1}^{C} p_c^2\right)}}$$

- **Correctly Predicted Bits** (CPB) [89]: a taxonomy-aware metric that credits a prediction when it lies on the correct branch of the taxonomy and rewards it more for being more specific. Let $\mathcal{T}$ denote the full taxonomy tree, $\mathcal{T}_v$ the subtree rooted at node v, $|\text{Leaves}(\mathcal{T}_v)|$ the number of species leaves under v, and $\mathcal{A}(y_i)$ the set of ancestors of y_i (including y_i itself). Then

$$\text{CPB} = \frac{1}{N}\sum_{i=1}^{N} \nVdash[\hat{y}_i \in \mathcal{A}(y_i)] \left(1 - \frac{|\text{Leaves}(\mathcal{T}_{\hat{y}_i})|}{|\text{Leaves}(\mathcal{T})|}\right).$$

Note that $\hat{y}_i$ may be a prediction at any taxonomic level, since species-level predictions can be coarsened to a higher taxon when confidence falls below a threshold. This metric is added to use when the number of output classes change, when coarsening predictions or predicting across multiple hierarchical levels.

In addition to these quantitative metrics, a qualitative analysis is performed to identify and explain further strong points and limitations of the current methods.

Chapter 4

Experiments and Results

We conducted multiple experiments to investigate ways of improving classifying tree species. Most exploratory testing and tuning was done on a fixed $\sim 10\%$ subset (175,987 trees, 200 classes) of all labelled trees due to compute budget restrictions. The best module and parameter setup for these smaller experiments is then selected for a run on the entire, countrywide dataset. Results from subset testing and the experiments on the full dataset will be presented in the following sections.

In an assessment on modality, we trained and evaluated classification performance between models using only aerial images as input, only point clouds as input, and compared those to multiple methods of combining the modalities. These modalities were fused using 1) concatenation of learned feature vectors of both modalities, 2) contrastive learning to create a shared latent feature space between the modalities and 3) applying cross-attention between the unpooled learned features of both modalities, where aerial tokens act as keys to the point tokens queries and values, and vice versa.

In addition, we compare the impact of multiple taxonomy-based additions by ablation. The first of these additions is the use of smoothed labels, based on the taxonomic distance between the predicted and target species. This taxonomic distance is defined as the number of steps taken in the taxonomy to reach the lowest level shared ancestor. Furthermore, a model using auxiliary classifier heads for multiple levels of the hierarchy is tested, as well as using a GNN embedding as a prior for the models. Finally, we also present the results of implementing graceful coarsening at inference, and we compare the performance across all levels of the taxonomy.

4.1 Comparison with Past Results

The baseline we compare our implementation to is the current best tree species model of NEO [74]. This model was trained only on cut out tree point clouds (using PointNet++) or 2D-projections thereof (using an MVCNN). The best MVCNN implementation reached a balanced accuracy of 0.35 on species level, while the PointNet++ reached 0.25, but outperformed the MVCNN on some of the higher taxonomic levels.

Important to mention is that these results were obtained on a small subset of only 33 classes of the total classes. What complicates direct reproductions, is that not all data processing and selection steps are reproducible exactly from the description, and additionally, the dataset has changed since the last implementation. For a fair direct comparison, we approach the dataset selection of the previous implementation by using similar exclusion criteria for the point cloud size: a minimum and maximum of 1,000 and 10,000 was used. Furthermore, we match the number of classes by selecting the 33 majority classes that remain after this filtering, to reach a similar distribution as in the previous implementation. As the exact number of trees used was not described clearly, keeping the point cloud sizes and number of classes equal allows us to still make a relatively fair comparison between our implementation and the previous one.

With these settings, our implementation reaches a balanced accuracy of 46%, which is a 11 percentage points improvement on balanced accuracy over the best previous implementation, albeit on a different dataset. Although we use the comparison with the past implementation as a baseline measure, most of the following investigations will focus on improving performance on classifying a larger number of classes, as this was the primary goal of the current investigation.

This leads to the results on a final run on the complete dataset, containing all municipalities. For this, we used the best performing model, which was the fusion model with cross-modal contrastive alignment (see Section 4.4 for justification). Furthermore, the filtering of the data was kept as described in Section 3.2, and batch size, length of training and other setting were kept as described in Section 3.6. On the complete dataset we reach an accuracy of 0.80, a balanced accuracy of 0.32 and an MCC score of 0.79. More detailed results per municipalities and species can be found in Appendix G and Appendix H.

The version of the model trained on the complete dataset will be used for any inference-level changes, such as the aforementioned graceful coarsening and analysis across taxonomic levels (Section 4.5), as well as any post-hoc imbalance mitigations (Section 4.6). Other experiments were performed on a smaller subset of $\sim 10\%$ of the dataset.

4.2 Comparison of Architectural Backbones

A comparison of the efficacy of different architectures for encoding the images and point clouds is shown in Table 4.1. Contrary to expectations, the smaller and standard architectures matched or outperformed the larger and transformer-based models. Among the convolutional and native point-cloud encoders the accuracies fall in a narrow band, with the larger ResNet-50 showing a wider train–validation gap (Appendix I) without any accuracy gain over ResNet-18, indicating overfitting rather than added capacity being useful at this dataset size. The transformer-based models (Swin and Erwin), by contrast, failed to converge, staying near chance level – consistent with the well-known data-hunger of attention mechanisms relative to the size of our labelled set. The smaller ResNet-18 and PointNet++ were also cheaper and faster to train. Given this, and the compute and time restrictions, ResNet-18 and PointNet++ were adopted as the encoders for the final model and used for most larger-scale testing, and we forgo further investigation into the transformer variants.

Backbone	Parameter Size	Accuracy	Balanced Accuracy	MCC
ResNet-18	11.2 M	**0.60**	**0.23**	**0.58**
ResNet-50	23.5 M	0.59	0.20	0.57
Swin-tiny	27.5 M	0.10	0.01	0.00
Mobilenet_small_v3	927 K	0.59	0.21	0.57
PointNet++	807 K	**0.64**	**0.26**	**0.62**
PointNext	3.3 M	0.62	0.25	0.61
Erwin	11.9 M	0.16	0.02	0.10
MV-CNN	11.2 M	0.59	0.22	0.57

Table 4.1: Comparison of aerial image and point cloud encoder backbones.

4.3 Uni- and Multimodal Classification

A comparison between models trained on different data modalities can be found in Table 4.2. For the individual loss, accuracy and balanced accuracy curves, see Appendix I. We can see that using a model on point clouds slightly outperforms a model on aerial images only, the point cloud model having an accuracy and balanced accuracy of 0.64 and 0.26 respectively, as opposed to 0.60 and 0.23 for the aerial model. The baseline fusion model, using feature concatenation, outperforms both unimodal models, reaching 0.71 accuracy and 0.30 balanced accuracy. Both modalities are therefore individually informative and their combination improves over either alone (+0.04 balanced accuracy over the point-cloud model). This motivates the comparison of fusion strategies in the following section.

Model	Accuracy	Balanced Accuracy	MCC	Number of Classes	Dataset Size
Aerial-only	0.60	0.23	0.57	200	175,987
Point clouds-only	0.64	0.26	0.62	200	175,987
Fusion (concatenated)	**0.71**	**0.30**	**0.70**	200	175,987

Table 4.2: Performance of models trained on separate modalities on the new dataset. Cutouts and point clouds were based on AHN5/6 or AHN4, depending on the province, trees were identified by stem, so that multiple trees within the same crown were labelled and classified as different trees.

4.4 Comparison of Fusion Methods

We compare three methods of fusing the aerial imagery and LiDAR point clouds. In all three, the features are extracted by separate image and point-cloud encoders, with the training setup and parameters kept equal. As shown in Table 4.3, every fusion method outperforms the unimodal models. Of the three, only the contrastive loss improves on plain concatenation, raising accuracy by 4 and balanced accuracy by 7 percentage points; relative to the best unimodal model it gains 11 points on both metrics. Adding the self- and cross-attention modules, by contrast, slightly lowers performance compared to concatenation alone. The cross-modal contrastive variant thus gives the best overall performance and is kept for all subsequent experiments; the relative ordering of the three strategies is interpreted in Section 5.1.

Fusion Type	Accuracy	Balanced Accuracy	MCC	Dataset Size (Number of Classes)
Concatenation	0.71	0.30	0.70	175,987 (200)
Contrastive	**0.75**	**0.37**	**0.74**	175,987 (200)
Cross-attention	0.70	0.28	0.69	175,987 (200)

Table 4.3: Performance comparison between different feature fusion methods.

4.5 Inclusion of Taxonomic Information

Under the assumption that species close together in the taxonomy should be more similar, we tested whether taxonomic relations could aid in the training process in decreasing amount and severity of misclassifications made. The following methods of including taxonomic information were tried: using a multi-head classifier for multiple levels of the taxonomy (Section 3.5.1), embedding the taxonomic lineages as multi-hot vectors (Section 3.5.3) and using label smoothing based on sibling species (Section 3.5.2). A minimal hyperparameter sweep was done for each of the ablations (see Appendix J), to direct us towards a semi-optimal value for each of the ablations on a 1% subset of the data. These semi-optimal values were then used for the ablations on a 10% subset of the data, to provide a better estimate of the impact of each addition. All additions are applied to the contrastive loss fusion model, and any performance gains and losses are compared to this model as a baseline and shown in Table 4.4.

Addition	Δ Accuracy	Δ Balanced Accuracy
Multi-head classifier	-0.01	-0.01
Lineage embeddings	+0.01	-0.02
Label smoothing	+0.00	-0.05

Table 4.4: Performance difference obtained applying different methods of including taxonomic information during model training.

We can see that none of the taxonomic implementations lead to improvements on the minority species, as can be seen from the balanced accuracy, all even lead to slight decreases. Improvement on overall accuracy is also minimal, if present. This suggests that including the taxonomy does not provide any extra information for the model, whether it be on the architectural, loss or target side of the training process. Its use for inference will be shown in the following paragraph.

4.5.1 Graceful Coarsening Using the Taxonomy

We use a similar confidence-based graceful coarsening as used by Park and Kwon [62]. Given a confidence threshold θ, the model predicts the species if the prediction confidence is equal or higher to this threshold θ, else it predicts the genus. If for the genus prediction the confidence is still below the threshold, we fall back to the level above, until either the threshold is passed or we reach the highest level.

The use of such a graceful coarsening is supported by the fact that the prediction confidence is quite strongly correlated with the correctness of the prediction. If this correlation was perfect, only incorrectly classified predictions would be coarsened, and correct predictions would be left at the finest label level. However, this is not the case, and thus some trade-off needs to be made to coarsen as many incorrect predictions while keeping the highest amount of correct predictions as specific as possible. In the left panel of Figure 4.1 an optimal threshold value $\theta^* = 0.91$ is found based on the highest relative gain of coarsening labels based on confidence: a higher threshold would lose label specificity relative to the accuracy gain, a lower threshold does not reach the maximum possible accuracy gain relative to label specificity.

The right panel of Figure 4.1 shows the impact of different threshold values on accuracy, balanced accuracy and CPB, against the coverage of labels kept as species. Using the threshold $\theta^* = 0.91$, accuracy increases by 14.6 percentage points (to 94.5%), while balanced accuracy by 18.4 percentage points (to 50.9%), while CPB increases by 3.3 percentage points (to 82.9%). Alternatively, a threshold of $\theta^*_{CPB} = 0.81$ leads to the highest CPB value of 83.7%. This is a theoretical optimum of label specificity versus accuracy.

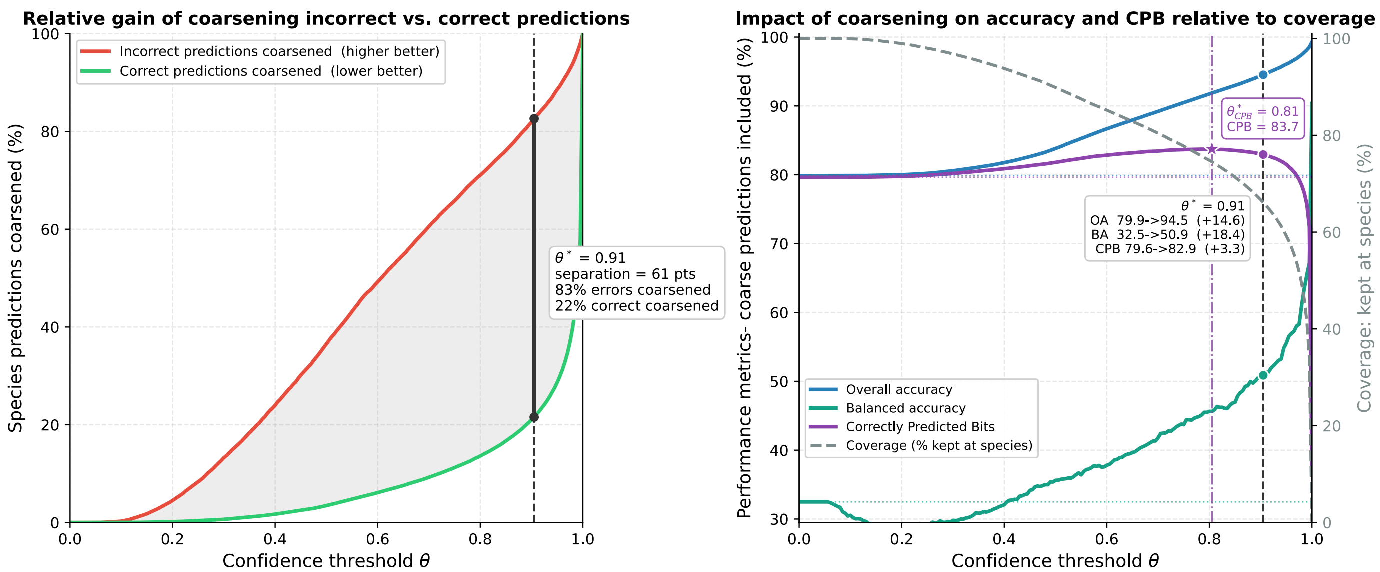


Figure 4.1: Left: the optimal confidence threshold found based on the highest gain of improving incorrect labels, relative to the amount of correct predictions that are coarsened incorrectly (loss of specificity). Right: the payoff of coarsening labels on accuracy, balanced accuracy and CPB metrics. The optimal confidence threshold θ^* from the left plot is highlighted, as well as a potential threshold where the CPB is highest.

4.5.2 Performance per Taxonomic Level

Besides gracefully coarsening the predictions to improve classification accuracy, performance on each level of the taxonomy was also analysed, these results can be found in Table 4.5. We see that the performance does increase for higher levels of the taxonomy. This result also holds even when considering the increasing chance of randomly guessing correctly on higher levels – there are fewer classes for higher levels – as can be seen in the MCC score. The Correctly Predicted Bits (CPB) metric shows an interesting trend here; it is highest for the genus level. This would mean that according to this metric, it would be better to predict at the genus level as opposed to the species level, and thus the drop in label specificity does not matter as much as the increase in classification accuracy. The CPB is quite low for most higher levels in the taxonomy, as the maximum entropy of the label distribution at those higher levels is low: there is a large label imbalance and there are very few classes, so the relative value of getting the majority class correct is low.

Level	Accuracy	Balanced accuracy	MCC	CPB	# classes
Species	0.80	0.32	0.79	0.79	345
Genus	0.86	0.44	0.84	0.83	91
Subfamily	0.84	0.54	0.79	0.77	11
Family	0.88	0.51	0.86	0.82	41
Order	0.89	0.58	0.86	0.78	22
Category	0.93	0.68	0.82	0.56	9
Class	0.99	0.80	0.82	0.16	3
Phylum	0.99	0.80	0.82	0.16	3

Table 4.5: Performance across all levels of the taxonomy. Results based on a fusion model trained on all municipalities. Note the lower number of classes for subfamily, as many lineages do not contain a subfamily. Also note that not all five phyla are there, as trees from the two others are filtered out due to their point cloud size.

4.6 Imbalance Mitigation

4.6.1 Logit Adjustment

Logit adjustment subtracts the log class-prior from the logits, scaled by a temperature τ. Table 4.6 sweeps τ, showing the trade-off it induces between head and tail accuracy. The head is binned by species that occur more than a 1,000 times, while the tail is binned by species appearing less than 50 times; the middle bin contains the species in between. With increasing τ, the binned tail accuracy increases, while the head accuracy drops quite drastically. Regarding balanced accuracy, the best results come from using $\tau = 1.0$, leading a 15 percentage point increase at the cost of an 8 percentage point drop in accuracy. For values higher than $\tau = 1.0$, the overall accuracy drops quite drastically, and the balanced accuracy decreases as well, although the accuracy for the very rare classes continues to increase.

4.6.2 Frequency-Floor Sensitivity

The minimum-sample threshold used during dataset construction trades the number of retained species against how well the tail can be learned. Table 4.7 reports this sensitivity: lowering the threshold retains more rare species but reduces tail recall. Note that the tree count is held constant across floor values: for each floor a stratified subsample of 175,987 trees was drawn so that all runs see the same amount of data, which is why the number of classes — not the number of trees — varies with the floor. A different subset had

τ	Accuracy	Balanced Accuracy	Head Accuracy	Middle Accuracy	Tail Accuracy
0.0	**0.80**	0.32	**0.82**	0.47	0.17
0.5	0.79	0.42	0.80	0.53	0.25
1.0	0.72	**0.47**	0.73	**0.55**	0.35
1.5	0.51	0.46	0.55	0.52	**0.39**
2.0	0.24	0.41	0.26	0.44	**0.39**

Table 4.6: Effect of the logit-adjustment temperature τ on balanced and per-bin accuracy.

to be used to keep the same amount of trees. In the final model a choice was made to keep the frequency floor at 10, to retain as many species as possible with the number of cross-validation folds. We return to the implications of the trade-off between species coverage and tail performance in Section 5.1.

Frequency floor	Accuracy	Balanced accuracy	# classes	# trees
10	0.71	0.22	345	175,987
100	0.70	0.33	176	175,987
1000	0.73	0.53	82	175,987

Table 4.7: Sensitivity of the dataset to the species frequency floor value.

4.7 Out-of-Distribution Generalisation

Two sets of held-out testing data were used to measure out-of-distribution performance. For the first, three municipalities were kept separate (Leeuwarden, Papendrecht and Utrecht), and the model was tested on these three held-out municipalities to test whether the classification performance was not partially influenced by confounding variables in municipalities. Images and point clouds collected in different parts of the country were collected at different points in time, under different lighting and weather conditions. Soil types also differ between regions, and can influence the visual appearance of a species. This means that an image of oak in Amsterdam may look slightly different than an oak in Groningen, due to variations in environmental and visual artefacts. By excluding some municipalities from the training data, we tested whether these effects are present. The held-out municipalities thus show a clear performance drop (see Table 4.8), confirming a measurable distribution shift across regions.

Test set	Accuracy	Balanced accuracy	MCC	# classes
In distribution	0.75	0.37	0.74	200
Held-out	0.56	0.19	0.53	200

Table 4.8: Results on out-of-distribution municipalities testing of the trained model.

The second held-out test set contained a set of tree species that were not present at all in the training set (see Table 4.9). This would trivially mean that the model cannot predict these species correctly. However, if the taxonomic information has been learned well enough, we expect the model to be able to predict the higher level taxa of the trees reasonably well. For example, if a *green oak* was not in the training set, but other oak species were, the model should be able to classify a green oak as a related species.

The results in Table 4.9 show how the model classifies out-of-distribution species. As expected, accuracy at the species level is zero, but already at the genus-level the model recovers a substantial part of the correct labels (0.32 accuracy at genus). For higher levels in the taxonomy, accuracy and balanced accuracy increase even further. The class and phylum levels reach very high accuracy (0.98 and 0.97), but here only two classes remain, and the lower balanced accuracy (0.70) and MCC (0.53) indicate that this is inflated by the majority

Level	Accuracy	Balanced accuracy	MCC	# classes
Species	0.00	0.00	0.00	100
Genus	0.32	0.17	0.29	52
Subfamily	0.40	0.21	0.32	5
Family	0.38	0.19	0.32	29
Order	0.47	0.23	0.35	17
Category	0.73	0.36	0.42	6
Class	0.98	0.70	0.53	2
Phylum	0.97	0.70	0.53	2

Table 4.9: Results on 100 out-of-distribution species testing of the trained model.

class rather than reflecting equally strong performance on both. Even without having seen these species, the model thus assigns them to progressively more accurate taxa as the hierarchy coarsens.

4.8 Qualitative Analysis of Results

One of the more salient results, is that the current method seems to underperform on trees in parks and other areas where they are more tightly spaced together. An example of this can be seen in Figure 4.2, that shows a section of the Vondelpark in Amsterdam, where occlusion occurs and tree species are more varied and more rare. Lane and street trees on the other hand, seem to be classified quite accurately on average. Figure 4.3 shows a section of the *Amsterdamse grachtengordel* (Amsterdam canal belt), where there is a low diversity of species and the individual trees are spaced out well.

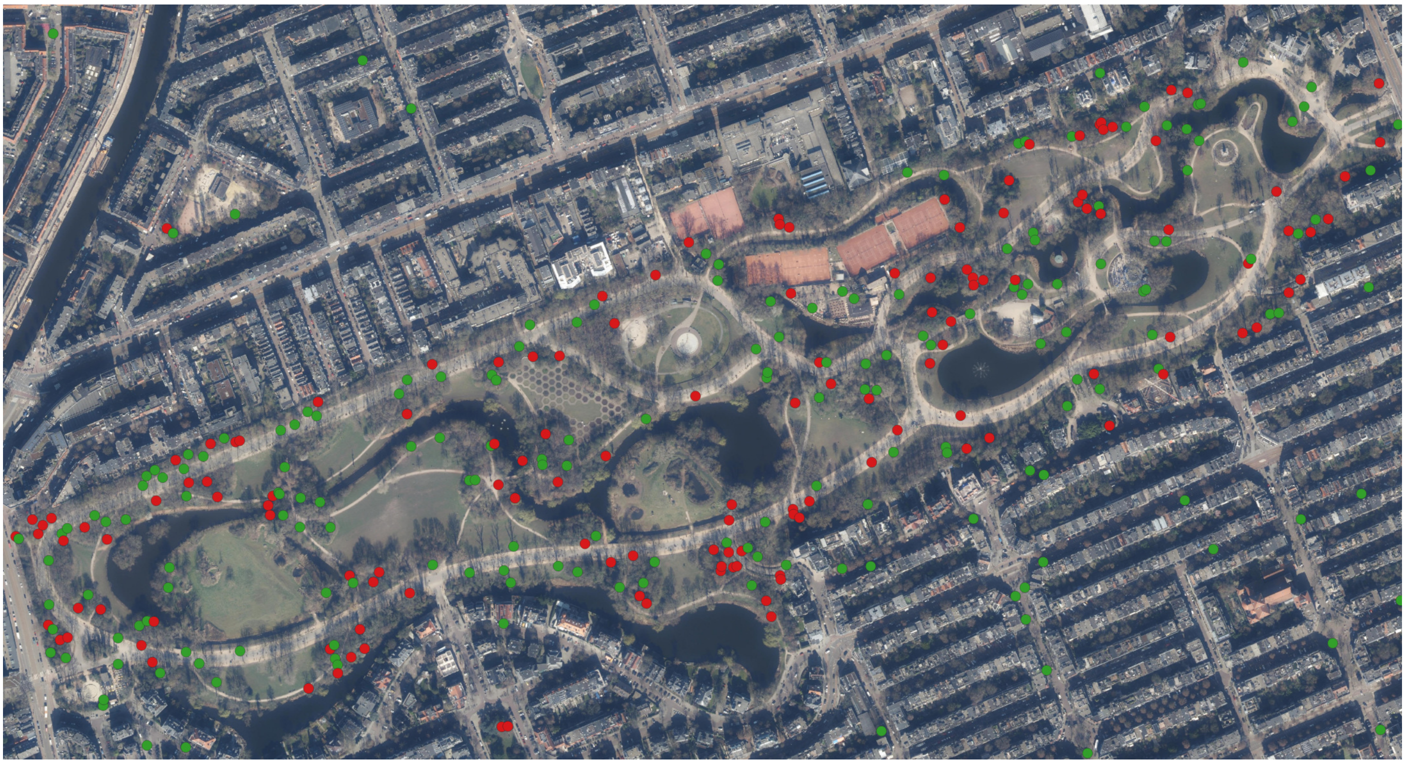


Figure 4.2: A section of the Vondelpark in Amsterdam, where correctly predicted species are marked with a green dot, while incorrectly predicted species are marked red.

Two reasons were identified for this: trees planted on lanes are often more homogeneous in species, and they are also more common in our labelled dataset. Park trees are generally more varied, and more occlusion

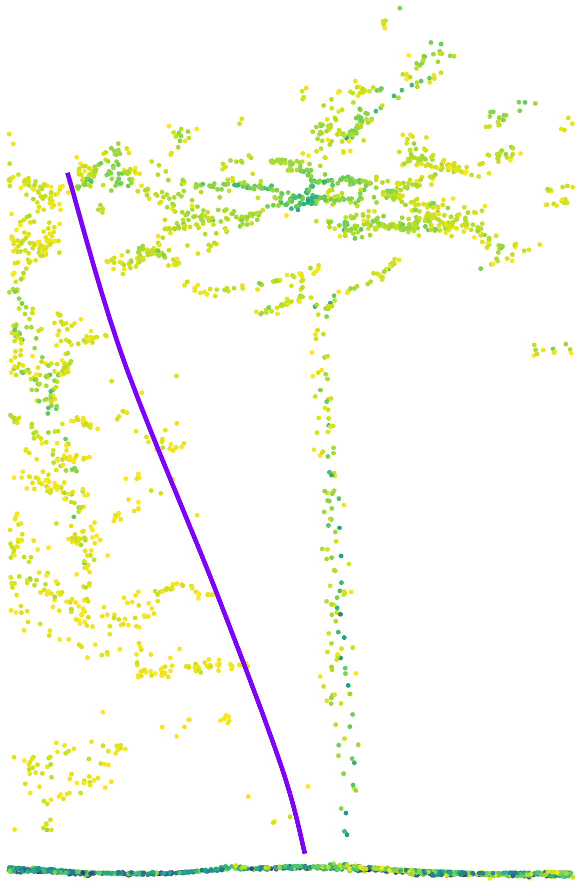
Figure 4.3: A section of the canal belt in Amsterdam, where correctly predicted species are marked with a green dot, while incorrectly predicted species are marked red.

occurs as trees grow closer to each other, at times overlapping. This can impact both the aerial images, as the smaller tree might be badly or not visible, and the point clouds, as the individual tree splitting algorithm may include part of the larger tree in the point cloud of the smaller tree, as the trees are split based on the top-down view and the stem locations. Overhanging branches could be combined with a smaller tree, which can confuse the model as the point cloud will contain structural information of multiple, potentially different species.

One theoretical advantage of using multimodal data, is that aerial images and point clouds of trees do not suffer from the same noise. If one of the modalities is noisy, the other may still clearly display a tree. For aerial images, smaller trees could be occluded by the canopy of a larger tree, while the small tree could still be visible as a point cloud. On the other hand, undergrowth such as bushes can introduce noise in point clouds, while this undergrowth may not be visible from the viewpoint of the aerial image. We see this in the data, as there is a significant amount of trees that is difficult to see on the aerial images, especially if they are smaller.

This could partially explain why the aerial-only model performs worse than the points-only models. For some images, no tree at all is visible, and the tree polygon seems to be misplaced (see Figure 4.4a). In this case, it is caused by the polygon splitting algorithm that split a smaller tree, and assigned the overhanging branches of the larger tree to the smaller tree. We somewhat combat this issue by adding some background to the images, so slightly misplaced trees may still be included in the image, and by filtering on the number of points in the point cloud, so that if a tree is not correctly extracted, it will be excluded. Nevertheless, many trees remain that are difficult to distinguish or have multiple trees in the image. Despite the individual tree cutout method, there are also trees that still consist of multiple trees (see Figure 4.4b).

(a) An example of an aerial image where the polygon does not match most of the location of the tree, rather it mostly covers the road. The long straight edge is the result of the tree splitting, as this tree crown had multiple stems underneath.



(b) An example of a point cloud of a different tree where multiple trees can clearly be seen (separated by the purple line), suggesting this consists of two trees.

Figure 4.4: Visual examples of erroneously sampled tree aerial image and point cloud.

Chapter 5

Discussion

5.1 Research Objectives

The primary aim of this research, was to be able to scale up the tree species classification model to be able to run on a country-wide basis, on a great variety of species. Although the current methods do not succeed entirely in creating an unbiased classifier that solves the long-tailed distribution of species, it does improve upon the previous method, and rivals past performance on a much larger scale (345 classes compared to 33 classes). The model can classify reasonably well reaching 80% accuracy and 32% balanced accuracy before the application of data imbalance mitigations. Through the graceful coarsening mechanism, it can give the most specific answer it is confident in, or instead fall back to a coarser, higher-level prediction when unsure, which increases accuracy and balanced accuracy to 95% and 51% respectively.

Modality Fusion

A significant part of the improvement in classification performance was due to the ability of the network to use both structural and spectral information in its decision. Through feature fusion, the network learns to distinguish trees based on both their RGB and NIR aerial images, as well as their LiDAR scanned point clouds. Having compared multiple methods of fusion, we find that interaction through a contrastive loss function that aligns the aerial images and point clouds of each tree yielded the largest improvement: +0.11 on both accuracy and balanced accuracy compared to the best unimodal model. We attribute this to two properties of the contrastive loss. First, it shapes the shared embedding space rather than adding a large interaction module of its own, improving fusion without the additional capacity that small labelled datasets struggle to constrain. Second, by pulling each tree's aerial image and point cloud together while pushing apart those of other trees, it forces both encoders to retain the modality-invariant, species-discriminative structure—crown geometry, size, and spectral signature—while discarding modality-specific noise, acting as a regulariser toward features that generalise.

In contrast, and against our expectations, the cross-attention model did worse compared to the baseline fusion method despite appearing promising on initial tests. This can be explained by the well described phenomenon that architectures that make use of attention mechanisms require very large amounts of data before they surpass more constrained models, amounts that the current labelled dataset may not reach. Another issue with this method might be the mismatch in modality tokens. The initial idea was for the model to be able to allow for interactions amongst patches of the image with grouped points from the point cloud. However, as the aerial images only provide a view from the top of the trees, the spatial correspondence between the image patches and the point ball queries may be limited – as only certain points will correspond to what is visible in the images. In combination with the relatively small dataset, this can lead to a lot of the training being fitting to noise rather than to the actual signal.

The relatively modest improvement of the fusion module overall compared to some similar approaches in the literature, may be in part attributed to the fact that the aerial and point cloud data are not obtained at the same time. The aerial images are collected much more frequently and on a relatively small time-frame (within a season), while the LiDAR scans are made over a period of years, and trees may have changed during this period.

Including the Taxonomy

The earlier mentioned graceful coarsening mechanism based on the taxonomy can positively impact classification performance at the cost of losing some coverage on the finer labels. With our selected threshold, we increase performance by 15 percentage points, to 95% accuracy. Having an adjustable threshold allows users to choose how precise or accurate the predictions need to be according to their needs.

At training time however, including taxonomic information seemed to slightly hurt performance. The idea that information from common trees could be used to help predict their rare sibling species seems not to work with the tested methods. Taxonomic information was separately included on the target side though label smoothing, on the architectural side by using auxiliary prediction heads for each level and by including a taxonomic prior through embeddings. It appears that the taxonomic structure is not informative enough to truly help out the rarer species. In botanical literature we find support for this idea. Remote sensing data may be too coarse for the fine details on which taxonomic classification is often based for trees. Many are based on whether trees reproduce through seed or spore, on the shape of the leaves and seeds, the texture of the bark, and sometimes root system as well [43]. All these distinguishing characteristics are too small to be visible on aerial images or LiDAR point clouds of our resolution. Additionally, taxonomies in itself are somewhat subjective, as for example the validity of the *Ginkgophyta* class as whole is not agreed on [93] and even the definition of species is not universally accepted [43].

Data Imbalance

One of the set objectives was to see whether the impact of dataset imbalance could be weakened. First we investigated the option of raising the frequency floor of species below which they would be excluded from the training set. We see a strong relation between a higher frequency floor on the balanced accuracy: 0.22 for a floor of 10 occurrences per species, 0.53 for a floor of 1000. Part of this may be due to excluding some very rare species that do not have enough training signal to be properly learned, but the literature suggests that the ratio between the most common classes and most rare classes may also be important, not just the absolute occurrences per species [11]. This experiment supports this claim by showing this trend, but the trade-off of losing such a significant amount of species may be too costly for the ambitions of this application, as too many species would be excluded.

A more subtle mechanism to counter this imbalance is post-hoc logit adjustment, which re-weights the predictions by the class priors at inference time rather than discarding rare species from the training set altogether [56]. Sweeping the temperature τ confirmed that this directly trades head accuracy for tail accuracy: at $\tau = 1.0$ the balanced accuracy rose by 15 percentage points while overall accuracy dropped by 8, with the gains concentrated in the middle and tail bins. This makes logit adjustment complementary to the frequency floor: where raising the floor improves balanced accuracy by removing the hardest tail classes from the evaluation entirely, logit adjustment retains all species and instead shifts the decision boundary towards them, recovering tail recall without sacrificing taxonomic coverage. The appropriate choice therefore depends on the application: a single τ can be tuned after training to favour either overall accuracy on the common species or balanced performance across the full long tail, without retraining the model.

Out-of-Distribution Tests

The model seems to generalise quite well to OOD species (32% accurate at the genus level for unseen species only). It can therefore potentially be used in environments where tree species occur that have not been labelled, which will most likely be the case in areas not included in the municipalities comprising the current dataset. The OOD municipality test suggests that the intuition that there is an influence of source-specific confounding variables is true, as the accuracy dropped by 0.19, and the balanced accuracy by 0.18. This highlights the importance of including a variety of regions in the input data of the model, as this would reduce bias towards confounds due to lighting conditions, differences in acquisition time and geographical influences (e.g. soil type). This effect could be mitigated by normalising lighting values over sources, but we leave the effectiveness of this to future research.

5.2 Limitations and Future Research

The current model is solely trained on images of individually cutout trees. With the recent rise of foundation models, it seems reasonable to assume that using pretrained embeddings on general remote sensing data would help increase a classification model such as the one presented in this study. The immensely dense representations learned could help the model in distinguishing between species after fine-tuning on our particular examples. Geospatial foundation models already exist, and some examples already show promise in land cover classification [34] and species recognition [77, 21, 4], so a similar downstream task of individual species classification seems to have potential, which could be investigated in future research.

In addition, we only used manually labelled trees as training input, gathered mostly in urban environments. Due to this, our model is biased towards urban tree species (which can also be seen in the drop in performance in densely vegetated areas such as parks). It may be worthwhile to include more forested areas in the input of the model, to see if the model can be used for more varied types of land cover. Self-supervised methods might be of great use here, as labelling trees in forested areas can be even more time-consuming than labelling urban trees. Moreover, self-supervised methods might be more robust to class imbalance [94, 49], so their advantages can be leveraged even more given the skewed distribution of species occurrences.

Another limitation is that the accuracy of the individual tree cutouts is not perfect, and this might influence the quality of the image or point cloud of the tree. Although we did try to mitigate this effect by excluding trees that had too few points in their point cloud (and were thus potentially wrongly segmented), some trees could still be off-centre or partially cut off, and the polygon bounding mask may be missing branches or including parts of the background. A solution to this problem that is supported in recent research is to use Segment Anything Models [42] for segmenting the trees. SAMs have been shown to be extremely capable of accurately segmenting trees, even in one-shot situations [79, 69]. Another idea is to apply 3D segmentation based on the tree point clouds; this way the problem of overhanging canopies could be mitigated by having separating margins that do not cut through vertically only, however, this would not solve the impact of overhanging trees for aerial images. Still, more accurate segmentation of the trees will lead to cleaner cutouts and clearer bounding boxes, and might greatly improve the accuracy on trees that are harder to locate and more occluded.

An already mentioned upside of using the available open-source datasets is that these are updated quite frequently, and older datasets can still be found. Another avenue of investigation would be to use multi-temporal data. By having datasets of different years, a model might become more robust to variations in tree size and lighting artefacts. For inference, data from multiple years can be used to inspect tree diversity over time, and municipalities can use it to analyse whether their planting interventions work as planned.

None of the implemented taxonomic modules were successful in improving classification performance. Part of this could be due to having too few visual similarities between sibling taxa on the scale of remote sensing data. Many taxonomic distinctions are made on very fine-grained features, which are not visible on aerial images and ALS LiDAR point clouds. The taxonomic information may therefore not be very informative for this type of data. Another idea would be to implement taxonomic lineages as embeddings similar to bag-of-words embeddings, where sibling species are kept closer together in embedding space. These embeddings can be learned or fixed, and even hyperbolic embeddings can be used [31], as taxonomies (and any tree-like structure) can be better represented in hyperbolic space as opposed to in Euclidean space. These embeddings can be combined with entailment cones [28, 31], such that each finer level in the hierarchy is contained in its parent region.

Chapter 6

Conclusion

In this work, we have presented experiments on the scalability of individual tree species classification on a country-wide basis. We have shown that classifying tree species using multimodal data is effective in leveraging more information from the trees and that decent performance can be obtained, reaching an accuracy of 80% and a balanced accuracy of 32% across the full species set. On a similarly sized subset as the previous best implementation of NEO, we reach an improvement of 11 percentage points on balanced accuracy. Performance is strongest on the most common species and on isolated, non-occluded trees. Due to the sheer number of species present, some of which occur only very rarely, the model still struggles to correctly predict the samples in the long tail of the species distribution.

Several of the imbalance-mitigation strategies we investigated trade performance on the head against performance on the tail. Logit adjustment improved balanced accuracy by 15 percentage points at the cost of 8 points of overall accuracy, while raising the frequency floor improved balanced accuracy by excluding the hardest rare species altogether. Depending on the goals of a particular application, each of these methods offers its own advantages. The taxonomic training modules, by contrast, did not improve species-level classification on any of the loss, architecture or target sides; taxonomy proved useful not for learning, but at inference time, where graceful coarsening allows the model to base its predictions on confidence which increases the usability of the model even when species-level predictions are imperfect.

Through these results we find support for the ambition of classifying tree species on a country-wide basis, at least one the size of the Netherlands, as they are promising especially when accounting for the small-sized architectures and compute restrictions used. Leveraging large pre-trained embeddings from foundation models, or using more potent encoding architectures, could scale these results to higher accuracy over more samples, especially once the untapped bulk of unlabelled trees can be used to self-supervise the model. Nevertheless, even light-weight applications show potential, and continued progress on the long-tailed distribution of species may allow such methods to fulfill this ambition.

Bibliography

[1] Hashem Akbari, Melvin Pomerantz, and Haider Taha. Cool surfaces and shade trees to reduce energy use and improve air quality in urban areas. *Solar energy*, 70(3):295–310, 2001.

[2] Arvid Axelsson, Eva Lindberg, and Håkan Olsson. Exploring multispectral als data for tree species classification. *Remote Sensing*, 10(2):183, 2018.

[3] Mattia Balestra, Suzanne Marselis, Temuulen Tsagaan Sankey, Carlos Cabo, Xinlian Liang, Martin Mokroš, Xi Peng, Arunima Singh, Krzysztof Stereńczak, Cedric Vega, et al. Lidar data fusion to improve forest attribute estimates: A review. *Current Forestry Reports*, 10(4):281–297, 2024.

[4] James GC Ball, Jana Annika Wicklein, Zhengpeng Feng, Jovana Knezevic, Sadiq Jaffer, Anil Madhavapeddy, Clement Atzberger, Michele Dalponte, and David Coomes. Geospatial foundation models enable data-efficient tree species mapping in temperate mountain forests. *bioRxiv*, pages 2026–02, 2026.

[5] K Paul Beckett, Peter Freer-Smith, and Gail Taylor. Effective tree species for local airquality management. *Arboriculture & Urban Forestry*, 26(1):12–19, 2000.

[6] K Paul Beckett, PH Freer-Smith, and Gail Taylor. Urban woodlands: their role in reducing the effects of particulate pollution. *Environmental pollution*, 99(3):347–360, 1998.

[7] Adam Berland, Sheri A Shiflett, William D Shuster, Ahjond S Garmestani, Haynes C Goddard, Dustin L Herrmann, and Matthew E Hopton. The role of trees in urban stormwater management. *Landscape and urban planning*, 162:167–177, 2017.

[8] Luca Bertinetto, Romain Mueller, Konstantinos Tertikas, Sina Samangooei, and Nicholas A Lord. Making better mistakes: Leveraging class hierarchies with deep networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12506–12515, 2020.

[9] Simon P Blomberg, Theodore Garland Jr, and Anthony R Ives. Testing for phylogenetic signal in comparative data: behavioral traits are more labile. *Evolution*, 57(4):717–745, 2003.

[10] Sebastian Briechle, Peter Krzystek, and George Vosselman. Silvi-net–a dual-cnn approach for combined classification of tree species and standing dead trees from remote sensing data. *International Journal of Applied Earth Observation and Geoinformation*, 98:102292, 2021.

[11] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural networks*, 106:249–259, 2018.

[12] E. B. Buis. Tree stem detection using low-density lidar-derived point clouds. Master's thesis, University of Amsterdam, 2024.

[13] Dongliang Chang, Kaiyue Pang, Yixiao Zheng, Zhanyu Ma, Yi-Zhe Song, and Jun Guo. Your" flamingo" is my" bird": Fine-grained, or not. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11476–11485, 2021.

[14] Jérôme Chave, Helene C Muller-Landau, Timothy R Baker, Tomás A Easdale, Hans ter Steege, and Campbell O Webb. Regional and phylogenetic variation of wood density across 2456 neotropical tree species. *Ecological applications*, 16(6):2356–2367, 2006.

[15] Jianchang Chen, Yiming Chen, and Zhengjun Liu. Classification of typical tree species in laser point cloud based on deep learning. *Remote Sensing*, 13(23):4750, 2021.

[16] Dengkai Chi, Jingli Yan, Kang Yu, Felix Morsdorf, and Ben Somers. Planting contexts affect urban tree species classification using airborne hyperspectral and lidar imagery. *Landscape and Urban Planning*, 257:105316, 2025.

[17] Moses Azong Cho, Renaud Mathieu, Gregory P Asner, Laven Naidoo, JAN Van Aardt, Abel Ramoelo, Pravesh Debba, Konrad Wessels, Russell Main, Izak PJ Smit, et al. Mapping tree species composition in south african savannas using an integrated airborne spectral and lidar system. *Remote Sensing of Environment*, 125:214–226, 2012.

[18] Clay. Clay foundation model. `https://madewithclay.org/`, 2025. Accessed: 22-01-2026.

[19] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9268–9277, 2019.

[20] Michele Dalponte, Lorenzo Bruzzone, and Damiano Gianelle. Fusion of hyperspectral and lidar remote sensing data for classification of complex forest areas. *IEEE Transactions on Geoscience and Remote Sensing*, 46(5):1416–1427, 2008.

[21] Kyle Doherty, Max A Gurinas, Erik Samsoe, Charles Casper, Beau G Larkin, Philip W Ramsey, Brandon Trabucco, and Russ Salakhutdinov. Leveraging foundation models for data-limited ecological applications. In *Neurips 2024 Workshop Foundation Models for Science: Progress, Opportunities, and Challenges*, 2024.

[22] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[23] Britta Eilmann and Andreas Rigling. Tree-growth analyses to estimate tree species' drought tolerance. *Tree physiology*, 32(2):178–187, 2012.

[24] EPPO. Eppo global database. https://gd.eppo.int, 2026. Accessed: 07-01-2026.

[25] Francisco J Escobedo and David J Nowak. Spatial heterogeneity and air pollution removal by an urban forest. *Landscape and urban planning*, 90(3-4):102–110, 2009.

[26] Ann-Catrin Fender, Dirk Gansert, Hermann F Jungkunst, Sabine Fiedler, Friderike Beyer, Klaus Schützenmeister, Björn Thiele, Kerttu Valtanen, Andrea Polle, and Christoph Leuschner. Root-induced tree species effects on the source/sink strength for greenhouse gases (ch4, n2o and co2) of a temperate deciduous forest soil. *Soil Biology and Biochemistry*, 57:587–597, 2013.

[27] Matheus Pinheiro Ferreira, Daniel Rodrigues dos Santos, Felipe Ferrari, Luiz Carlos Teixeira Coelho, Gabriela Barbosa Martins, and Raul Queiroz Feitosa. Improving urban tree species classification by deep-learning based fusion of digital aerial images and lidar. *Urban Forestry & Urban Greening*, 94:128240, 2024.

[28] Octavian Ganea, Gary Bécigneul, and Thomas Hofmann. Hyperbolic entailment cones for learning hierarchical embeddings. In *International conference on machine learning*, pages 1646–1655. PMLR, 2018.

[29] Haimiao Ge, Liguo Wang, Haizhu Pan, Yanzhong Liu, Cheng Li, Dan Lv, and Huiyu Ma. Cross attention-based multi-scale convolutional fusion network for hyperspectral and lidar joint classification. *Remote Sensing*, 16(21):4073, 2024.

[30] Fariborz Ghorbani, Yi-Chen Chen, Markus Hollaus, and Norbert Pfeifer. A robust and automatic algorithm for tls–als point cloud registration in forest environments based on tree locations. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 17:4015–4035, 2024.

[31] ZeMing Gong, Chuanqi Tang, Xiaoliang Huo, Nicholas Pellegrino, Austin T Wang, Graham W Taylor, Angel X Chang, Scott C Lowe, and Joakim Bruslund Haurum. Hyperbolic multimodal representation learning for biological taxonomies. *arXiv preprint arXiv:2508.16744*, 2025.

[32] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[33] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[34] Xin He, Yushi Chen, Lingbo Huang, Danfeng Hong, and Qian Du. Foundation model-based multimodal remote sensing data classification. *IEEE Transactions on Geoscience and Remote Sensing*, 62:1–17, 2023.

[35] Johannes Heinzel and Barbara Koch. Investigating multiple data sources for tree species classification in temperate forest and use for single tree delineation. *International Journal of Applied Earth Observation and Geoinformation*, 18:101–110, 2012.

[36] Marco Helbich, Matthew HEM Browning, Dirk Voets, and Payam Dadvand. Adherence to the 3+ 30+ 300 urban green space rule and mental health, physical activity, and overweight: A population-based study in the netherlands. *Environment International*, 202:109643, 2025.

[37] Hanna Herasimchyk, Robin Labryga, and Tomislav Prusina. Multi-label plant species prediction with metadata-enhanced multi-head vision transformers. *arXiv preprint arXiv:2508.10457*, 2025.

[38] Yao Jin and Chen Wu. Plant taxonomy meets deep learning: Hierarchical classification for fine-grained tree species identification with hyperspectral foundation models. *Available at SSRN 5573998*, 2025.

[39] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations (ICLR)*, 2020.

[40] Daeyeol Kim, Youngkeun Song, Hansoo Kim, Ohsung Kwon, Young-Kwang Yeon, and Taiyang Lim. Airborne multi-seasonal lidar and hyperspectral data integration for individual tree-level classification in urban green spaces at city scale. *International Journal of Applied Earth Observation and Geoinformation*, 136:104319, 2025.

[41] Kathryn R Kirby and Catherine Potvin. Variation in carbon storage among tree species: Implications for the management of a small-scale carbon sink project. *Forest Ecology and Management*, 246(2-3):208–221, 2007.

[42] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 4015–4026, 2023.

[43] George HM Lawrence. *An introduction to plant taxonomy.* Central Book Depot, 1955.

[44] Xiaofan Li, Lanying Wang, Haiyan Guan, Ke Chen, Yufu Zang, and Yongtao Yu. Urban tree species classification using uav-based multispectral images and lidar point clouds. *Journal of Geovisualization and Spatial Analysis*, 8(1):5, 2024.

[45] Yingjie Li, Yuanyuan Mao, Lisa Mandle, Anders Rydström, Roy P Remme, Xin Lan, Tong Wu, Chao Song, Yougeng Lu, Kari C Nadeau, et al. Acute mental health benefits of urban nature. *Nature Cities*, 2(8):720–731, 2025.

[46] Hongjin Lin, Matthew Nazari, and Derek Zheng. Pctrees: 3d point cloud tree species classification using airborne lidar images. *arXiv preprint arXiv:2412.04714*, 2024.

[47] Bingjie Liu, Yuanshuo Hao, Huaguo Huang, Shuxin Chen, Zengyuan Li, Erxue Chen, Xin Tian, and Min Ren. Tscmdl: multimodal deep learning framework for classifying tree species using fusion of 2-d and 3-d features. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–11, 2023.

[48] Haoran Liu, Hao Zhong, Wenshu Lin, and Jinzhuo Wu. Tree species classification based on pointnet++ deep learning and true-colour point cloud. *International Journal of Remote Sensing*, 45(16):5577–5604, 2024.

[49] Hong Liu, Jeff Z HaoChen, Adrien Gaidon, and Tengyu Ma. Self-supervised learning is more robust to dataset imbalance. *arXiv preprint arXiv:2110.05025*, 2021.

[50] Shengheng Liu, Xiang Li, Zhen Zhen, and Yinghui Zhao. Switcher-hnet: A switchable hierarchical network for tree species classification from forest stand to individual tree tasks. *ISPRS Journal of Photogrammetry and Remote Sensing*, 231:329–344, 2026.

[51] Yang Liu, Lei Zhou, Pengcheng Zhang, Xiao Bai, Lin Gu, Xiaohan Yu, Jun Zhou, and Edwin R Hancock. Where to focus: Investigating hierarchical attention relationship for fine-grained visual classification. In *European Conference on Computer Vision*, pages 57–73. Springer, 2022.

[52] Ziwei Liu, Zhongqi Miao, Xiaohang Zhan, Jiayun Wang, Boqing Gong, and Stella X Yu. Large-scale long-tailed recognition in an open world. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2537–2546, 2019.

[53] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[54] Oscar Manas, Alexandre Lacoste, Xavier Giró-i Nieto, David Vazquez, and Pau Rodriguez. Seasonal contrast: Unsupervised pre-training from uncurated remote sensing data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9414–9423, 2021.

[55] William J Manning. Plants in urban ecosystems: Essential role of urban forests in urban metabolism and succession toward sustainability. *The International Journal of Sustainable Development & World Ecology*, 15(4):362–370, 2008.

[56] Aditya Krishna Menon, Sadeep Jayasumana, Ankit Singh Rawat, Himanshu Jain, Andreas Veit, and Sanjiv Kumar. Long-tail learning via logit adjustment. In *International Conference on Learning Representations (ICLR)*, 2021.

[57] Satyam Mohla, Shivam Pande, Biplab Banerjee, and Subhasis Chaudhuri. Fusatnet: Dual attention based spectrospatial multimodal fusion network for hyperspectral and lidar classification. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 92–93, 2020.

[58] Rafael Müller, Simon Kornblith, and Geoffrey E Hinton. When does label smoothing help? *Advances in neural information processing systems*, 32, 2019.

[59] Tamara Münkemüller, Sébastien Lavergne, Bruno Bzeznik, Stéphane Dray, Thibaut Jombart, Katja Schiffers, and Wilfried Thuiller. How to measure and test phylogenetic signal. *Methods in Ecology and Evolution*, 3(4):743–756, 2012.

[60] NEO. Boombasis. https://www.neo.nl/boombasis/, 2025. Accessed: 05-01-2026.

[61] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748*, 2018.

[62] Junyoung Park, Hyung Wook Kwon, and Sungtek Kahng. Mantis: Multi-granularity adaptive network for taxonomy-informed species classification. In *2026 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pages 369–374. IEEE, 2026.

[63] Diane E Pataki, Marina Alberti, Mary L Cadenasso, Alexander J Felson, Mark J McDonnell, Stephanie Pincetl, Richard V Pouyat, Heikki Setälä, and Thomas H Whitlow. The benefits and limits of urban tree planting for environmental and human health. *Frontiers in Ecology and Evolution*, 9:603757, 2021.

[64] PDOK. Pdok luchtfoto infrarood (open). https://www.pdok.nl/introductie/-/article/pdok-luchtfoto-infrarood-open-, 2025. Accessed: 06-01-2026.

[65] PDOK. Pdok luchtfoto rgb (open). https://www.pdok.nl/introductie/-/article/pdok-luchtfoto-rgb-open-, 2025. Accessed: 06-01-2026.

[66] Kevin M Potter, Maria E Escanferla, Robert M Jetton, Gary Man, and Barbara S Crane. Prioritizing the conservation needs of united states tree species: Evaluating vulnerability to forest insect and disease threats. *Global Ecology and Conservation*, 18:e00622, 2019.

[67] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *arXiv preprint arXiv:1612.00593*, 2016.

[68] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30, 2017.

[69] Haohua Que, Haojia Gao, Weihao Shan, Mingkai Liu, Jianshuo An, Fei Deng, Shuo Feng, Xinghua Yang, and Lei Mu. Fm-sam: individual tree crown delineation and classification based on segmentation anything model (sam) and yolov10 in uav imagery for forest monitoring. *Computers and Electronics in Agriculture*, 240:111162, 2026.

[70] Mohammad A Rahman, Laura MF Stratopoulos, Astrid Moser-Reischl, Teresa Zölch, Karl-Heinz Häberle, Thomas Rötzer, Hans Pretzsch, and Stephan Pauleit. Traits of trees for cooling urban heat islands: A meta-analysis. *Building and Environment*, 170:106606, 2020.

[71] Omid Reisi Gahrouei, Jean-François Côté, Philippe Bournival, Philippe Giguère, and Martin Béland. Comparison of deep and machine learning approaches for quebec tree species classification using a combination of multispectral and lidar data. *Canadian Journal of Remote Sensing*, 50(1):2359433, 2024.

[72] Luuk Romeijn, Andrius Bernatavicius, and Duong Vu. Mycoai: Fast and accurate taxonomic classification for fungal its sequences. *Molecular Ecology Resources*, 24(8):e14006, 2024.

[73] Horst Schulz, Sigrid Härtling, and Claus F Stange. Species-specific differences in nitrogen uptake and utilization by six european tree species. *Journal of Plant Nutrition and Soil Science*, 174(1):28–37, 2011.

[74] J. Schäfer. Classifying tree species using lidar point clouds. Master's thesis, University of Amsterdam, 2025.

[75] MM Seeley, BC Wiebe, GP Asner, AJ Abraham, HF Cooper, CA Gehring, KR Hultine, GJ Allan, TG Whitham, T Goulden, et al. Why do classification models go wrong? the importance of adaptations and acclimations in driving landscape-level spectral variation in fremont cottonwood. *Remote Sensing of Environment*, 334:115240, 2026.

[76] Cuiping Shi, Xin Zhao, and Liguo Wang. A multi-branch feature fusion strategy based on an attention mechanism for remote sensing image scene classification. *Remote Sensing*, 13(10):1950, 2021.

[77] Samuel Stevens, Jiaman Wu, Matthew J Thompson, Elizabeth G Campolongo, Chan Hee Song, David Edward Carlyn, Li Dong, Wasila M Dahdul, Charles Stewart, Tanya Berger-Wolf, et al. Bioclip: A vision foundation model for the tree of life. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19412–19424, 2024.

[78] AHN Development Team. Actueel hoogtebestand nederland. https://www.ahn.nl/ahn-the-making-of, 2024. Accessed: 06-01-2026.

[79] Mélisande Teng, Arthur Ouaknine, Etienne Laliberté, Yoshua Bengio, David Rolnick, and Hugo Larochelle. Assessing sam for tree crown instance segmentation from drone imagery. *arXiv preprint arXiv:2503.20199*, 2025.

[80] Andreas Tockner, Ralf Kraßnitzer, Christoph Gollob, Sarah Witzmann, Tim Ritter, and Arne Nothdurft. Tree species classification using intensity patterns from individual tree point clouds. *International Journal of Applied Earth Observation and Geoinformation*, 139:104502, 2025.

[81] Jan Richard Vahrenhold, Melanie Brandmeier, and Markus Sebastian Müller. Mmtscnet: Multimodal tree species classification network for classification of multi-source, single-tree lidar point clouds. *Remote Sensing*, 17(7):1304, 2025.

[82] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 8769–8778, 2018.

[83] Lars Vesterdal, Nicholas Clarke, Bjarni D Sigurdsson, and Per Gundersen. Do tree species influence soil carbon stocks in temperate and boreal forests? *Forest Ecology and Management*, 309:4–18, 2013.

[84] Lars Vesterdal, Inger K Schmidt, Ingeborg Callesen, Lars Ola Nilsson, and Per Gundersen. Carbon and nitrogen in forest floor and mineral soil under six common european tree species. *Forest Ecology and Management*, 255(1):35–48, 2008.

[85] Linlong Wang, Huaiqing Zhang, Rurao Fu, Kexin Lei, Yang Liu, Tingdong Yang, Jing Zhang, and Xiaoning Ge. Mtscfnet: a novel framework for improving tree species classification in a subtropical forest using rgb, lidar-derived, and gf-2 data. *Journal of Forestry Research*, 37(1):22, 2025.

[86] Xudong Wang, Long Lian, Zhongqi Yu, Mingfei Gao, and Stella X. Yu. Long-tailed recognition by routing diverse distribution-aware experts. In *International Conference on Learning Representations (ICLR)*, 2021.

[87] Yue Wang, Yongbin Sun, Ziwei Liu, Sanjay E Sarma, Michael M Bronstein, and Justin M Solomon. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5):1–12, 2019.

[88] Kathleen L Wolf, Sharon T Lam, Jennifer K McKeen, Gregory RA Richardson, Matilda van Den Bosch, and Adrina C Bardekjian. Urban trees and human health: A scoping review. *International journal of environmental research and public health*, 17(12):4371, 2020.

[89] Tz-Ying Wu, Pedro Morgado, Pei Wang, Chih-Hui Ho, and Nuno Vasconcelos. Solving long-tailed recognition with deep realistic taxonomic classifier. In *European Conference on Computer Vision*, pages 171–189. Springer, 2020.

[90] Wenxuan Wu, Zhongang Qi, and Li Fuxin. Pointconv: Deep convolutional networks on 3d point clouds. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 9621–9630, 2019.

[91] Xiaoyang Wu, Li Jiang, Peng-Shuai Wang, Zhijian Liu, Xihui Liu, Yu Qiao, Wanli Ouyang, Tong He, and Hengshuang Zhao. Point transformer v3: Simpler faster stronger. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 4840–4851, 2024.

[92] Jinru Xue and Baofeng Su. Significant remote sensing vegetation indices: A review of developments and applications. *Journal of sensors*, 2017(1):1353691, 2017.

[93] Yong Yang, David Kay Ferguson, Bing Liu, Kang-Shan Mao, Lian-Ming Gao, Shou-Zhou Zhang, Tao Wan, Keith Rushforth, and Zhi-Xiang Zhang. Recent advances on phylogenomics of gymnosperms and a new classification. *Plant diversity*, 44(4):340–350, 2022.

[94] Yuzhe Yang and Zhi Xu. Rethinking the value of labels for improving class-imbalanced learning. *Advances in neural information processing systems*, 33:19290–19301, 2020.

[95] Wenbo Yu, Lianru Gao, He Huang, Yi Shen, and Gangxiang Shen. Hi2d2fnet: Hyperspectral intrinsic image decomposition guided data fusion network for hyperspectral and lidar classification. *IEEE Transactions on Geoscience and Remote Sensing*, 61:1–15, 2023.

[96] Xiaowei Yu, Juha Hyyppä, Paula Litkey, Harri Kaartinen, Mikko Vastaranta, and Markus Holopainen. Single-sensor solution to tree species classification using multispectral airborne laser scanning. *Remote Sensing*, 9(2):108, 2017.

[97] Yuanwen Yue, Damien Robert, Jianyuan Wang, Sunghwan Hong, Jan Dirk Wegner, Christian Rupprecht, and Konrad Schindler. Litept: Lighter yet stronger point transformer. *arXiv preprint arXiv:2512.13689*, 2025.

[98] Jianpeng Zhang, Jinliang Wang, Feng Cheng, Weifeng Ma, Qianwei Liu, and Guangjie Liu. Natural forest als-tls point cloud data registration without control points. *Journal of Forestry Research*, 34(3):809–820, 2023.

[99] Maqun Zhang, Feng Gao, Tiange Zhang, Yanhai Gan, Junyu Dong, and Hui Yu. Attention fusion of transformer-based and scale-based method for hyperspectral and lidar joint classification. *Remote Sensing*, 15(3):650, 2023.

[100] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. Deep long-tailed learning: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 45(9):10795–10816, 2023.

[101] Zhenyu Zhang, Jian Wang, Yunze Wu, Youlong Zhao, and Binjie Wu. Deeply supervised network for airborne lidar tree classification incorporating dual attention mechanisms. *GIScience & Remote Sensing*, 61(1):2303866, 2024.

[102] Hengshuang Zhao, Li Jiang, Jiaya Jia, Philip HS Torr, and Vladlen Koltun. Point transformer. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 16259–16268, 2021.

[103] Maksim Zhdanov, Max Welling, and Jan-Willem van de Meent. Erwin: A tree-based hierarchical transformer for large-scale physical systems. *arXiv preprint arXiv:2502.17019*, 2025.

[104] Qiang Zheng, Yafei Qi, Chen Wang, Chao Zhang, and Jian Sun. Pointvig: A lightweight gnn-based model for efficient point cloud analysis. *arXiv preprint arXiv:2407.00921*, 2024.

[105] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9719–9728, 2020.

Appendix A

Example Tree Point Clouds

Figure A.1: Example LiDAR point clouds of different tree species.

Appendix B

Example Winter Aerial Image Cutouts

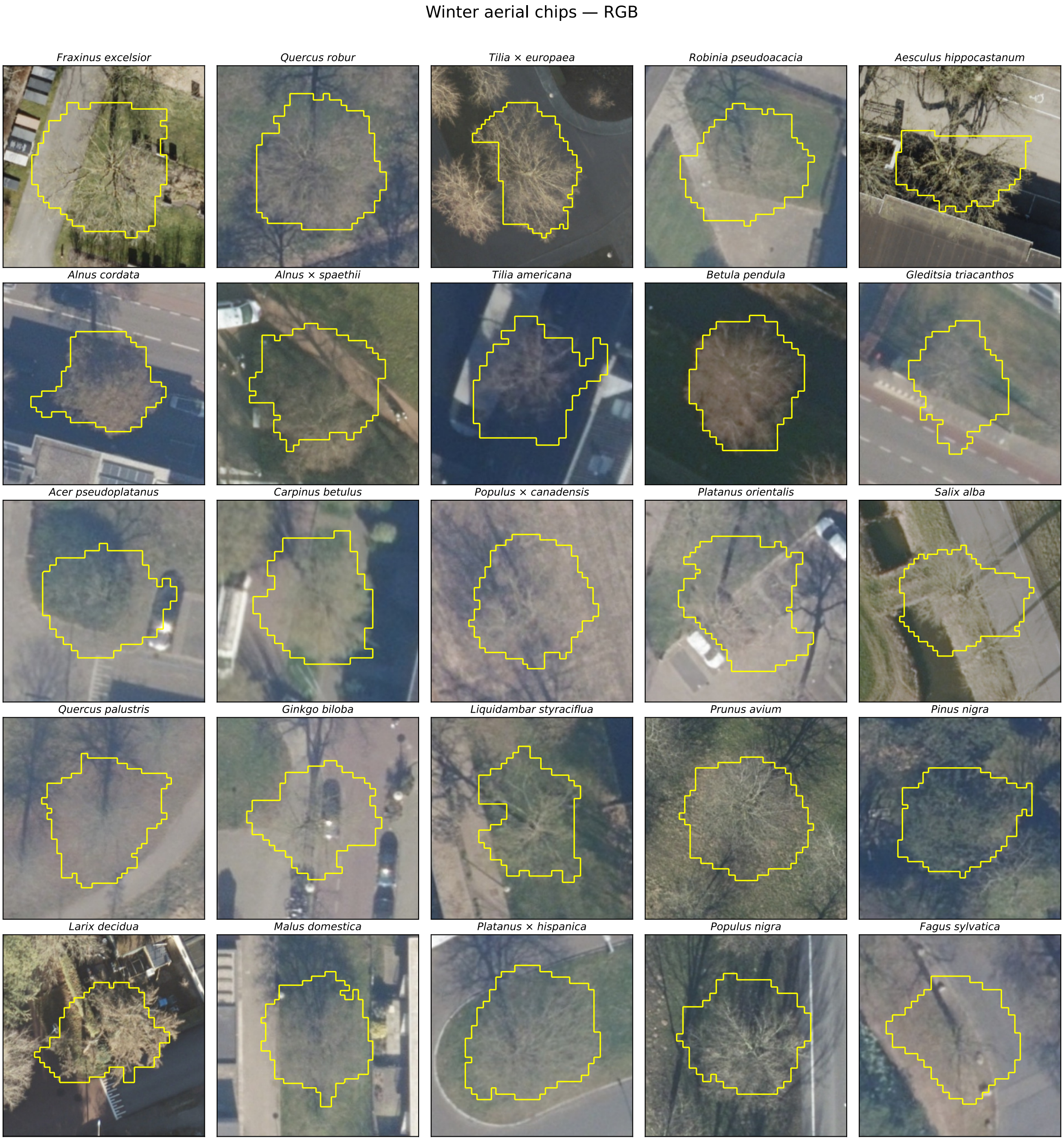


Figure B.1: Example RGB aerial images taken during winter of different tree species.

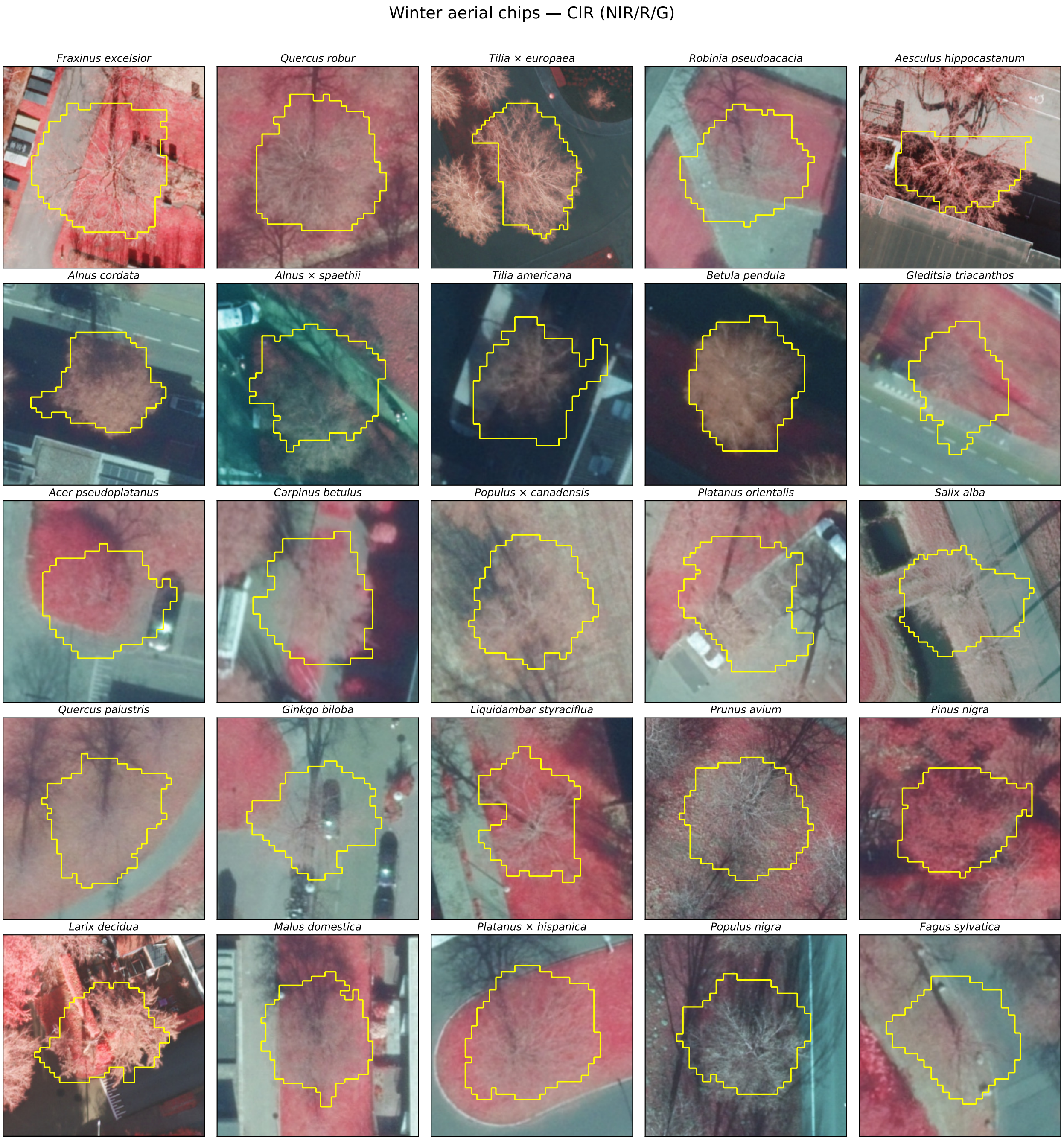


Figure B.2: Example false colour infrared aerial images taken during winter of different tree species.

Appendix C

Example Summer Aerial Image Cutouts

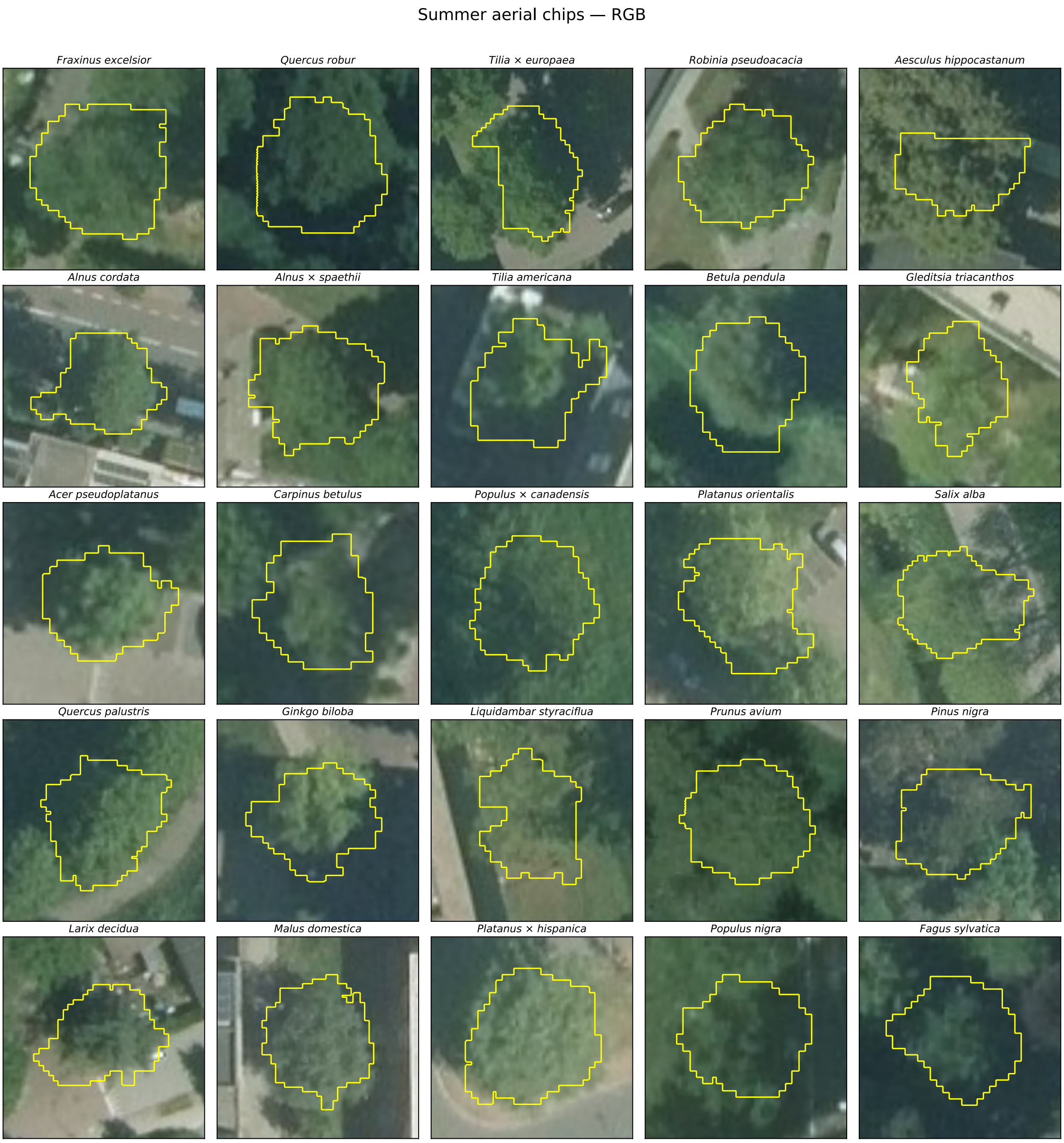


Figure C.1: Example RGB aerial images taken during summer of different tree species.

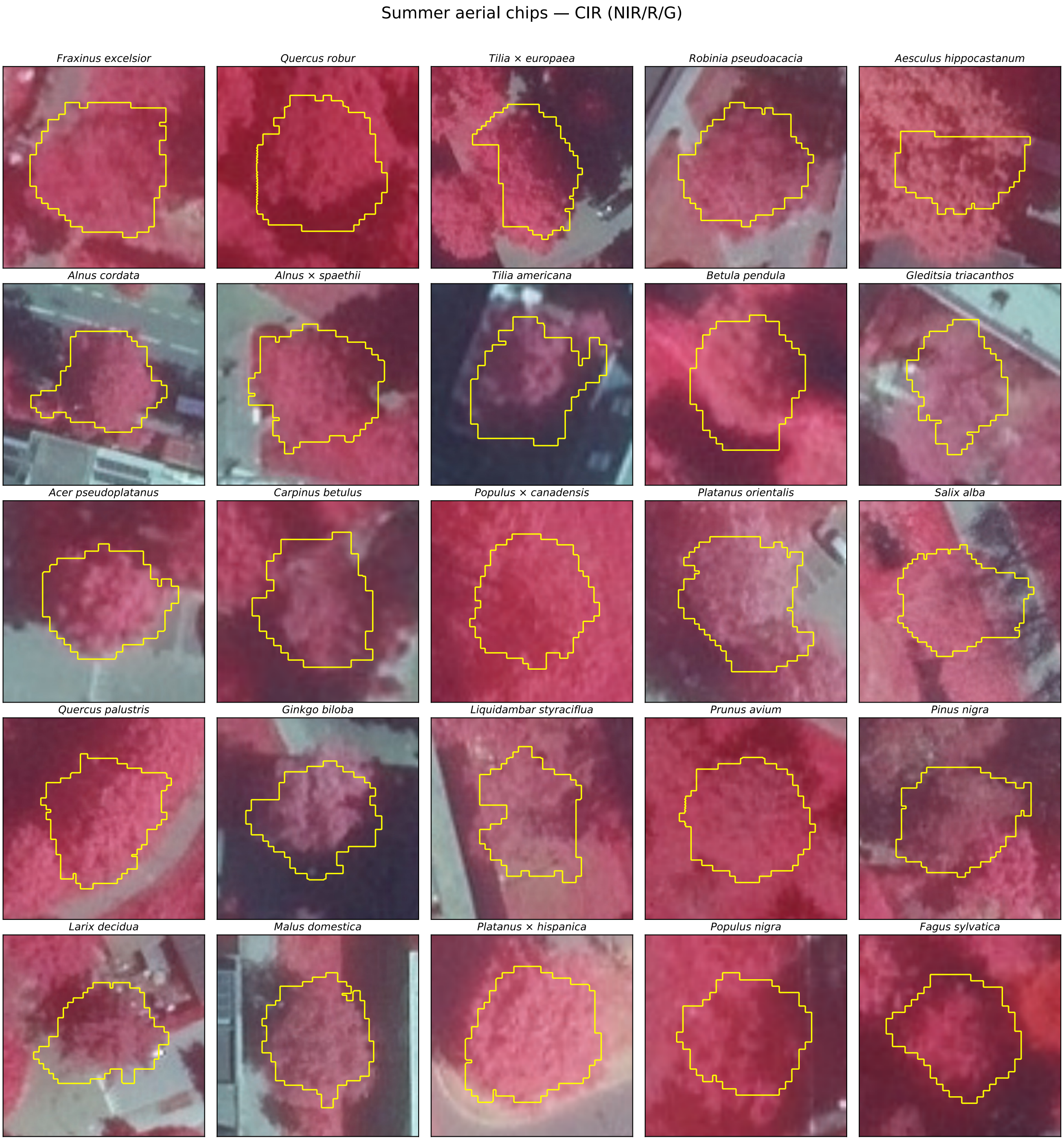


Figure C.2: Example false colour infrared aerial images taken during summer of different tree species.

Appendix D

Unimodal Architecture Diagrams

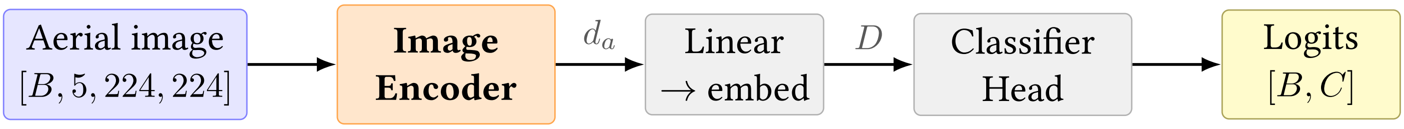


Figure D.1: Aerial-only model. A multi-channel orthophoto chip (RGB, NIR, and optional foreground mask) is passed through an image encoder (Mobilenet_small_v3), projected to the shared embedding dimension D, and classified.

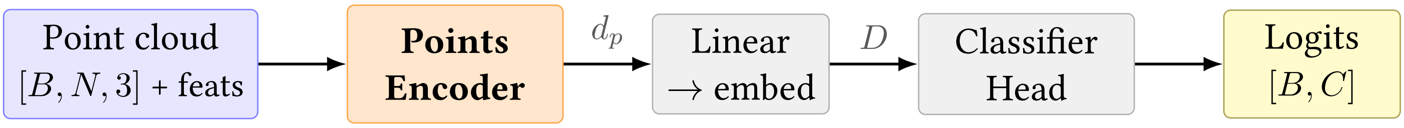


Figure D.2: Point-cloud-only model. The per-tree point cloud (XYZ plus intensity / relative-return features) is passed through a points encoder (default PointNet++) that produces a global feature vector of size d_p, projected to D and classified.

Appendix E

Additional Fusion Architecture Diagrams

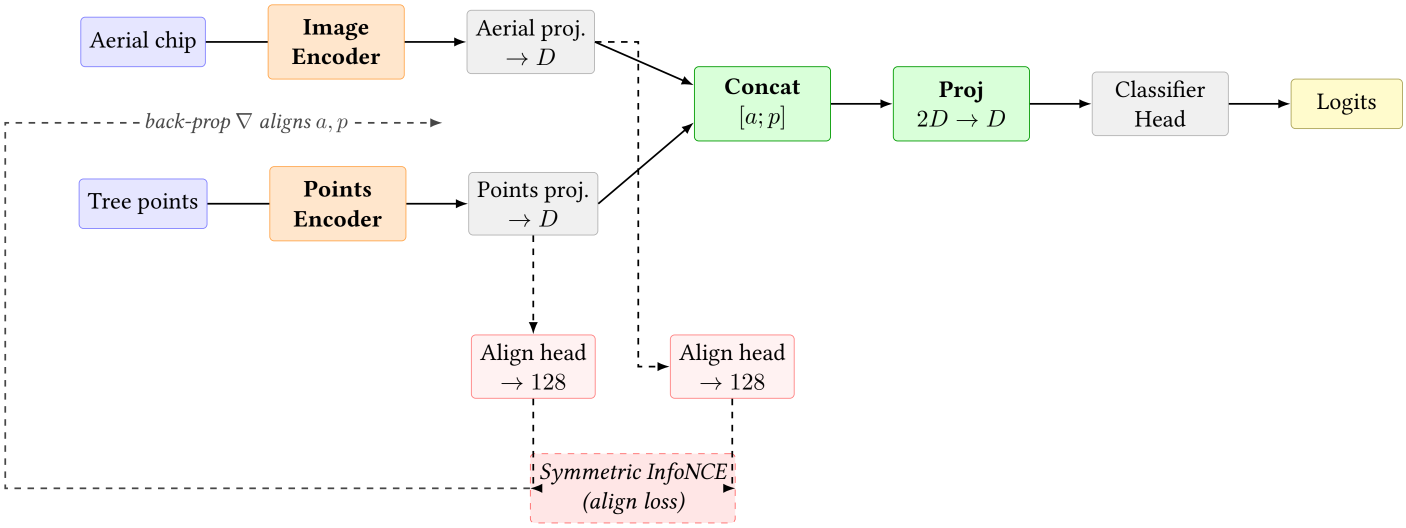


Figure E.1: Fusion model — contrastive variant. The two projected embeddings are concatenated and projected back to D before the classifier. A separate pair of L_2-normalised projection heads feed a symmetric InfoNCE loss (dashed) that pulls the aerial and point-cloud embeddings of the same tree together in a shared hyperspherical space. The objective has no forward path to the classifier; instead it back-propagates into the shared embeddings a, p (grey feedback arrow) — the same vectors the concat fusion and classifier reuse — so it regularises the classifier features only indirectly and at training time.

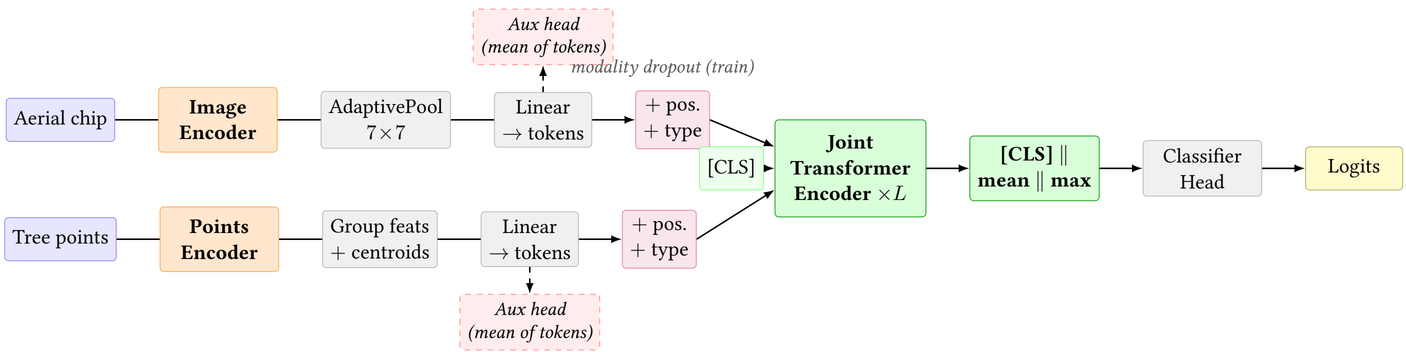


Figure E.2: Fusion model — spatial cross-attention variant. Aerial spatial features are pooled to a 7×7 grid and tokenised; point-cloud group features are tokenised with their centroid positions. After adding positional and type embeddings (and optional modality dropout) the union of tokens with a learned [CLS] is fed through a joint Transformer encoder. The fused representation is read out as the concatenation of [CLS], mean and max pools of the modality tokens, then classified. Auxiliary heads on the mean of each branch's projected tokens (dashed) keep both encoders supervised on the representation the transformer actually consumes.

Appendix F

Taxonomy of Tree Species

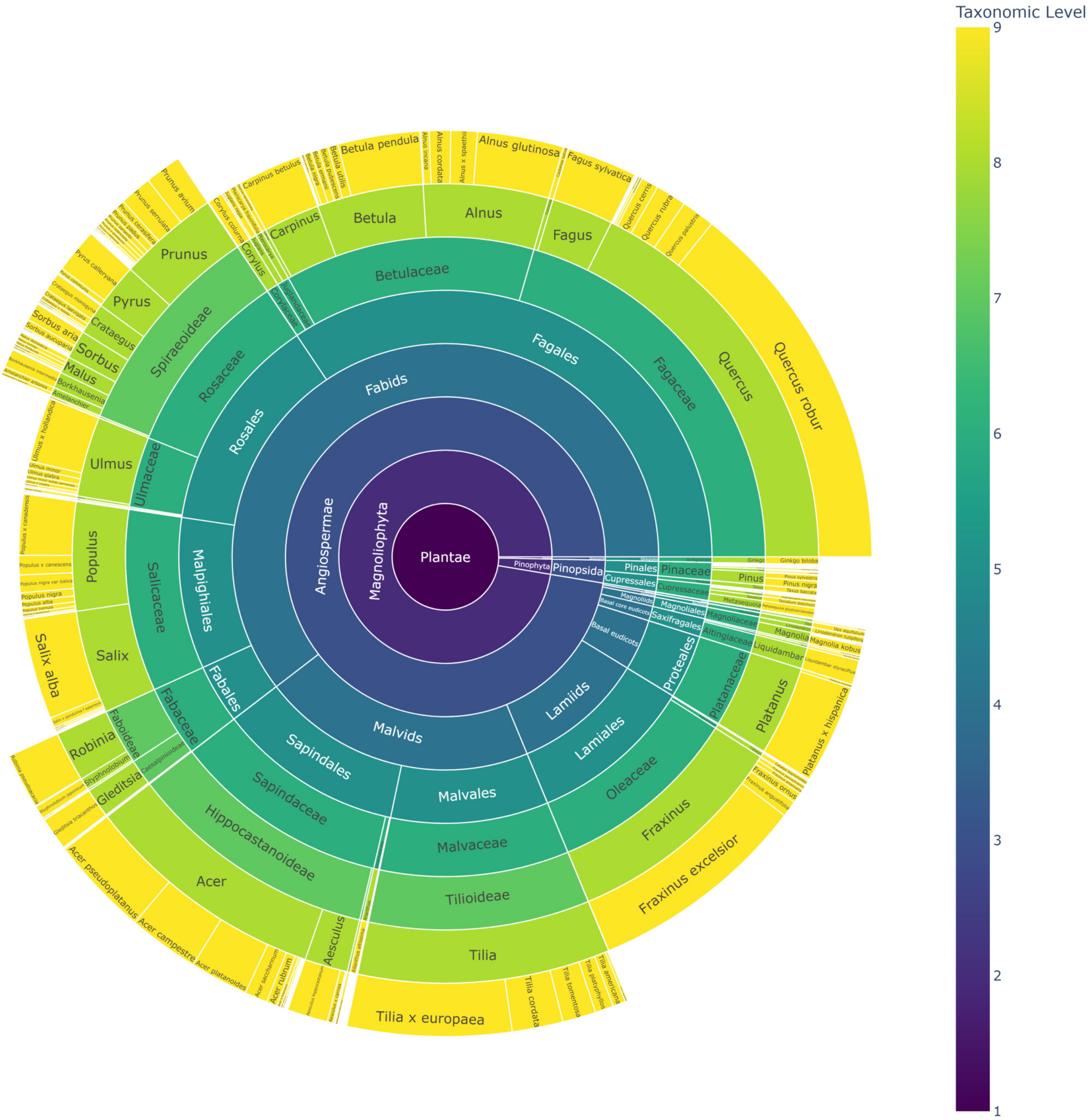


Figure F.1: Distribution of taxonomic classes in our labelled dataset, shown in rings where each layer is a taxonomic level, ordered from kingdom (centre) to species (outside edges). The size of each taxon is scaled relative to the frequency of each taxon in the dataset.

Appendix G

Per Municipality Metrics

Municipality	# samples	# species	Accuracy	Balanced accuracy	Mean Confidence
Alblasserdam	331	53	0.7885	0.5851	0.8590
Albrandswaard	598	77	0.8378	0.6242	0.8680
Amersfoort	3088	132	0.8491	0.5082	0.9028
Amsterdam	13688	211	0.7694	0.3431	0.8154
Apeldoorn	4527	106	0.8361	0.4424	0.8924
Arnhem	2369	121	0.7986	0.5013	0.8686
Assen	2986	88	0.8617	0.4565	0.8966
Barendrecht	1042	84	0.8983	0.7168	0.8993
Delft	1881	106	0.7682	0.4856	0.8419
Den Haag	7215	198	0.7211	0.3399	0.7816
Dordrecht 1	397	61	0.8413	0.5653	0.9199
Dordrecht 2	2722	113	0.8170	0.4657	0.8696
Dronten	1100	87	0.9027	0.6910	0.9278
Ede	870	30	0.8782	0.6683	0.9334
Eindhoven	6611	154	0.8282	0.4995	0.8722
Groningen	7632	152	0.7331	0.3778	0.8075
Haarlem-Zandvoort	3192	131	0.7027	0.4008	0.7870
Hengelo	1834	99	0.8103	0.4465	0.8705
Hilversum	1805	136	0.7518	0.4943	0.8262
Kempen 1	1683	61	0.8283	0.4302	0.8924
Kempen 2	2589	72	0.8521	0.4996	0.9074
Leeuwarden	156	33	0.7051	0.5267	0.8285
Lelystad	1032	81	0.5833	0.3150	0.6768
Nijmegen	3662	143	0.7835	0.3940	0.8421
Ouder-Amstel	352	46	0.6335	0.5287	0.7856
Papendrecht	168	34	0.8274	0.5629	0.8602
Ridderkerk	1143	94	0.8215	0.5625	0.8614
Sliedrecht	298	67	0.6644	0.4759	0.8160
Sudwest-Fryslan	2555	94	0.7804	0.4844	0.8583
Tilburg	6151	142	0.8498	0.4790	0.8909
Utrecht	7615	150	0.8004	0.4050	0.8654
Velsen	831	79	0.7653	0.5671	0.8203
Zaanstad	2435	108	0.8045	0.5242	0.8398
Zoetermeer	2433	102	0.8545	0.6345	0.8850
Zundert	943	24	0.9109	0.4892	0.9431

Table G.1: Statistics and performance metrics per municipality.

Appendix H

Per Species Metrics

Species	Support	# Predicted	Recall	Precision	Mean Confidence
QUERO	18405	19240	0.9156	0.8759	0.9180
FRXEX	9250	9787	0.8752	0.8272	0.8891
TILVU	5406	5785	0.8892	0.8309	0.8994
PLTHY	4633	4740	0.9719	0.9500	0.9729
ACRPP	3521	3780	0.7291	0.6791	0.7800
FAUSY	3379	3553	0.8390	0.7979	0.8762
ALUGL	3075	3097	0.7317	0.7265	0.7939
POPCA	2920	3179	0.8627	0.7924	0.8708
BETPE	2840	3115	0.7461	0.6803	0.7933
ULMHO	2749	2886	0.7796	0.7426	0.8200
FRXAX	12	7	0.2500	0.4286	0.7661
CTLSP	12	3	0.1667	0.6667	0.7979
PRNDO	12	2	0.0833	0.5000	0.3335
PIUST	12	5	0.2500	0.6000	0.6383
ACRCS	11	6	0.4545	0.8333	0.6186
TILFL	10	7	0.6000	0.8571	0.6433
CYLAV	10	2	0.1000	0.5000	0.5334
MAGSO	10	5	0.2000	0.4000	0.6173
SOUTH	10	2	0.0000	0.0000	0.4194
AECPV	10	2	0.0000	0.0000	0.4504
CIPCO	1	1	0.0000	0.0000	0.5943
QUEEL	1	0	0.0000	NaN	0.3414
CUYCV	1	1	1.0000	1.0000	0.4887
PTFRH	1	0	0.0000	NaN	0.3335
PRNNI	1	0	0.0000	NaN	0.2406
FRXXA	1	0	0.0000	NaN	0.4531
IUPCH	1	0	0.0000	NaN	0.7311
IUPCO	1	0	0.0000	NaN	0.1142
IUPVI	1	0	0.0000	NaN	0.6246
THJDO	1	0	0.0000	NaN	0.6265

Table H.1: Performance per Species. The rows are binned by 10 species from the head, 10 from the middle and 10 from the tail section of the frequency distribution (given by the support). Note that for some species there are NaN values, this occurs for species that are not predicted at inference and some metrics can therefore not be calculated.

Appendix I

Accuracy and Loss Curves

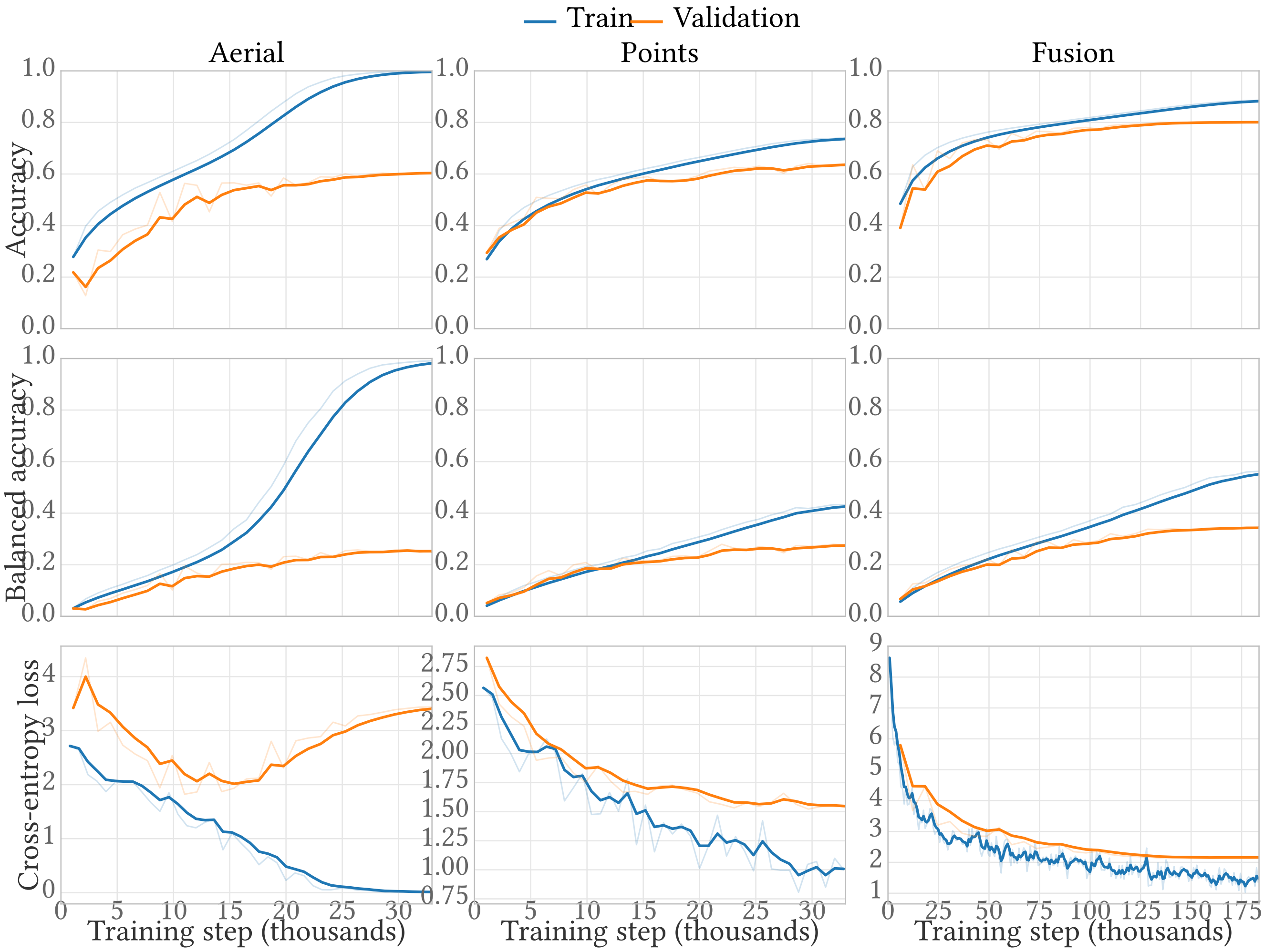


Figure I.1: Training and validation accuracy, balanced accuracy and loss curves. Note that for the aerial and point cloud models these are collected from 10% subset run, while the fusion model curves are from the complete-dataset run.

Appendix J

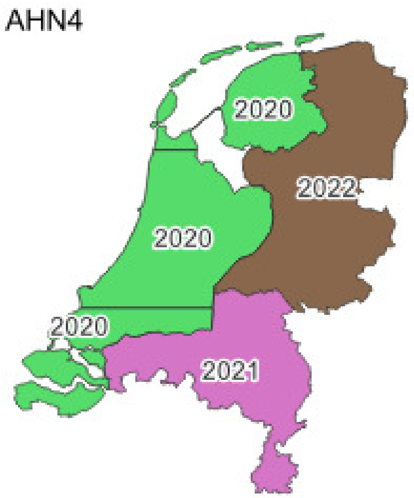
Taxonomy Hyperparameter Sweeps

Function	Value	Δ Accuracy	Δ Balanced Accuracy
Baseline	0.0	0.26	0.05
Taxonomic label smoothing	**0.1**	0.31	0.05
	0.2	0.28	0.05
	0.3	0.30	0.05
Multi-head classification	0.1	0.28	0.05
	0.3	0.31	0.05
	0.5	0.28	0.05
Taxonomic embedding	0.1	0.29	0.04
	0.3	0.29	0.06
	0.5	0.28	0.05
	1.0	0.28	0.05

Table J.1: Swept hyperparameter values for the taxonomic functions. The selected value for each of the taxonomic functions, based on the accuracy and balanced accuracy metrics, is given in bold. These results were obtained on a $\sim 1\%$ subset of the dataset.

Appendix K

AHN Data Collection Areas



(a) Areas over which AHN4 was collected [78].

(b) Areas over which AHN5 was collected [78].

(c) Areas over which AHN6 was collected [78].