Supplementary Material for WACV 2025 Submission N°1854: ”PureForest: A Large-Scale Aerial Lidar and Aerial Imagery Dataset for Tree Species Classification in Monospecific Forests”

A. Data Access and Code Repositories

The PureForest dataset is hosted on HuggingFace under the Open Licence 2.0 of Etalab: IGNF/PureForest.

Benchmark code for 3D models comes from the Myria3D library [7] which was adapted for the task of scene classification. It is available at github.com/IGNF/myria3d/tree/PureForest/Encoder-Nocolor.

Benchmark code for the image classifier is available at github.com/IGNF/PureForest-Baseline.

B. Structure of files and directories in PureForest

```
PureForest
├── lidar
│   └── train\val\test
│       └── {SPLIT}-{species}-C{class_id}-{id}.laz
├── imagery
│   └── train\val\test
│       └── {SPLIT}-{species}-C{class_id}-{id}.tiff
└── metadata
    ├── PureForest-patches.csv
    ├── PureForest-patches.gpkg
    └── PureForest-dictionnary.csv
```

Naming convention for data patches is shown in the above figure, where `SPLIT` is either TRAIN, VAL, or TEST, `class_id` indicates the semantic class identifier (zero-based), and `patch_id` is a unique patch identifier. For instance, `TEST-Quercus_pubescens-C0-199_7_327.laz` refers to a Lidar point cloud of a deciduous oak forest from the test set. In folder metadata, one can find three files:

- `PureForest-patches.gpkg`: listing of all patches with class labels and some metadata, including their membership to an annotation polygon.
- `PureForest-patches.csv`: same content, except without patch geometries, provided for convenience.
- `PureForest-dictionnary.csv`: gives a reference mapping for all species present in the dataset to their french/english/latin names, their category in the *BD Forêt V2*, and their species code as defined by the NFI.

C. PureForest statistics

C.1. Support of semantic classes in train, val and test sets.

Class	ID	Train Patches	Train Polygons	Val Patches	Val Polygons	Test Patches	Test Polygons
Deciduous oak	0	15840	63	4374	14	27841	14
Evergreen oak	1	11609	26	372	4	10380	6
Beech	2	7008	21	1626	4	4036	4
Chestnut	3	3337	16	147	2	200	3
Black locust	4	1663	83	323	12	317	12
Maritime pine	5	4568	20	960	3	2040	4
Scotch pine	6	11330	34	2429	7	4506	5
Black pine	7	4356	16	942	3	1928	3
Aleppo pine	8	4028	15	233	2	438	2
Fir	9	96	2	722	1	22	1
Spruce	10	2579	16	627	3	868	4
Larch	11	2536	7	503	1	255	1
Douglas	12	161	11	265	2	104	2

C.2. Support of tree species in train, val and test sets.

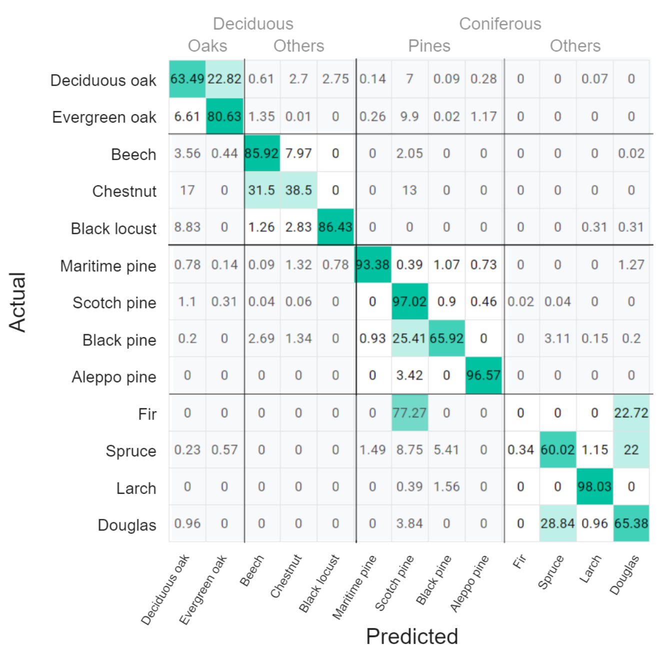
Class	ID	Tree species	Train Patches	Train Polygons	Val Patches	Val Polygons	Test Patches	Test Polygons
Deciduous oak	0	Quercus robur	144	4	1	1	302	1
Deciduous oak	0	Quercus pubescens	12937	51	4084	11	27496	11
Deciduous oak	0	Quercus petraea	2749	6	279	1	43	2
Deciduous oak	0	Quercus rubra	10	2	10	1	0	0
Evergreen oak	1	Quercus ilex	11609	26	372	4	10380	6
Beech	2	Fagus sylvatica	7008	21	1626	4	4036	4
Chestnut	3	Castanea sativa	3337	16	147	2	200	3
Black locust	4	Robinia pseudoacacia	1663	83	323	12	317	12
Maritime pine	5	Pinus pinaster	4568	20	960	3	2040	4
Scotch pine	6	Pinus sylvestris	11330	34	2429	7	4506	5
Black pine	7	Pinus nigra laricio	1824	7	916	2	1288	2
Black pine	7	Pinus nigra	2532	9	26	1	640	1
Aleppo pine	8	Pinus halepensis	4028	15	233	2	438	2
Fir	9	Abies nordmanniana	29	1	0	0	0	0
Fir	9	Abies alba	67	1	722	1	22	1
Spruce	10	Picea abies	2579	16	627	3	868	4
Larch	11	Larix decidua	2536	7	503	1	255	1
Douglas	12	Pseudotsuga menziesii	161	11	265	2	104	2

D. Experimental results for the image model

D.1. Classwise test metrics of the image model.

Class	Prec.	Rec.	F1	Acc.	Patches
Deciduous oak	94.8	63.5	76.1	63.5	48055
Evergreen oak	56.7	80.6	66.6	80.6	22361
Beech	88.9	85.9	87.4	85.9	12670
Chestnut	6.3	38.5	10.9	38.5	3684
Black locust	25.9	86.4	39.9	86.4	2303
Maritime pine	**95.2**	93.4	94.3	93.4	7568
Scotch pine	54.2	97.0	69.5	97.0	18265
Black pine	89.9	65.9	76.1	65.9	7226
Aleppo pine	64.1	96.6	77.0	96.6	4699
Fir	0.0	0.0	0.0	0.0	840
Spruce	84.7	60.0	70.3	60.0	4074
Larch	87.4	**98.0**	**92.4**	**98.0**	3294
Douglas	22.9	65.4	33.9	65.4	530

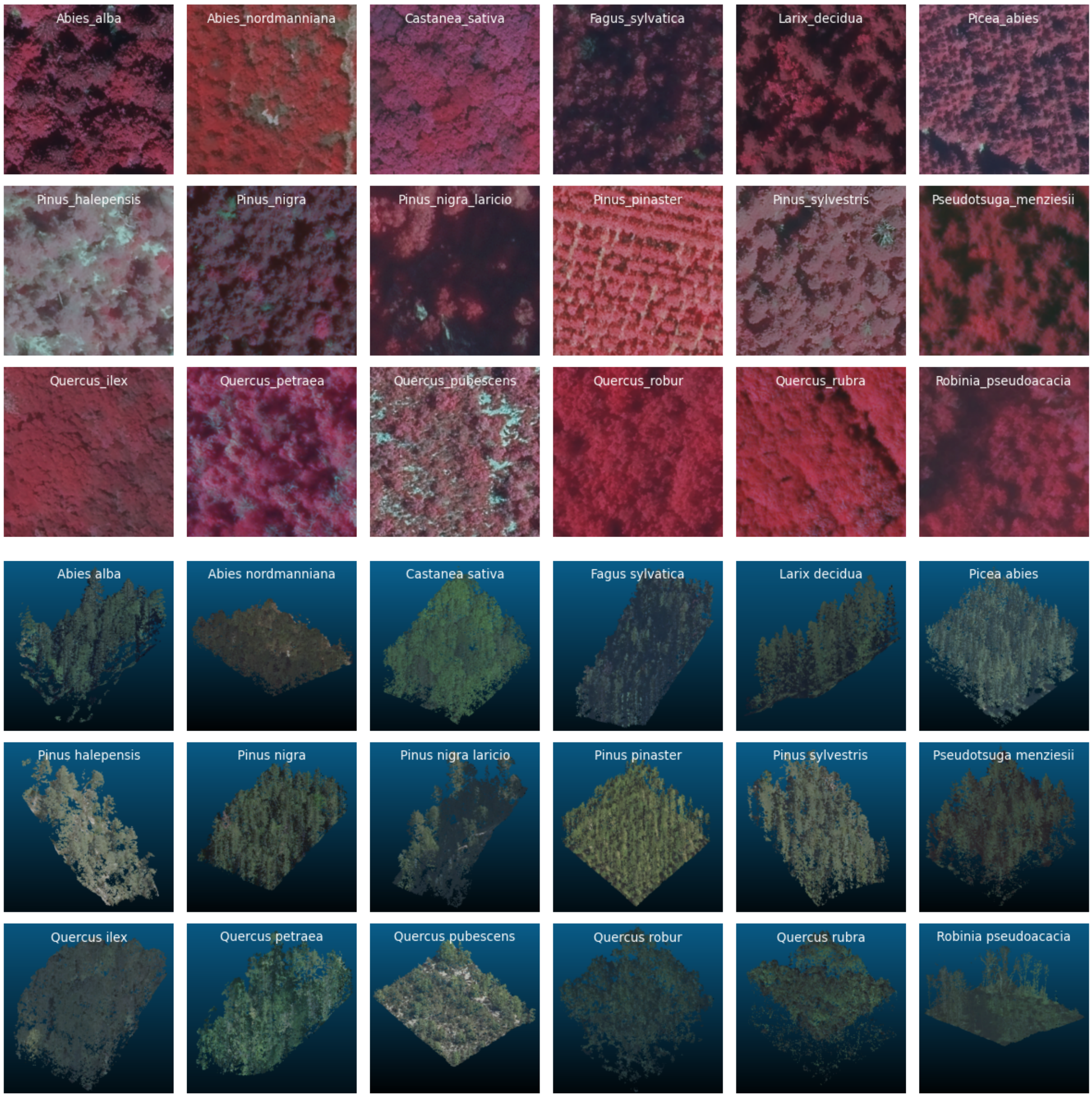
D.2. Confusion matrix of the image model on the test set normalized by rows.



E. Semantic relationship between tree species, classification labels, and foliage type.

french name	latin name	english name	label	foliage
Chêne rouvre	Quercus petraea	Sessile oak	Deciduous oak	deciduous
Chêne pubescent	Quercus pubescens	Downy oak		
Chêne pédonculé	Quercus robur	English oak		
Chêne rouge d'Amérique	Quercus rubra	Northern red oak		
Chêne vert	Quercus ilex	Holm oak	Evergreen oak	
Hêtre commun	Fagus sylvatica	European beech	Beech	
Châtaignier commun	Castanea sativa	Sweet chestnut	Chestnut	
Robinier faux-acacia	Robinia pseudoacacia	Black locust	Black locust	
Pin maritime	Pinus pinaster	Maritime pine	Maritime pine	coniferous
Pin sylvestre	Pinus sylvestris	Scots pine	Scotch pine	
Pin laricio de Corse	Pinus nigra laricio	Corsican pine	Black pine	
Pin noir	Pinus nigra	Austrian pine		
Pin d'Alep	Pinus halepensis	Aleppo pine	Aleppo pine	
Sapin pectiné	Abies alba	Silver fir	Fir	
Sapin de Nordmann	Abies nordmanniana	Nordmann fir		
Épicéa commun	Picea abies	European spruce	Spruce	
Mélèze d'Europe	Larix decidua	European larch	Larch	
Douglas vert	Pseudotsuga menziesii	Douglas fir	Douglas	

F. Image and point cloud samples for each of the 18 tree species in PureForest.



Aerial images are displayed in fake colors (near-infrared, red, green)

G. Introduction to 3D deep learning architectures

To provide some background, we give a brief overview of 3D deep learning approaches, and then talk briefly about their application to tree species classification.

Processing unstructured, unordered sets of points is challenging, and researchers would initially try to turn point clouds into structured data i.e. put them “on a grid”. In voxel-based models such as SegCloud [31], 3D convolutions process a voxelized version of a point cloud. In multi-view approaches like SnapNet [3], 2D views or projections of point clouds would be processed by traditional 2D convolutional neural networks. In these methods, the semantic segmentation happens on a regular grid and is then projected back to the point cloud.

Deep learning that operates directly from point clouds is recent [2]. In 2016, [28] introduce PointNet, a pioneering architecture that deals directly with disordered point sets using multiple shared MLPs and symmetric pooling operations. One year later, [29] propose PointNet++, which builds on PointNet layers to process points clouds on nested partitions of the input point cloud. PointNet++ is the best known representative of point-based methods which are characterized by PointNet-like operations hierarchically organized in a U-shaped architecture. From a performance point of view, an interesting successor of PointNet++ is RandLA-Net [12]. Introduced in 2019, RandLa-Net features some performance improvements thanks to the use of random subsampling combined with an explicit consideration of the relative positions of points in space. This makes it a suitable architecture for the large scales characteristic of remote sensing applications.

Point clouds can be represented as graphs: nodes correspond to points and edges capture spatial relationships, which can then be processed by a Graph Convolutional Networks (GNN). In 2018, DGCNN (Dynamic Graph CNN) was introduced by [32], in which edge convolutional layers capture geometric structures. Alternatively, Graph Attention Networks (GAN) have been proposed: in 2021, [9] introduce the Point Cloud Transformer, which fully relies on attention layers to capture geometric structures.

Along 3D neural architectures, new representations of point clouds have been proposed. In 2017, [21] introduce their SuperPoint Graph, a partitioning of a point cloud scene into groups of semantically homogeneous points that can be processed with high efficiency by graph neural networks. Most recently, [30] turned the SuperPoint Graph into a Hierarchical SuperPoint Partition i.e. a partition of nested, increasingly coarser SuperPoints. Additionally, they propose a variant of graph-attention networks to operate directly on the SuperPoint Partition, resulting in highly efficient cloud processing.

For tree species classification from Lidar point clouds, PointNet++ is considered by many as the default solution [5,23–25]. While not the most recent architecture, it is still highly robust and competitive. Interestingly, [23] show that performance is task dependent. Compared to 5 other state-of-the-art architectures, PointNet++ underperforms in a generic 3D object classification benchmark but is among the top three performers in classifying 7 tree species from Unmanned Laser Scanning (ULS) data. In addition, we note that some researchers choose to transpose point cloud data into the 2D domain, allowing them to use image-based methods. For example, [34] build a canopy height model (CHM) from ULS data, which they then feed into a pre-trained image neural network.