SELECTED TOPICS OF DEEP LEARNING APPLICATION IN FOREST RESEARCH

by

Fanyou Wu

A Dissertation

Submitted to the Faculty of Purdue University

In Partial Fulfillment of the Requirements for the degree of

Doctor of Philosophy

Department of Forestry and Natural Resources

West Lafayette, Indiana

December 2021

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF COMMITTEE APPROVAL

Dr. Rado Gazo, Co-chair

Department of Forestry and Natural Resources

Dr. Eva Eaviarova, Co-chair

Department of Forestry and Natural Resources

Dr. Bedrich Benes

Department of Computer Graphics Technology and Department of Computer Science

Dr. Henry J. Quesada

Sustainable Biomaterials Department, Virginia Polytechnic Institute and State University

Approved by:

Dr. Robert G. Wagner

To the memory of my father, 吴春清

ACKNOWLEDGMENTS

Firstly, I am highly grateful to my supervisors, Prof. Rado Gazo and Prof. Eva Haviarova, for their valuable advice, continuous support, and patience during my Ph.D. study. Their tremendous knowledge and abundant experience have encouraged me in my academic research.

Also, I would like to thank Prof. Benes Bedrich and Prof. Henry J. Quesada for their technical support and wise suggestions on my research projects.

Finally, I would like to express my gratitude to my mother, 田国英, and my girlfriend 刘倩云. Without their understanding and support, it would be impossible for me to complete my study.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Digital Forestry uses digital technology and multidisciplinary expertise to measure, monitor, and manage urban and rural forests to maximize social, economic, and ecological benefits.

In chapter 2, we investigated the potential use of CNNs for hardwood lumber identification based on tangential plane images. In chapter 3, we developed deep bark, a lightweight tree species identification application, by using deep learning. In chapter 4, we first introduced a new dataset of images of hardwood species annotated for tree ring detection. We applied the state-of-art semantic segmentation models to the dataset. In chapter 5, we combined the observed classes and non-observed classes by distinguishing the attributes of objects and applied zero-shot learning to microscopic wood images.

The results above chapters demonstrated the potential and effectiveness of machine learning in many forestry-related tasks. Those applications help both the research community and industry to conduct better digital forestry business. However, we still need to point out that the availability, quality, and quantity of data and annotation are critical factors in conducting meaningful research and applications in forestry.

1. INTRODUCTION

1.1 Machine Learning

1.1.1 The history of Machine Learning and Deep Learning

Machine learning (ML) is the study of computer algorithms that can improve automatically through experience and by the use of data [1]. Figure. 1.1 briefly shows some milestones of the development of machine learning research. The concept of Machine Learning was first introduced by Samuel [2] in 1959. Around 1986, when Neural Networks started receiving more attention, Dechter [3] used a new word, deep learning, to describe those methods. Three years Later, French computer scientist Yann Lecun first introduced Convolutional Neural Network (CNN) [4] and later, the variants of CNN became state-of-the-art methods for many computer vision tasks like image classification. After 2009, there was a so-called big bang for Deep Learning. Due to the development of hardware (GPUs and storage), deep learning increased in popularity, and the number of research articles related to it increased exponentially.

There are two types of data for machine learning in real-world scenarios: structure data (sometimes called tabular data) and unstructured data. Structured data comprises clearly defined data types with patterns, e.g., tables and unstructured data that are "everything else," such as images and videos. Tabular data are generally more suitable for tree ensemble methods, while deep learning methods commonly process unstructured data. Since we focus on the unstructured data (images) in this dissertation, we will introduce some deep learning methods in the following sections.

1.1.2 Multilayer Perceptron (MLP)

Multilayer Perceptron (MLP), also called deep feed-forward networks, is the fundamental deep learning model [5]. In some literature, it is also wrongly named a BP-network, where BP represents back propagation. Back Propagation is the algorithm to train the network. The goal of an MLP is to approximate some function f^*. For example,for a classifier,$y = f^*(\boldsymbol{x})$ maps an input x to a category y. An MLP defines a mapping $\boldsymbol{y} = f(\boldsymbol{x}; \boldsymbol{\theta})$ and learns the

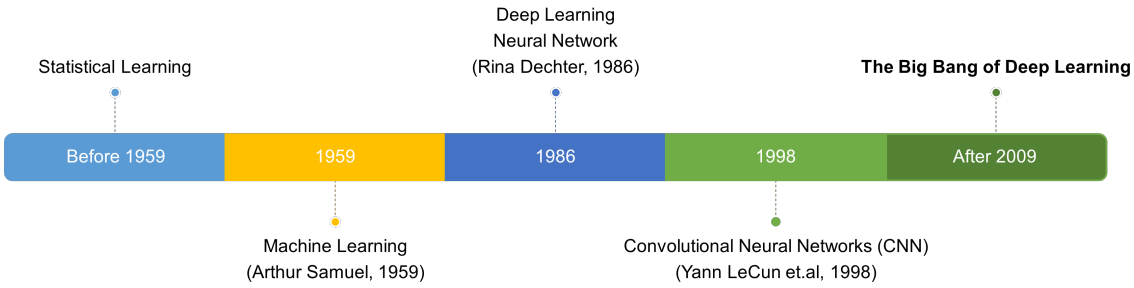


Figure 1.1. Development of machine learning research.

value of the parameters $\boldsymbol{\theta}$ that result in the best function approximation. These models are called feed-forward because information flows through the function being evaluated from x, the intermediate computations used to define f, and finally to the output y. There are no feedback connections in which outputs of the model are fed back into itself. When MLPs are extended to include feedback connections, they are called recurrent neural networks (RNN), as presented in section 1.1.4. When sparse connections are applied, *e.g.*,, for a 2D images, only nearby pixels are connected, then MLP is equal to CNN in section 1.1.3.

Figure 1.2 is an example architecture of a dense MLP. More mathematical: MLP is a group of layers function, each layer is a function $f : \mathbb{R}^m \to \mathbb{R}^n$:

$$f(\boldsymbol{x}) = \sigma(\boldsymbol{A}^\top \boldsymbol{x} + \boldsymbol{b}) \tag{1.1}$$

Here $\boldsymbol{A}$ and $\boldsymbol{b}$ are learn-able parameters and σ is the element-wise operation called activation functions which bring non-linearity to MLP.

Designing and training a neural network e.g., MLP, is similar to training any other machine learning model with gradient descent. An optimization procedure, a cost function, and a model family (network architecture) must be defined in advance.

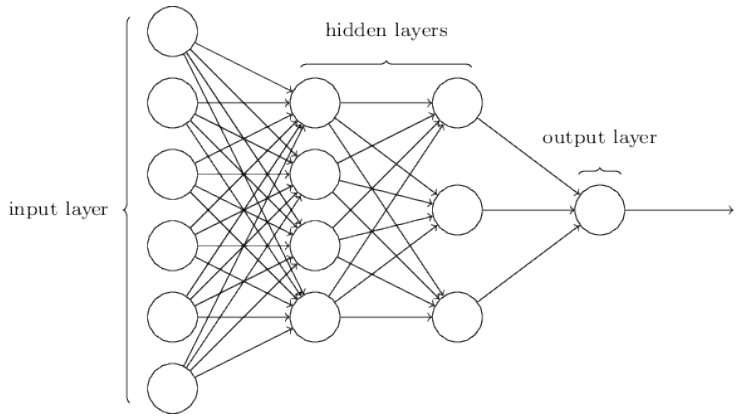


Figure 1.2. Example architecture of a dense MLP [6]

1.1.3 Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) is a deep learning architecture inspired by the natural visual perception mechanism of living creatures. Lecun, Bottou, Bengio, *et al.* [4] developed a multi-layer neural network called LeNet-5 which could classify handwritten digits from MNIST data set. Recent CNNs are comprised of groups of convolutional, pooling, activation, and fully-connected linear functions and include hundreds of thousands connections [5]. In order to speed up the training process and regulate the over-fitting of the network, Batch Normalization (BN) and Dropout layers are often applied in the training phase [7], [8].

There are many implementations of CNNs and among them ResNet [9], DenseNet [10], and MobileNet-V2 [11] are commonly used. ResNet first introduced residues operation, which helps in reducing the problem of accuracy becoming saturated and then degrading rapidly with increasing network depth. DenseNet and MobileNet-V2 were designed to reduce network parameters for real time utilization for portal devices.

1.1.4 Recurrent Neural Network (RNN)

Recurrent neural networks, or RNNs [12], are a family of neural networks for processing sequential data. A recurrent neural network is a neural network that is specialized for processing a sequence of values $\boldsymbol{x}^{(1)}, \cdots \boldsymbol{x}^{(\tau)}$. RNNs can scale to much longer sequences

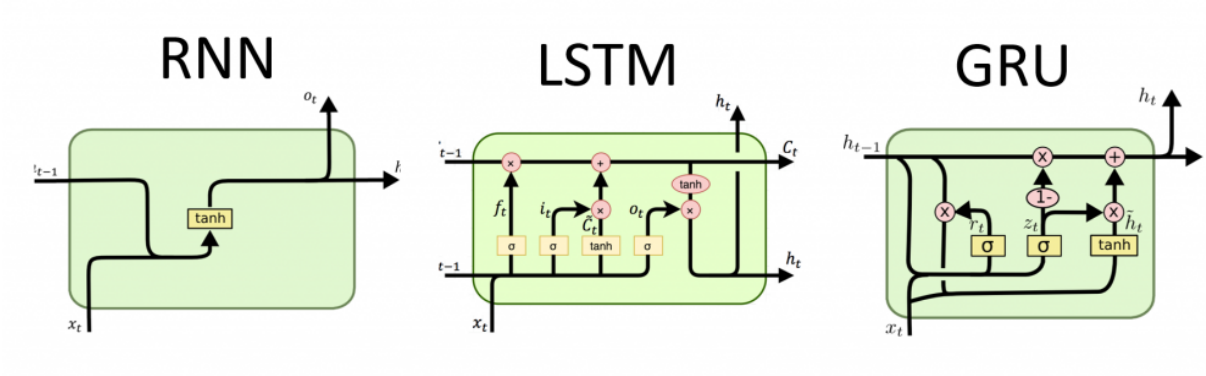


Figure 1.3. Example architectures of RNN, LSTM and GRU cell [16].

than would be practical for networks without sequence-based specialization. Most recurrent networks can also process sequences of variable length. Mathematically, RNNs can be treated as a function f of a dynamic system:

$$\boldsymbol{s}^{(t)} = f(\boldsymbol{s}^{(t-1)}; \boldsymbol{\theta}) \tag{1.2}$$

Here, $\boldsymbol{s}^{(t)}$ is called the state of the system. To predict $\boldsymbol{s}^{(t)}$, we need to know what is $\boldsymbol{s}^{(t-1)}$ which lead RNNs perform well for sequences of values.

However, the vanilla RNN cell (see Figure. 1.3) can easily produce gradient vanish or gradient exploring. LSTM [13] and GRU [14] were developed later to deal with these issues. Adding new parameters to let the cell learn when to cut down the sequence, LSTM and GRU are explicitly designed to avoid the long-term dependency problem (mainly caused by gradient vanish and gradient exploring).

Although RNNs are the ideal architectures for modeling sequence, users found that RNNs are not easy to train due to lack of parallel computing ability and the short effective length of a sequence, which lead to some tasks *e.g.* translating long paragraphs (seq2seq) failing to work. Attention [15] mechanisms are developed to deal with this problem. For most scenarios in text tasks, applying transformers (a cell with attention) became very common.

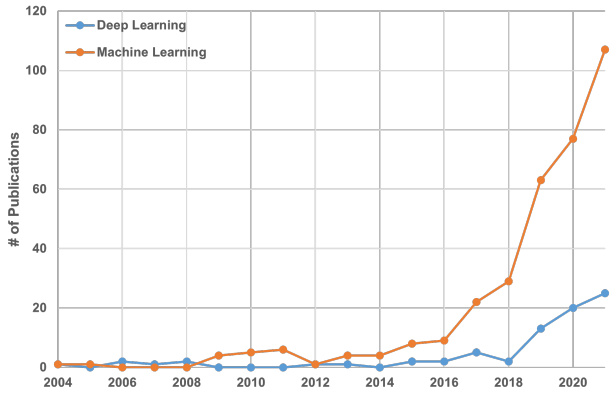


Figure 1.4. The trends of machine learning research in forestry. Data source: Web of Science.

1.2 Machine Learning in Forestry

In forestry, machine learning research and applications have become popular. Figure. 1.4 shows this trends. From 2018, the publication numbers surge when searching keywords Machine Learning plus Forestry or Deep Learning plus Forestry on the Web of Science. We predict this trend will continue in the following 10 to 15 years.

Forestry is a broad science focusing on everything related to forests. So the objectives and data types are diverse. We have 1D data (sound), 2D data (images), and 3D data (spatiotemporal data or cloud points) to solve different tasks in different subfields, e.g., biodiversity, forest profiling, and forest products. In this following section, we will narrow down our eyesight to forest products.

Bark classification has most frequently been treated as a texture classification task. For instance, Local Binary Patterns (LBP) [17], [18], and SIFT descriptors [19] combined with a linear classifier (*e.g.* SVM) are often used. At the same time, [20] extracted four statistical parameters (uniformity, entropy, asymmetry and smoothness) used in texture classification on trunk images, and applied decision tree algorithm for classification. Furthermore, [21] developed a custom segmentation algorithm based on watershed segmentation methods, extracted saliency, roughness, curvature and shape features and fed them to a Random Forest classifier. [22] extracted texture features based on Gabor wavelet and used a radial basis probabilistic network as the classifier.

Deep learning has also been employed for tree identification from bark information, but using a different type of image. In their work, Mizoguchi, Ishii, Nakamura, *et al.* [23] used LiDAR scans instead of RGB images. Some authors have started exploring deep learning on RGB images of textures. By leveraging extracted features from CNNs pre-trained on ImageNet and different region segmentation algorithms, Cimpoi, Maji, and Vedaldi [24] used an SVM to classify texture materials, notably on the Flickr Material Dataset [25].

Similar to bark, wood classification has most frequently been treated as a texture classification task as well. In some works [26]–[30], the basic extraction of wood identification are Gabor Filter, GLCM features and HOG features. Deep Learning methods became popular in recent years. *e.g.,* in [31], [32], CNNs are applied to identity CITES species with relatively high accuracy.

In automatic species identification, the trend of moving from traditional methods (feature engineering plus classifier) to end-to-end deep learning methods is obvious and unstoppable, mainly due to the better performance in predicting and convenience in training. However, these deep learning methods rely highly on large data sets. Our research communities are very traditional, very different from computer science communities, and are not very willing to share data. The data availability issue becomes the major barrier to building better identification applications.

1.3 Computing Platform

In this dissertation, all of the work was conducted using Python [33] programming language with Pytorch [34] frameworks. We used two different types of hardware. For a relatively small dataset, we used a desktop computer equipped with Intel Core^{TM} i7-8700K CPU @ 3.70GHz $\times$12, with 32 GB of memory, and with NVIDIA GeForce RTX 1080 Ti GPU. For large datasets, we used a computing cluster with 20 nodes and each node equipped with a quad-core $\times$ Intel Xeon E5-2630 v4 CPU running at 2.20GHz, with 128 GB of memory, and with 4 $\times$ NVIDIA GeForce RTX 2080 Ti GPU.

1.4 Objectives

The main objective of this dissertation is to develop and apply better machine learning algorithms, mainly focusing on deep learning, to forestry-related tasks. Specific objectives include:

- Develop data-driven deep learning approaches for both macro and micro wood identification.
- Collect bark images from both existing databases and field sampling, improve the state-of-art performance of existing CNNs methods, and develop tree bark identification APP.
- Discover other forestry related tasks which could be solved by deep learning, e.g., detection of ring edge boundaries.

2. WOOD IDENTIFICATION

Acknowledgement

This is a post-peer-review, pre-copyedit version of an article published in Wood Science and Technology. The final authenticated version is available online at: https://doi.org/10.1007/s00226-021-01261-1.

Abstract

Automatic species identification has the potential to improve the efficacy and automation of wood processing systems significantly. Recent advances in deep learning allowed for the automation of many previously difficult tasks, and in this paper, we investigate the feasibility of using Deep Convolutional Neural Networks (CNNs) for hardwood lumber identification. In particular, we tested two highly effective CNNs (ResNet-50 and DenseNet-121) as well as lightweight MobileNet-V2. Overall, we achieved 98.2% accuracy for 11 common hardwood species classification tasks.

Wu, F., Gazo, R., Haviarova, E., & Benes, B. (2021). Wood identification based on longitudinal section images by using deep learning. Wood Science and Technology, 55(2), 553-563.

2.1 Introduction

Current lumber scanners used in industrial wood manufacturing plants, such as rough mills and flooring plants, are used to measure and evaluate the quality and optimize processing of solid wood [35], [36]. Because wood species differ significantly in their color, grain structure, natural characteristics, defects, and density, the scanner sensors often need to be calibrated for each species for their optimal performance. When production switches from one species to another one, the scanner settings must often be manually set. In this study, we attempt to automate the species identification based on image recognition so that the manufacturing equipment can automatically adapt to species being processed or even be able to process batches of mixed species.

Convolutional Neural Network (CNN) is a deep learning architecture inspired by the natural visual perception mechanism of the living creatures. Lecun, Bottou, Bengio, *et al.* [4] developed a multi-layer neural network called LeNet-5, which could classify handwritten digits from the MNIST data-set. Recent CNNs are comprised of groups of convolutional, pooling, activation, and fully-connected linear functions, and they include hundreds of thousands of connections [5]. Batch Normalization (BN) and Dropout layers are often applied in the training phase. BN can speed up the training, while Dropout is a regularization tool that can mitigate overfitting [7], [8]. There are many CNN architectures. Among them ResNet, DenseNet, and MobileNet-V2 are commonly used [9]–[11]. ResNet first introduced residues connections, which can help in reducing the problem of accuracy becoming saturated and then degrading rapidly with increasing network depth. DenseNet improves the short-cut mechanism, connects each layer to every other layer in a feed-forward fashion. MobileNet-V2 was designed to reduce network parameters for real-time utilization for portable devices.

Researchers often treat wood classification as a texture classification task. Gray-Level Co-Occurrence Matrix (GLCM), Local Binary Pattern (LBP), and Gabor filters are popular techniques for the analysis of textures and pattern discrimination [37]–[39]. Paula Filho, Oliveira, Nisgoski, *et al.* [30] compared those techniques combined with a support vector machine to classify 41 Brazilian forest species based on cross-section images. Other techniques for wood identification through macroscopic images have also been proposed, e.g.,[31],

[40], [41]. However, most of the wood identification studies are based on the clean-cut cross-section and follow human expert knowledge. This approach may hinder the real industry application where longitudinal sections are commonly found and surfaces are rough (see Sect. 2.2.1). Recent research in computer vision has shown that CNNs can learn texture instead of shape [42]. Since images of the longitudinal section of boards contain almost all low-frequency texture features, applying CNNs to these images is an obvious choice for species detection. In this work, we apply CNNs to images of longitudinal lumber sections to develop an industrial application for wood species classification.

2.2 Materials and Methods

The pipeline of a CNN classification task is composed of the following steps: 1) data pre-processing, 2) building of CNN networks, 3) training 4) testing and evaluation. In the sections below, we describe each step in detail.

2.2.1 Data Pre-processing

Manually labeled longitudinal images of 11 hardwood species (Figure. 2.1 & Table. 2.1) was acquired by Microtec Goldeneye 300 Multi-Sensor Quality Scanner™. The lumber used in this study was "rough", random width, random length, and kiln-dried. These are industry terms that mean that the lumber was unsorted, of varying lengths and widths, moisture content of 6-8 percent, and the surface was not planed, sanded, or otherwise cleaned. Such surface is called rough because it is cut by a sawmill band saw, then air-dried and kiln-dried for several weeks or months. Boards are often dipped in an anti-stain chemical solution prior to air drying to protect against staining and decay. This treatment tends to preserve the natural color of wood somewhat. Additionally, walnut lumber is steamed prior to drying to achieve a uniform, brown color. As boards travel on conveyors throughout the sawmill and then spend time in the yard and kilns, the surface becomes somewhat weathered and possibly marked by the handling and processing equipment. Additionally, drying stickers can often leave marks on the board surface. This is an important distinction from the identification of wood-based on clean or freshly surfaced wood samples. In this study, each

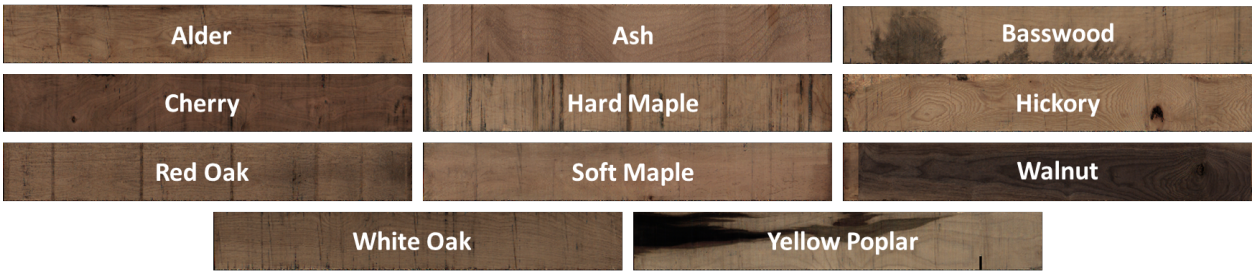


Figure 2.1. Sample images. The area of one pixel corresponds to 0.004 mm^2.

Table 2.1. Species list: Board # represents the number of boards we screened, and patch # is the final patch (70×70 pixels) count for each species. Alder has the original resolution of $500 \times 1,000$ pixels which is 20 times larger than other boards.

Species	**Common name**	**Board #**	**Patch #**
Alnus serrulata	Alder	81	*15714
Fraxinus sp.	Ash	200	2478
Tilia americana	Basswood	40	480
Prunus serotina	Cherry	48	576
Acer saccharum	Hard Maple	818	9816
Carya ovata	Hickory	13	156
Quercus rubra	Red Oak	478	5736
Acer saccharinum	Soft Maple	720	8640
Juglans nigra	Walnut	66	792
Quercus sp.	White Oak	586	7032
Liriodendron tulipifera	Yellow Poplar	108	1296
Total		3158	52716

board is represented by two images, one on top and one on the bottom board face, and the area of one pixel corresponds to 0.004 mm^2. We sampled several image patches (70 pixels $\times$70 pixels) from each board without overlapping. Subsequently, we applied the up-sampling of all patches by using bi-linear interpolating into 224×224, which are commonly used input sizes. In total, we processed 3,158 lumber images, or 52,716 patches, which are listed in Table. 2.1. Hickory is uncommon commercial hardwood species, and its production runs are limited [43]. It was available to us during the data collection process only in a limited volume.

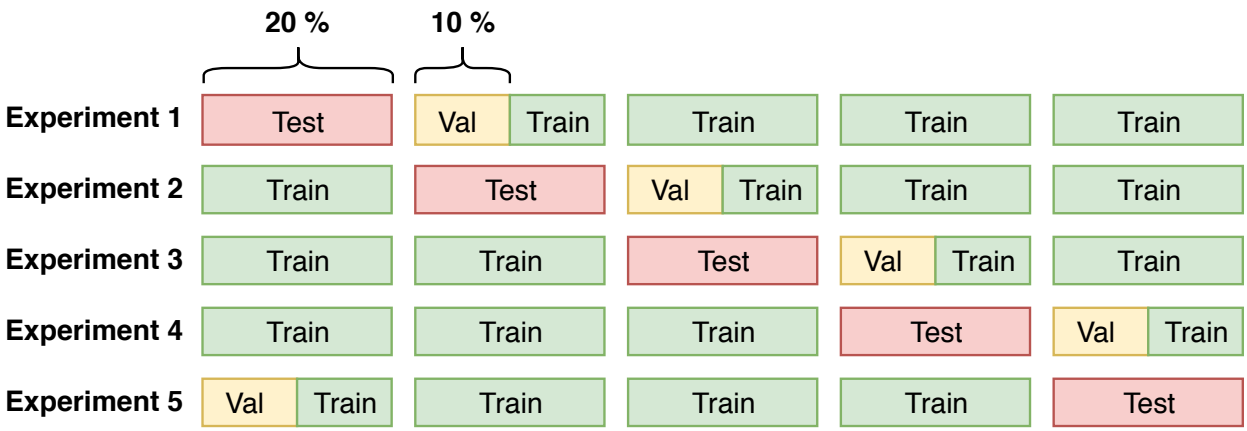


Figure 2.2. Visualization of the cross-validation and splitting of data. Test, Val, and Train represent test set, validation set, and train set, respectively. A rectangle is 20 % of the whole data.

To fully evaluate the performance of CNN, we applied stratified k folds cross-validation to the whole data-set. Cross-validation is a common technique used in machine learning [44]. We applied stratified five folds (20% each) cross-validation without overlapping (Figure. 2.2). For each experiment, four folds of data (80%) were used during training, as the training set (70%) and validation set (10%), and the rest one-fold of the data (20%) was the test set. Data splitting was performed at the board level. Training set and validation set were used in the training phase. The training set was used to train CNN, while the validation set was used as the indicator for the choice of final model parameters. In our experiments, the final model weights were selected based on maximizing the accuracy of the validation set. Finally, the reported accuracy is based on the test set.

2.2.2 Architecture

We used the state-of-the-art CNN architectures: ResNet, DenseNet, and MobileNet. These architectures contain several versions, which differ in the total number of layers. In this study, based on our 10-thousand-level data-set size, we select the relatively shallow version of the above-mentioned CNNs: Resnet-50, DenseNet-121, and MobileNet-V2. Table. 2.2 summaries our selected computing parameters. Resnet-50 has the largest number of parameters (25.55M) and best predicting performance. However, as the trade-off for larger param-

Table 2.2. Parameter Numbers, FLOPS and estimated wall times of CNNs.The estimation is based on a 10 GFLOPS device.

	Params (M)	**FLOPS (G)**	**Wall Time (ms)**
ResNet-50	25.55	4.14	414
DenseNet-121	7.98	2.90	290
MoblieNet-V2	3.50	0.33	33

eter space, Resnet-50 performs slower during the inference phase and may become difficult to train or easy to overfit when the data-set is small. DenseNet-121 and MobileNet-V2 are advanced architectures that reduce parameter numbers (7.98M and 3.50M respectively) and speed up inference time while retaining as high classification accuracy as possible. Table. 2.2 shows the number of parameters and the floating-point operations per second (FLOPS) for each model. FLOPS is a model inference speed quantification and it provides information about the inference time when the device capacity is known. We also show an estimated wall time of each model when processing one image. The estimation is based on a 10 GFLOPS device which is common for CPU. The real inference speed is related to FLOPS and it also depends on the implementation. While DesnesNet-121 has smaller FLOPS than ResNet-50, the training and inference time of DesnesNet-121 are in our implementation.

2.2.3 Training

We used SGD optimizer with a mini-batch size of 64, momentum of 0.9, and Cross-Entropy loss for the training of all models. To further validate the effects of the optimizer, we use the same learning rate, and we also use the Adam optimizer to train those models. The learning rate used in Adam optimizer varies from SGD because Adam is a self-adaptive optimizer. Common computer vision data augmentation methods for rotation, flip, and transforming images into gray-scale were used. An identical learning rate schedule was used for all three models. First, we trained all networks with an initial learning rate of 0.045 for 30 epochs with a weight decay of 0.94 for two epochs. Subsequently, we picked the model weights, which performed the best on the validation set, and used the initial learning rate of 0.025 with the same weight decay strategy for ten more epochs. We repeated the second

step but used the initial learning rate of 0.001 for five more epochs. The second and third steps are based on the observation that validation set accuracy tends to fluctuate, indicating that the current learning rate is too large. This training strategy is efficient for data mining competitions with relatively small data-sets and can be considered as a regularization [5].

Using a pre-trained model is a transfer learning method, which works well in a small to medium-sized data-set. Common practical transfer learning approaches for classification tasks are based on a two-step method - first, train the last fully connected layer and freeze the remaining layers for several epochs, and then fine-tune the entire network. This two-step method can stabilize the entire training process. However, based on our preliminary tests, this two-step method does not perform better than a single-step transfer learning, which trains the entire network directly. We further discuss this in the ablation study in Sect. 2.4.

2.2.4 Testing and Evaluation

Ensemble learning is useful to improve accuracy during the testing phase. We used majority voting and models ensemble to enhance the robustness and performance of the results. The majority of voting is suitable for our purposes since patch images are sampled from boards. Depending on different wood species, the number of images per board is either 12 or 14, except for Alder (194). The majority voting applied to board level classification leads to incremental results. The model ensemble is the summary of all probability outputs per class of all the above models as the ensemble output. We used a simple probability ensemble with equal weight for all three models in this research.

2.3 Results and Discussion

We have implemented our system on a desktop computer equipped with Intel Core^{TM} i7-8700K CPU @ 3.70GHz $\times$12 , with 32 GB of memory, and with NVIDIA GeForce RTX 1080 Ti GPU. All the implementations of models and Grad-CAM are based on PyTorch 1.4.

Table. 2.3 shows the overall accuracy and the macro F1 of the predictions of our models. Even though our data-set is not very balance, the macro F1 is still aligned with accuracy in our cases. So in the following, we will focus on analyzing accuracy for simplicity. For single

Table 2.3. Model performance. The value is represented as mean ± variance for 5 models.

Data Level	Model	Accuracy	Macro F1
Patch	ResNet-50	0.9519 ± 0.0065	0.8893 ± 0.0231
	DenseNet-121	0.9384 ± 0.0078	0.8634 ± 0.0323
	MobileNet-V2	0.9352 ± 0.0185	0.8507 ± 0.0363
	Average Ensemble	$\mathbf{0.9560 \pm 0.0075}$	$\mathbf{0.8944 \pm 0.0330}$
Board	ResNet-50	$\mathbf{0.9815 \pm 0.0079}$	$\mathbf{0.9558 \pm 0.0237}$
	DenseNet-121	0.9755 ± 0.0088	0.9408 ± 0.0401
	MobileNet-V2	0.9712 ± 0.0095	0.9076 ± 0.0446
	Average Ensemble	0.9772 ± 0.0087	0.9424 ± 0.0420

model, ResNet-50 performs best in both patch and lumber identification (0.9519 and 0.9815 respectively), followed by DenseNet-121 (0.9384 and 0.9755) and MobileNet-V2 (0.9352 and 0.9712). The performance order of the three model architectures is also in line with models trained and evaluated by the ImageNet data-set. For ensemble models, as expected, the top-1 accuracy slightly increases compared to any single model in patch identification. However, for lumber identification, ensemble models do not exceed the performance of ResNet-50. This fact generally indicates that the performance of all single models is highly correlated.

The patch level confusion matrix for all single models is shown in Figure. 2.3. For model comparison, similar to the above-discussed overall accuracy, ResNet-50 performs best in all species classification. Hickory is the most challenging to identify. In Table. 2.4, the precision and recall of Hickory are far less than other species, which also proves the difficulty. One reason is that the training sample is not large: only 156 patches for the whole data-set. Moreover, Hickory is also similar to Ash, which is in line with traditional wood identification. Ash and Hickory are ring porous species with visible rays, visible parenchyma cells and relative dark color, leading to similar patterns in longitudinal section. Alder becomes easiest to identify in patch identification. We need to point out that the high accuracy of the alder patch might not transform to the real-world application because the image resolution and shape of Alder are different from other species in the collected data-set.

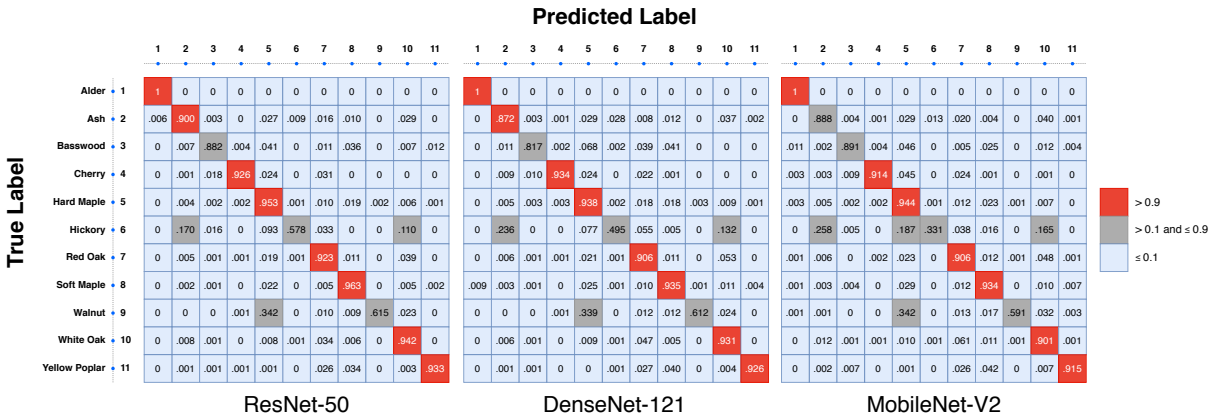


Figure 2.3. Patch level confusion matrix for three models. The x-axis represents the predicted label of models, and the y-axis represents the true label of the patch. Numbers and their corresponding species names are listed on the y-axis. The numbers represent the proportions of the predicted label for one species and should sum up to one.

When diving into the board-level confusion matrix (see Figure. 2.4), majority voting plays a critical role in increasing the accuracy. All three models behave similarly, except predicting Hickory in DenseNet-121. One Densenet-121 model out of five failed to identify Hickory since the Hickory set is small.

Our method has several limitations. First, the data-set is not balanced. There are 11 species with a total of 3,158 boards. However, some species, e.g., Hickory, have less than 0.5% (13/3158) of total boards. This non-balanced issue may affect the results when our trained models are transformed into a real-world application when the unknown species distributions do not parallel those of our data-set. Data sampling and weight method might relieve this problem but will not help when the imbalance rate is significant. In the future, we plan to collect more data from overcoming this issue. Second, some lumber samples exhibited dark stains (see Figure. 2.1) that are not part of the natural anatomy of wood. Instead, they are very common lumber processing marks. In this case, we consider it as the bias in our data-set that might potentially slightly help to increase the final accuracy of CNNs, since these stains appear most frequently in Soft and Hard Maple lumber. Our data does not have labels from the tree level, which may lead to an overestimation of the performance. US hardwood industry is fragmented. While there are few large companies, most are small

Table 2.4. Precision and Recall scores for each species based on patch level prediction and ResNet-50 model.

Species	**Common name**	**Precision**	**Recall**
Alnus serrulata	Alder	0.9999	0.9940
Fraxinus sp.	Ash	0.8719	0.9071
Tilia americana	Basswood	0.8179	0.8529
Prunus serotina	Cherry	0.9330	0.9194
Acer saccharum	Hard Maple	0.9392	0.9210
Carya ovata	Hickory	0.4945	0.4286
Quercus rubra	Red Oak	0.9056	0.8793
Acer saccharinum	Soft Maple	0.9341	0.9553
Juglans nigra	Walnut	0.6126	0.9264
Quercus sp.	White Oak	0.9299	0.9127
Liriodendron tulipifera	Yellow Poplar	0.9253	0.9576

to medium-sized. The nature of the wood industry is such that processing of rough, kiln-dried lumber in factories that utilize a scanner in their process is far removed from the tree harvesting and board milling operations. It is not uncommon for a sawmill to have several hundred small local log suppliers. Each board goes through processes of sawmilling, green grading, sorting by species, length, width, thickness, and grade. When a sufficient quantity is accumulated in each subgroup, the green lumber is sold or air-dried at the mill. Then, it is re-graded and sorted, kiln-dried, graded, and sorted again. When sufficient quantity is accumulated in each sub-group, it is sold and possibly goes through other merchandising steps at concentration yards, distribution yards, and international trade. A secondary wood products manufacturer typically has dozens of lumber suppliers. While the likelihood that any two boards in a package of lumber at a secondary wood processing facility came from the same region is moderately high, the likelihood that any two boards came from the same tree is extremely low. Such point-of-origin information is not kept or available for industrial lumber. Due to these facts, we consider our lumber sample for each species to have sufficient between-trees variation within the US Midwest hardwood region. While there may be slight appearance variation in woods from different regions, their anatomical features do not differ significantly within the species. Therefore the results of this study should transfer to other regions of the hardwood industry.

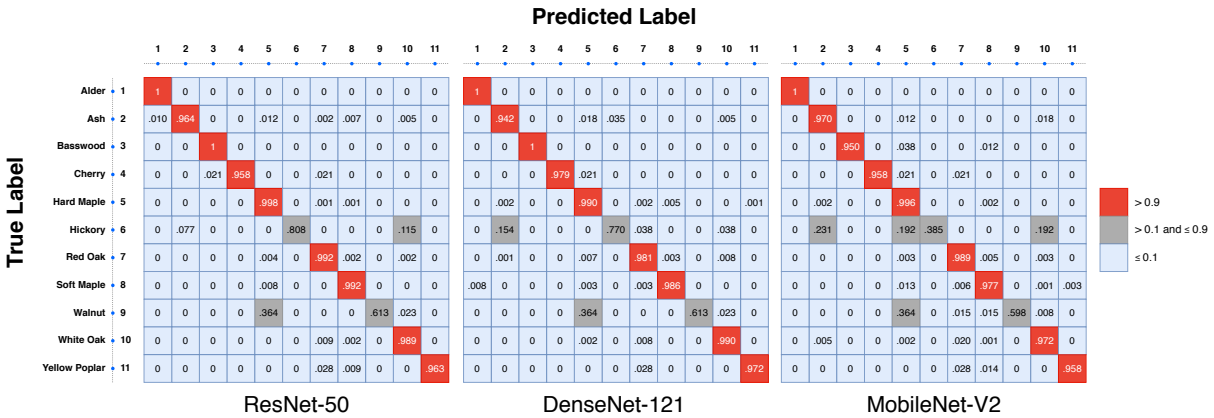


Figure 2.4. Board level confusion matrix for three models. Here x-axis is the predicted label of models, while the y-axis is the true label of the board. Numbers and their corresponding species names are listed on the y-axis. The numbers in each line represent the proportions of the predicted label for one species and should sum up to one.

2.4 Ablation Study

We present an ablation study that reports patch accuracy, where the difference is measurable. Table. 2.5 and Table. 2.6 list the experiments. The baseline refers to the pipeline described in Sect. 2.4.

The input size of the image is critical for model performance. Table. 2.5 shows that when we replace the input size of images from 70×70 to 224×224, where those images are almost identical since the later images are just re-scale from the former ones, the model accuracy increases approximate 2-3 percentages. The possible explanation is that for large input, zero padding effects decrease, thus improving accuracy.

SGD and ADAM [45] are commonly used to optimize the model. In most conditions, SGD is slower but theoretically guarantees to converge, while ADAM is slightly faster but may not guarantee the convergence. Table. 2.5 shows that, in our scenario, ADAM performs similar to SGD, but ADAM is more suitable for DenseNet-121 than ResNet-50 and MobileNet-V2.

Our initial intention was to use gray-scale augmentation to train a more robust model because we considered grain to be a more robust feature than color. By removing gray-scale augmentation (see Table. 2.5), we obtained a slightly more accurate model.

Table 2.5. Ablation study. The value is represent as mean ± variance for 5 models. Here s is the input size to the model.

Experiment	**model**	**Accuracy** $s = 224$	**Accuracy** $s = 70$
Baseline	ResNet-50	0.9519 ± 0.0065	0.9192 ± 0.0197
	DenseNet-121	0.9384 ± 0.0078	0.8846 ± 0.0217
	MobileNet-V2	0.9352 ± 0.0185	0.8942 ± 0.0231
Using Adam as optimizer	ResNet-50	0.9344 ± 0.0080	0.9135 ± 0.0064
	DenseNet-121	0.9483 ± 0.0025	0.9173 ± 0.0062
	MobileNet-V2	0.9391 ± 0.0049	0.9093 ± 0.0060
Removing gray-scale augmentation	ResNet-50	0.9585 ± 0.0129	0.9186 ± 0.0139
	DenseNet-121	0.9467 ± 0.0093	0.9033 ± 0.0119
	MobileNet-V2	0.9561 ± 0.0069	0.8746 ± 0.0328
Using two step transfer learning	ResNet-50	0.9264 ± 0.0153	0.8734 ± 0.0169
	DenseNet-121	0.9338 ± 0.0085	0.8663 ± 0.0107
	MobileNet-V2	0.9227 ± 0.0137	0.8273 ± 0.0384

Table 2.6. Model accuracy after removing maple data. The value is represented as the mean ± and the variance for five models. Here s is the input size to the model.

model	**Accuracy** $s = 224$	**Accuracy** $s = 70$
ResNet-50	0.9550 ± 0.0069	0.9169 ± 0.0384
DenseNet-121	0.9433 ± 0.0039	0.8931 ± 0.0267
MobileNet-V2	0.9366 ± 0.0400	0.8875 ± 0.0277

We also tested the effects of the two-step transfer learning method: first, train the last fully connected layer and freeze the remaining layers for several epochs, and then fine-tune the entire network. We fixed the parameter of the feature extractor for the first six epochs. Table. 2.5 shows that two-step methods performed slightly worse than training the model directly. This phenomenon typically happened when the input domain was very different from the pre-trained data to new data.

Table. 2.6 shows the model performance after removing maple species.

2.5 Conclusion and Future Research

In this study, we investigated the potential use of CNNs for hardwood lumber identification based on tangential plane images. We achieved over 95% successful classification rate for a single model and 98% by applying the model ensemble. The selected CNNs can identify lumber through the tangential plane correctly. In the future, we will focus on analyzing the feature importance of our data by removing specific features and comparing the decrements of performance for different species.

3. BARK IDENTIFICATION

This is a post-peer-review, pre-copyedit version of an article published in European Journal of Forest Research. The final authenticated version is available online at: https://doi.org/10.1007/s10342-021-01407-7.

Abstact

Species identification is one of the key steps in the management and conservation planning of many forest ecosystems. We introduce `Deep BarkID`, a portable tree identification system that detects tree species from bark images. Existing bark identification systems rely heavily on massive computing power access, which may be scarce in many locations. Our approach is deployed as a smartphone application that does not require any connection to a database. Its intended use is in a forest, where internet connection is often unavailable. The tree bark identification is expressed as a bark image classification task, and it is implemented as a Convolutional Neural Network (CNN). This research focuses on developing light-weight CNN models through knowledge distillation. Overall, we achieved 96.12% accuracy for tree species classification tasks for ten common tree species in Indiana, USA. We also captured and prepared thousands of bark images - a dataset that we call Indiana Bark Dataset - and we make it available at https://github.com/wufanyou/DBID.

Wu, F., Gazo, R., Benes, B., & Haviarova, E. (2021). Deep BarkID: a portable tree bark identification system by knowledge distillation. European Journal of Forest Research.

3.1 Introduction

Species recognition is one of the key steps in management and conservation planning of many forest ecosystems [46]–[48]. Autonomous forest inventory could be performed by automatically identifying tree species. Functionality and productivity of forwarders, harvesters and other tree harvesting operations, such as sorting by species, could be improved by automating tree identification [49]. Similarly, automated tree species identification could streamline sawmill merchandising. sorting and processing operations. Tree identification is useful in industrial processing, but can also assist non-professionals in tasks, e.g., land price estimating [50], and in public education.

Bark, leaves, leaf shape, needle distribution, and fruits are important features commonly used to help in tree species identification. Using bark to identify trees has more advantages than using features such as leaves or fruits [51]. The bark is present in all seasons, it does not change significantly between seasons, and it even maintains its main structure after harvesting and during log yard storage. The bark is easily accessible and localized as opposed to tree feature distribution that requires overall tree visibility, presence of leaves or buds. Moreover, tree bark is visually accessible to most machines in standing tree inventory, where foliage and fruits may not be observable. However, using bark alone to identify some tree species may be complicated and unreliable even for experts [52].

Several studies have been conducted in the last two decades to improve tree identification accuracy based on bark, treating it as a texture recognition task [53]. A typical pipeline of texture recognition is to use two-step methods that first extract features from images. Those features are then fed into either linear (e.g., support vector machine (SVM)) or no-linear (e.g., Multilayer Perceptron (MLP)) classifiers . Zheru Chi, Houqiang Li, and Chao Wang [54] proposed a method using Gabor filter banks, and Yuan-Yuan Wan, Ji-Xiang Du, De-Shuang Huang, *et al.* [55] applied the co-occurrence matrices, histogram, and auto-correlation methods to bark identification. Yuan-Yuan Wan, Ji-Xiang Du, De-Shuang Huang, *et al.* [55] reported that adding color features could improve performance, and Jiatao Song, Zheru Chi, Jilin Liu, *et al.* [56] employed the Grey-Level Co-occurrence Matrix (GLCM) assisted by Long Connection Length Emphasis (LCLE) for bark classification. Bertrand, Ameur, Cerutti,

et al. [57] used handcraft features, considering shape, color, structure, and orientation of bark by using Canny filters, hue histogram, and Gabor filters. Boudra, Yahiaoui, and Behloul [58] introduced Termed Statistical Macro Binary Pattern (SMBP), a variant of Local Binary Pattern that represents the intensity distribution within the macrostructure of large spatial support by one macro pattern code. Fekri-Ershad [59] used Local Ternary Patterns (LTP) and then fed them to the Multilayer Perceptron (MLP). Remeš and Haindl [60] introduced rotationally invariant multispectral textural features and reported 90.4% accuracy on BarkNet [51] while using nearest neighbor classifier.

In addition to texture recognition, tree identification based on the bark can also be treated as an image classification task by employing Convolutional Neural Networks (CNNs) [4]. CNNs were successfully used for bark identification in several studies [50], [51], [53], [61]–[63]. These studies report accuracy equally good or better as compared to texture classification methods with benefits of easy implementation and end-to-end training. However, they all utilize large models that are relatively heavily dependent on computing resources, e.g., VGG-19 [64] or ResNet [9].

Currently, the well-performing bark identification systems rely on the internet connection to transfer the bark image to a server and to access massive computing power. However, in many forests, remote online server connections are often not available. An offline framework implemented on a portable device may solve this problem. Knowledge distilling [65] is a modern neural network technique for reducing the size of the neural network while maintaining its performance. Our research focuses on developing a lightweight CNN model through knowledge distillation for tree identification based on the bark.

3.2 Material and Methods

3.2.1 Study Area

The bark images used in this study were collected at Martell Forest near West Lafayette, Indiana, USA (40°25' N; 87°2' W). Martell Forest is operated by Purdue University, Forestry, and Natural Resource Department, Fig. 3.1. It has a total area of 193 ha, of which 70% is covered by deciduous forest.

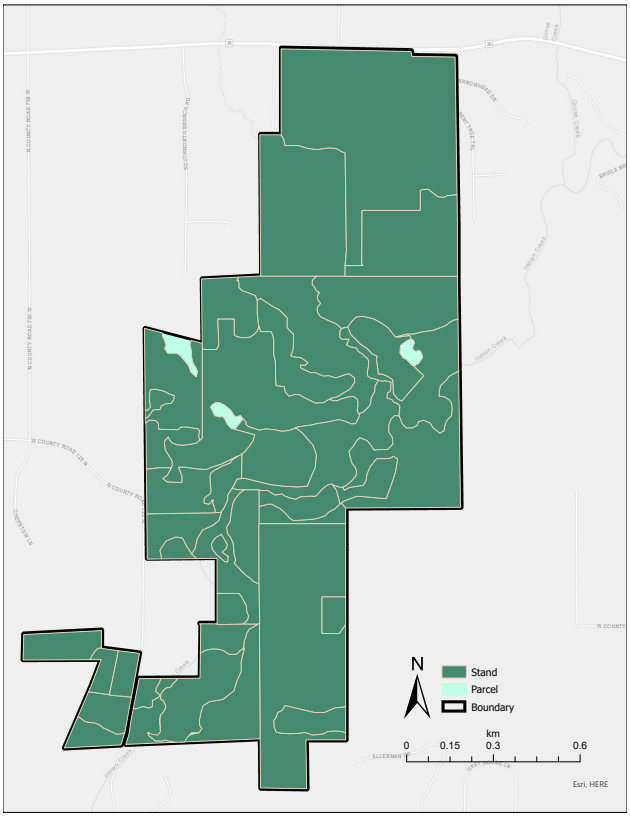
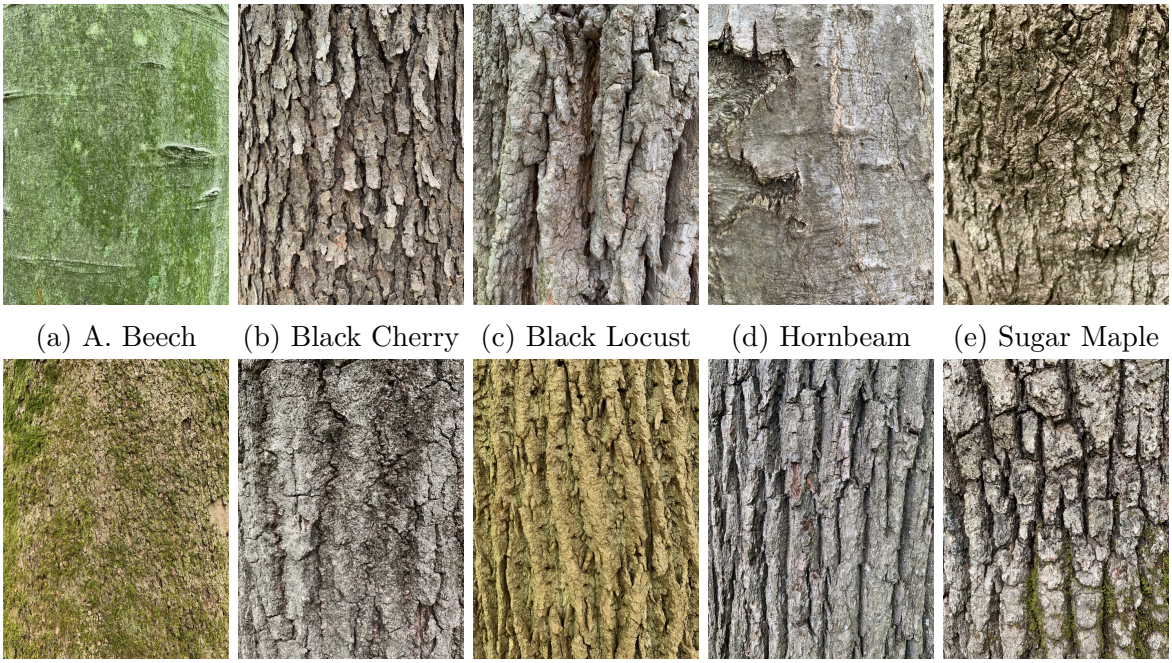


Figure 3.1. Map of study area at Martell Forest near West Lafayette, Indiana, USA (40°25' N; 87°2' W).

3.2.2 Bark Image Data

We collected 309 images from 61 trees of 10 different species (see Fig. 3.2): Sugar Maple (*Acer saccharum*), American Hornbeam (*Carpinus caroliniana*), American Beech (*Fagus grandifolia*), Yellow Poplar (*Liriodendron tulipifera*), Black Walnut (*Juglans nigra*), American Sycamore (*Platanus occidentalis*), Black Cherry (*Prunus serotina*), White Oak (*Quercus alba*), Northern Red Oak (*Quercus rubra*), and Black Locust (*Black Locust*). We used an iPhone Xs to capture the images. The original image resolution was $3,024 \times 4,032$. We took 5-7 images per tree at a distance between 20 - 60 cm away from the trunk. The Diameter at Breast Height (DBH) varied between 20 - 100 cm. The images were divided into non-overlapping patches of resolution 224×224 suitable for deep learning, resulting in 18,540 individual images with about 2,000 images representing each tree species (see



(a) A. Beech (b) Black Cherry (c) Black Locust (d) Hornbeam (e) Sugar Maple

(f) A. Sycamore (g) N. Red Oak (h) Yell. Poplar (i) Black Walunt (j) White Oak

Figure 3.2. Sample images for Indiana Bark Dataset.

Tab. 3.1 for details). We call this Indiana Bark Dataset (IBD), and it is available at https://github.com/wufanyou/DBID.

We also used data from BarkNet [51] that includes 20 different tree species ranging from 24 to 109 trees per species (see Tab. 3.1). The total number of trees is 998. Each tree species is represented by 596 to 2,724 images, and the total number of images is 23,359. The original BarkNet dataset contains 23 different species, but only 20 species were used during their experiments since 3 species have an insufficient number of images to use.

3.2.3 Knowledge Distillation

We applied a vanilla Response-Based Knowledge Distillation. The main idea is that the student model mimics the teacher model to obtain a competitive or superior performance. Here, the teacher model and student model are standard terms in knowledge distillation. For model compression purposes, typically, the teacher model's parameters size is much larger than that of the student model. The response-based knowledge distillation is sim-

Table 3.1. Species list for BarkNet and Indiana Bark Dataset. The last column is the number of non-overlapping sub-images given the crop size 224×224 and down sample rate 2. We directly deleted three species from the BarkNet list which are not used in the experiments due to small number of images.

Dataset	Species	Common name	Trees	Img	SubImgs
Indiana Bark	*Acer saccharum*	Sugar Maple	6	31	1,860
	Carpinus caroliniana	American hornbeam	6	30	1,800
	Fagus grandifolia	American Beech	5	24	1,440
	Liriodendron tulipifera	Yellow Poplar	7	35	2,100
	Juglans nigra	Black Walnut	6	30	1,800
	Platanus occidentalis	American Sycamore	6	30	1,800
	Prunus serotina	Black Cherry	6	30	1,800
	Quercus alba	White Oak	6	32	1,920
	Quercus rubra	Northern Red Oak	7	35	2,100
	Robinia pseudoacacia	Black Locust	6	32	1,920
	Total Indiana Bark		61	309	18,540
BarkNet	*Abies balsamea*	Balsam Fir	41	922	28,235
	Acer rubrum	Red Maple	64	1,676	48,925
	Acer saccharum	Sugar Maple	81	1,999	68,040
	Betula alleghaniensis	Yellow Birch	43	1,255	37,325
	Betula papyrifera	White Birch	32	1,285	33,892
	Fagus grandifolia	American Beech	41	840	2,3904
	Fraxinus americana	White Ash	61	1,472	5,3995
	Larix laricina	Tamarack	77	1,902	11,4956
	Ostrya virginiana	American Hophornbeam	29	612	29,723
	Picea abies	Norway Spruce	72	1,324	35,434
	Picea glauca	White Spruce	44	596	19,673
	Picea mariana	Black Spruce	44	885	43,127
	Picea rubens	Red Spruce	27	740	22,819
	Pinus resinosa	Red Pine	29	596	14,694
	Pinus strobus	Eastern White Pine	39	1,023	25,621
	Populus tremuloides	Quaking Aspen	58	1,037	63,247
	Quercus rubra	Northern Red Oak	109	2,724	72,618
	Thuja occidentalis	Northern White Cedar	38	746	19,523
	Tsuga canadensis	Eastern Hemlock	45	986	27,271
	Ulmus americana	American Elm	24	739	27,821
	Total BarkNet		998	23,359	810,843
	Total all		1,059	23,668	829,383

plistic yet effective for model compression and has been broadly used in various tasks and applications [66].

Specifically, in this paper, we train a larger teacher model and then use the predicted labels of it as soft labels to train a smaller student model. The hard target is the ground truth of an image expressed as a one-hot vector, while the soft target is a predicted vector from a teacher network. The new soft vector label can be seen as a teacher who helps the student network to learn the difficult hard target [65], [66]. Numerically, knowledge distillation is similar to label smoothing, and can regularize the model during training. A detailed discussion of knowledge distillation can be found in [67].

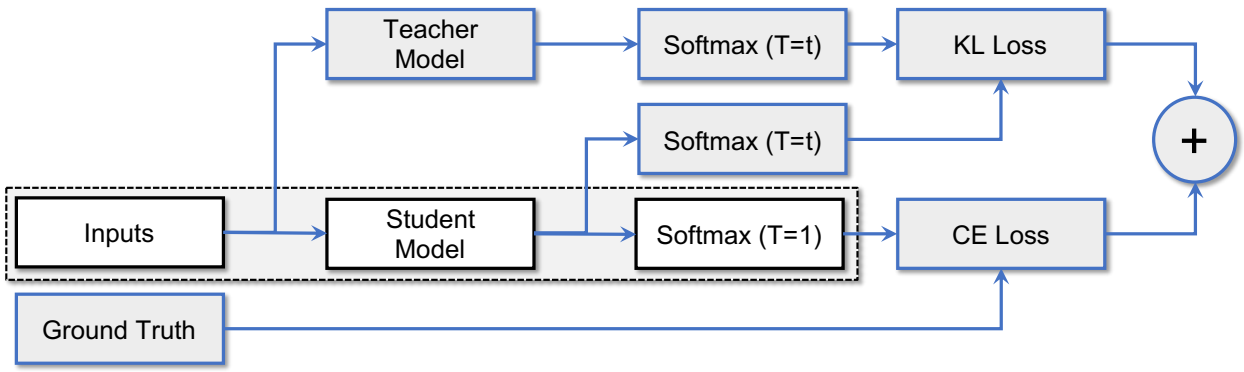


Figure 3.3. Visualization of our implementation of knowledge distillation during training. The black box contains the steps performed during the inference. The KL loss and CE loss are the standard Kullback–Leibler divergence and cross-entropy loss, respectively. t will be set to 5 in this study as the parameter of temperature T.

Fig. 3.3 shows an overview of our implementation of knowledge distillation during the training, while the black boxes contain all the steps used for the inference. The input is a set of images, and the output differs in training and inference. For training, the output is the weighted sum of Kullback–Leibler divergence (KL) loss [68] and cross-entropy (CE) loss [69], while for the inference, the output is the prediction of the student model activated by Softmax.

It is common to calculate the probability q_{i} by Softmax as:

$$q_{\mathrm{i}} = \frac{exp(z_{\mathrm{i}}/T)}{\sum_{\mathrm{j}} exp(z_{\mathrm{j}}/T)}, \tag{3.1}$$

where z_{i} and z_{j} and are the ith and j^{th} components of output of the model. i and j are bounded by the number of classes. $T(T >= 1)$ is a factor called temperature. Like simulated annealing, as T grows, the outputs become smoother, providing more information about which classes the teacher found more similar to the predicted class [65].

We use the following loss function to apply knowledge distillation:

$$\mathcal{L}(\mathbf{Q}_S, \mathbf{Q}_S^\tau, \mathbf{Q}_T^\tau, y) = \alpha T^2 \mathcal{L}_{KL}(\mathbf{Q}_S^\tau, \mathbf{Q}_T^\tau) + (1-\alpha)\mathcal{L}_{CE}(\mathbf{Q}_S, y), \tag{3.2}$$

where $\mathbf{Q}_S$ is the output of the student, and $\mathbf{Q}_S^\tau$ and $\mathbf{Q}_T^\tau$ are the soft outputs from the student and the teacher, respectively, and y is the true label. The symbols $\mathcal{L}_{KL}$ and $\mathcal{L}_{CE}$ are the standard Kullback–Leibler divergence (KL divergence) and cross-entropy loss, respectively, and α is a weight parameter to control the power of knowledge distillation (we used $\alpha = 0.5$ in our experiments). The KL divergence (Sometimes called relative entropy) is a metric commonly used to measure the difference between two distributions. Formally:

$$\mathcal{L}_{KL}(\mathbf{P}, \mathbf{Q}) = \sum_{\mathrm{i}} P_{\mathrm{i}} \log \left(\frac{P_{\mathrm{i}}}{Q_{\mathrm{i}}} \right), \tag{3.3}$$

here P_{i} and Q_{i} are the probability of two difference distribution $\mathbf{P}$ and $\mathbf{Q}$ for i^{th} classes.

3.2.4 `Deep BarkID`

We used two state-of-the-art CNN architectures: ResNet [9] and MobileNet [11]. Each is provided in several versions, which differ in the number of layers. Tab. 3.2 summarize the architectures that we used in this paper. We selected shallow versions of these CNNs: ResNet-34 and MobileNet-V2. Since in the original BarkNet [51] used ResNet-34, for comparison purposes, we use ResNet-34 as well. ResNet-34 has a larger number of parameters (21.79M) and better fitting capacity. However, as the trade-off for larger parameter space, ResNet-34 is slower during the inference phase since it requires 3.68 GMACs (giga multiply–accumulate operation per second). A larger model might become difficult to train or easy to over-fit when the dataset is small. MobileNet-V2 is the advanced architecture that reduces parameter numbers (3.50M), requires only 0.31 GMACs, and is 11.9× faster than ResNet-34. This significantly speeds up inference time while retaining high classification accuracy [65], [66].

We used ResNet-34 as a teacher model and MobileNet-V2 as a student model. We then applied the complete Knowledge Distillation, achieving high prediction accuracy and inference performance by utilizing each model's benefits. We refer to this method as `Deep BarkID`.

Table 3.2. Model architecture summary for ResNet-34 and MobileNet-V2. Residual block and inverted residual block are composed of two convolutional layers that differ in the intermediate channel numbers.

Model	Output Size	Layer	Param (M)
ResNet-34	$64 \times 112 \times 112$	7×7, stride 2	0.01
	$64 \times 56 \times 56$	3×3 max pool, stride 2	0.074
	$64 \times 56 \times 56$	$(3 \times 3, 64$ residual block$)\times 2$	0.148
	$128 \times 28 \times 28$	$(3 \times 3, 128$ residual block$)\times 4$	1.116
	$256 \times 14 \times 14$	$(3 \times 3, 256$ residual block$)\times 6$	6.822
	$512 \times 7 \times 7$	$(3 \times 3, 512$ residual block$)\times 3$	13.114
	$1000 \times 1 \times 1$	average pool, 1000-d fc, softmax	0.513
	Total		21.79
MobileNet-V2	$32 \times 112 \times 112$	3×3, stride 2	0.001
	$16 \times 112 \times 112$	$(3 \times 3, 16$ inverted residual block$)\times 1$	0.001
	$24 \times 56 \times 56$	$(3 \times 3, 24$ inverted residual block$)\times 2$	0.014
	$32 \times 28 \times 28$	$(3 \times 3, 32$ inverted residual block$)\times 3$	0.040
	$64 \times 14 \times 14$	$(3 \times 3, 64$ inverted residual block$)\times 4$	0.184
	$96 \times 14 \times 14$	$(3 \times 3, 96$ inverted residual block$)\times 3$	0.303
	$160 \times 7 \times 7$	$(3 \times 3, 160$ inverted residual block$)\times 3$	0.795
	$320 \times 7 \times 7$	$(3 \times 3, 320$ inverted residual block$)\times 1$	0.474
	$1280 \times 7 \times 7$	$(3 \times 3, 1280$ inverted residual block$)\times 1$	0.412
	$1000 \times 1 \times 1$	average pool, 1000-d fc, softmax	1.281
	Total		3.505

3.2.5 Implementation

Inspired by the implementation from [51], we first downsampled the whole image to the half size of its original resolution to speed up the image reading process. Tab. 3.3 lists all details of our implementation. Most hyperparameter are the same in the [51]. We followed the augmentation approach, and we used image flip and gray-scale, because a fair amount of randomness in terms of illumination and scale, was present during the data gathering process.

We used transfer learning to speed up our training. We applied the ImageNet pre-trained model for most of our experiments and for both ResNet-34 and MobileNet-V2. The pretrained models are obtained from TORCHHUB. We employed a slightly different fine-tuning strategy than the original BarkNet paper: we did not freeze the first layer (7×7

↑https://pytorch.org/hub/

Table 3.3. Implementation Details.

Name		**Parameter**
Training	Optimizer	Adam
	Initial learning rate	10^{-4}
	Batch size	32
	Input Size	224
	Max epoches	40
	Learning rate decay	0.2 at epoch 16 and 33
Knowledge Distillation	Temperature T	5
	Weight Factor α	0.5

CONV), because the bark image data distribution was significantly dissimilar to those from ImageNet. We trained the CNNs with batch size 32 and 40 epochs.

3.2.6 Deployment

Our experiment was implemented on a desktop computer equipped with a quad-core × Intel Xeon E5-2630 v4 CPU running at 2.20GHz, with 128 GB of memory, and with 4 × NVIDIA GeForce RTX 2080 Ti GPU. The training time was about one hour (BarkNet) and five minutes (IBD) for each model. Since the main speed bottleneck during the training phase in our environment is the File IO, there is no significant speed difference for the teacher or the student network. All models were developed based on PyTorch 1.4. We also deployed this model to an iPhone X based on ONNX and Core ML.

3.3 Results and Discussion

The objective of this research was to explore the effectiveness of knowledge distillation. To fully evaluate CNN's performance, we used five-fold cross-validation, which is the same as in [51]. We applied five folds (20% each) cross-validation without overlapping. For each model, four folds of data (80%) were used during training, and the remaining 20% was used for testing. Data splitting was performed tree-level for BarkNet that different trees will be used during the training and testing. Due to the lack of tree-level labels, the image splitting for IDB can only be performed on the image level. However, we can make sure that no same

Table 3.4. Model accuracy comparison. '-' indicates the lack of results in the particular article on the given dataset and bold values indicate the best values given each condition. The method column indicates either the model architecture (e.g., MobileNet-V2) or hybrid methods.

Dataset	Method	Single Crop	Multiple Crop
IBD	ResNet-34	91.20%	97.09%
	MoblieNet-V2	89.32%	95.80%
	`Deep BarkID`	91.90%	96.12%
BarkNet	ResNet-34 [51]	87.04%	93.88%
	Boudra, Yahiaoui, and Behloul [70]	79.10%	-
	Remeš and Haindl [60]	90.04%	-
	ResNet-34 (ours)	90.02%	94.62%
	MoblieNet-V2	88.45%	93.51%
	`Deep BarkID`	88.75%	94.36%

bark area is used for both training and testing. We report average results of single crop accuracy and multiple crop accuracy based on majority voting in Tab. 3.4. Generally, using multiple crops as an ensemble technique will increase the accuracy.

3.3.1 Performance on IBD Dataset

ResNet-34 performed best on the IBD, and this result was in line with our expectations. Knowledge distillation used in `Deep BarkID` contributed to this result, increasing MobileNet-V2 performance from 89.32% to 91.90% and from 95.80% to 96.12% for single crop and multiple crops, respectively. Fig. 3.4 shows the confusion matrix of our `Deep BarkID`. It shows that Yellow Poplar was hard to identify and might be confused with Black Cherry (*Prunus serotina*).

3.3.2 Performance on BarkNet Dataset

For the BarkNet dataset, ResNet-34 performed best and reached 90.02% and 94.62% for single and multiple crops. The performance of our ResNet-34 was higher than reported in the original study (87.04% and 83.88%), probably because we did not freeze its first layer. Knowledge distillation also had a distinct effect. Compared to the vanilla MobileNet-V2, our `Deep BarkID` increased the performance of MobileNet-V2 from 88.45% to 88.75% and

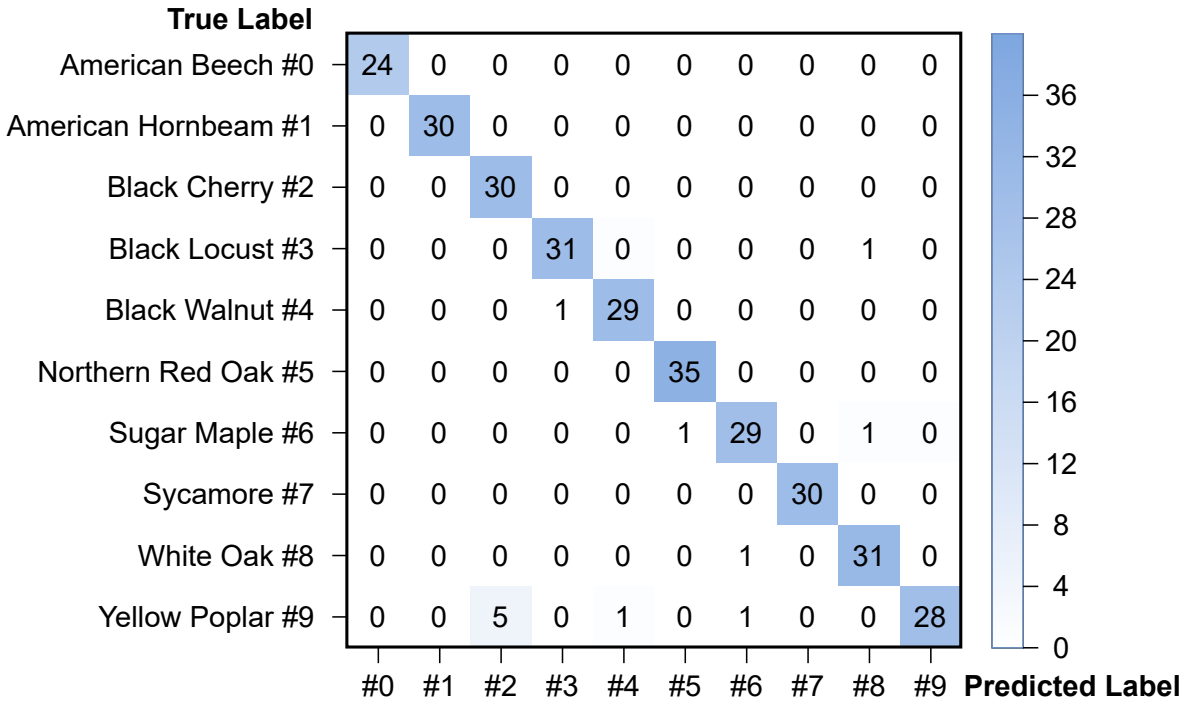


Figure 3.4. The confusion matrix for Multiple Crop of `Deep BarkID` using Indianan Bark Dataset.

Table 3.5. ResNet-34 model accuracy for different weight initialization using the IBD.

Weight initialization	Single Crop	Multiple Crop
[9]	58.60%	66.67%
ImageNet Pretrained	91.20%	97.09%
BarkNet Pretrained	**92.24%**	**97.41%**

93.51% to 94.36% for single and multiple crops, respectively. Since the dataset is relatively large, the effectiveness of knowledge distillation slightly decreases. Remeš and Haindl [60] proposed a texture classification method with relatively higher accuracy on a single crop (90.4%). However, the author did not mention how they split dataset.

3.3.3 Validity of Transfer Learning

Many studies, e.g., [31], [50], [51], [61] show that transfer learning helps in plant identification applications. Still, most of these studies only used the ImageNet pre-trained model. Using a pre-trained model speeds up training based on our experience, but may not improve

the performance. Transfer learning will have better performance if the datasets have similar features and distribution, so we conducted an ablation study to check the power of transfer learning.

Tab. 3.5 shows the results confirming our expectation that using the BarkNet pre-trained model would help to increase model accuracy. In particular, accuracy increased 91.20% to 92.24%. Training the model from the beginning is often challenging (see Tab. 3.5), and we achieved an accuracy of only 58.6% and 66.67%. This result supports the common agreement that transfer learning is useful for plant identification.

3.3.4 Bark Images Dataset for Deep Learning

In this research, we test our methods on IDB and BarkNet datasets only. Currently, there are several other publicly available tree bark image databases, such as AFF Dataset [52], Trunk12 Dataset [71], BarkTex dataset [72] and Bark101 [73]. The AFF bark dataset is a collection of the most common Austrian trees. It contains 1,182 bark samples (960 $\times$ 1325 pixel) belonging to 11 classes. The size of each class varies between 7 and 213 images. AFF samples are captured at different scales and under varying illumination conditions. The Trunk12 dataset (3000 $\times$ 4000 pixels) contains 393 images of tree bark of 12 different trees in Slovenia. The number of images per class varies between 30 and 45 images. Bark images are captured under controlled scale, illumination, and conditions. The types are more homogeneous than those of AFF. The BarkTex dataset contains 408 samples from 6 species, i.e., 68 images per species. Those images have small (256 $\times$ 384 pixels) resolution, and they have unequal natural illumination and scale. The Bark-101 dataset is composed of 101 classes of tree barks from various age and size for a total of 2592 images (69-800)$\times$(112-804) pixels with noisy data like shadows, mosses or illumination changes.

IDB, AFF, Trunk12, and BarkTex datasets share a similar limitation. They are relatively small in size, and the species variance is slight. In other words, it makes the performance evaluation meaningless for most deep learning methods since all methods will achieve good performance. To properly evaluate the performance of deep learning techniques, we highly recommend using large datasets, e.g., BarkNet or Bark-101.

3.3.5 Potential of Deep Learning Methods

Several studies confirm the advantage of using deep learning methods for bark identification [51], [53], [61]. However, some of them point to several drawbacks. Šulc and Matas [53] pointed out that deep learning methods may require massive computing resources and large dataset sizes and argued that portable model, e.g., MobileNet, tends to decrease the model accuracy. We propose that this always does not hold true, and in this paper, we show that using knowledge distillation, a portable model can achieve a comparable performance.

3.3.6 Limitations and Future Work

While our approach shows results that are either comparable or better than the state-of-the-art algorithms, it does not come without limitations. Indiana Bark Dataset size is smaller than the BarkNet. It would be useful to capture more images from more trees in different light conditions, different seasons, and different resolutions and retrain our models. We have developed an App for a portable iOS device shown in Fig. 3.5. However, its user interface could be improved. We also plan to deploy it on on Android devices.

3.4 Conclusion

We developed `Deep BarkID`, a light-weight tree species identification application, by using deep learning. We used transfer learning from BarkNet and knowledge distillation to reduce the inference time of tree species identification from bark images. We achieved 96.12% accuracy for ten tree species classification tasks with the multi-crop setup using the `Deep BarkID`.

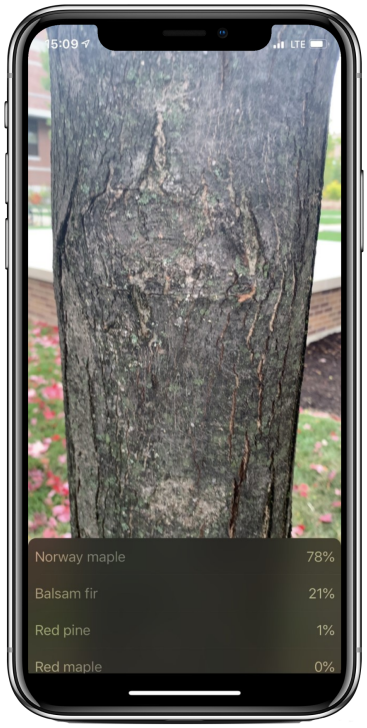


Figure 3.5. A snapshot of `Deep BarkID` deployed on an iPhone X.

4. TREE RING MEASUREMENT

Acknowledgement

A version of this chapter has been under the submission process.

Abstract

Dendrochronology (tree-ring dating) is a quantitative dating method based on analyzing the information from the growth ring of trees. Tree-ring dating is not only beneficial for scientific purposes but also is essential in the wood industry. Basic image processing techniques have been applied to automatic detection of tree-ring boundaries and consequently help in tree ring measurement. But performance of such approaches is limited, especially when the wood surface is rough. In this poster, we focus on providing a better dataset for the research of automatic detection of tree-ring boundaries as well as developing deep learning-based models to solve the problem.

Wu, F., Gazo, R., Benes, B., & Haviarova, E. (2021).

4.1 Introduction

Dendrochronology (tree-ring dating) is a quantitative dating method based on analyzing the information from the growth rings of trees [74], which has been applied in many scientific fields, such as archeology [75], climatology [76], hydrology [77], quaternary geology [78] and other environmental studies. Tree-ring dating is not only essential for scientific purposes but can be beneficial in forest management and the wood industry. In forest management and procurement, the mean annual ring width of logs could help the managers determine average growth rate after previous forest management interventions and the wood quality during processing, and use that information to adjust their forest management strategies. Norell [79] pointed out that tree ring measurement could be performed on either clean cross-sections of wood or rough end faces of tree trucks depending on different demands and circumstances for end face treatment, imaging techniques, and analysis time.

Wood is the secondary xylem of tree plants. It is commonly used to perform tree-ring dating. Earlywood and latewood are frequently combined in a single growth ring. Earlywood has a thinner cell walls and a larger lumens than latewood, and it usually grows early in the season. From an anatomy perspective, softwood (wood from conifers) and hardwood (wood from deciduous trees) have different structures. Softwood is relatively simple, consisting primarily of tracheid cells (90 — 95 % by mass). And in the transverse section (cross section), the growth rings of softwood are either district or not, according to the earlywood to latewood transitions, growth rate of trees, and other factors. Hardwood, in contrast to softwood, is more complex and has more cell types. Within one growth cycle, pores (cross sections of vessel elements), are distributed differently and can be divided into ring-porous (larger pore diameters in earlywood), diffuse-porous (similar pore diameters within one growth ring), and semi ring-porous (between two distinct types). The difficulty of detecting growth ring edges in hardwood species correlates with the pore distribution (simpler in ring-porous species), pore diameter (simpler in large pore diameter species), and other factors, e.g., the color of the wood. Since the hardwood species have more diversity in the cross section, it is still a significant challenge to identify ring edges accurately and automatically for many hardwood species.

Nowadays, the tree-ring measurement process is still time-consuming and often conducted by experts with the help of stereoscopes and software [80]. Commercial software, for example, WinDENDRO™, is helpful in analyzing tree rings by scanners or cameras images. Apart from that, there are also some open-source equivalents, e.g., measuRing [81] and MtreeRing [82], all of which require mostly clear samples for better detecting of tree-ring boundaries. X-ray screening is often used to obtain better images, which helps to differentiate ring edges, and some software e.g., LignoVision™, now supports X-ray images. Either surface cleaning or X-ray screening is not very economically or technically feasible in the forest or sawmill, so it is still valuable to discover better methods to measure annual tree ring widths when the surface of wood log is rough (Figure. 4.1a).

(a) Rough surface.

(b) Clean surface with some annotations.

Figure 4.1. Sample images of Hickory (*Carya spp.*).

Image processing methods have been used to measure tree ring widths automatically. Soille and Misson [83] applied morphological operations, e.g., erosion and dilation, opening and closing, and watershed segmentation, to semi-automatically measure the tree ring area for Norway Spruce (*Picea ablies* L.). Cerda, Hitschfeld-Kahler, and Mery [84] used a Generalized Hough Transform to estimate tree rings with the help of image gradient and edge detection. Such image processing algorithms are often used in tree ring measurement

literature [85]. A more comprehensive review of image processing-based methods of tree ring measurement could be found in Fabijańska and Danek [80], These methods rely highly on the assumption that tree ring images from trunks contain high-contrast edges and trees rings are in a common circular shape. When those methods analyze noisy images, their performance declines drastically.

Modern methods, such as machine learning and especially deep learning, can also be employed to detect tree rings. Fabijańska and Danek [80] treated tree rings detection as a semantic segmentation with the state-of-the-art model U-Net [86]. Similarly, Habite, Abdeljaber, and Olsson [87] applied pix2pix [88], a derivative from generative adversarial network (GAN) [89], to obtain output ring edges. Compared to image processing methods, these supervised learning-based methods often show better result, but simultaneously, require larger dataset size and laborious annotation of images. To our knowledge, there are only few publicly available ring image datasets with annotations, e.g, in Fabijańska, Danek, Barniak, *et al.* [90], which somehow handle the development of creative machine learning application for tree ring edge detection.

In this paper, we focus on 1) developing a new dataset of images of major hardwood species annotated for tree ring detection and 2) apply the state-of-art semantic segmentation models to the dataset.

4.2 Material and Methods

Figure. 4.2 shows the overall methodology of the research. Steps 1-3 are the data collection processes, Steps 4-5 are the annotation processes, and Step 6 is the training and testing process. In this section, we will introduce those steps in detail.

4.2.1 Data Collection

We obtained in total 136 wood cookies (trunk cross sections approximately 10-12 cm thick) of 11 representative US hardwood species with diameters from 25 to 60 cm. Table. 4.1 summaries critical information for this dataset.

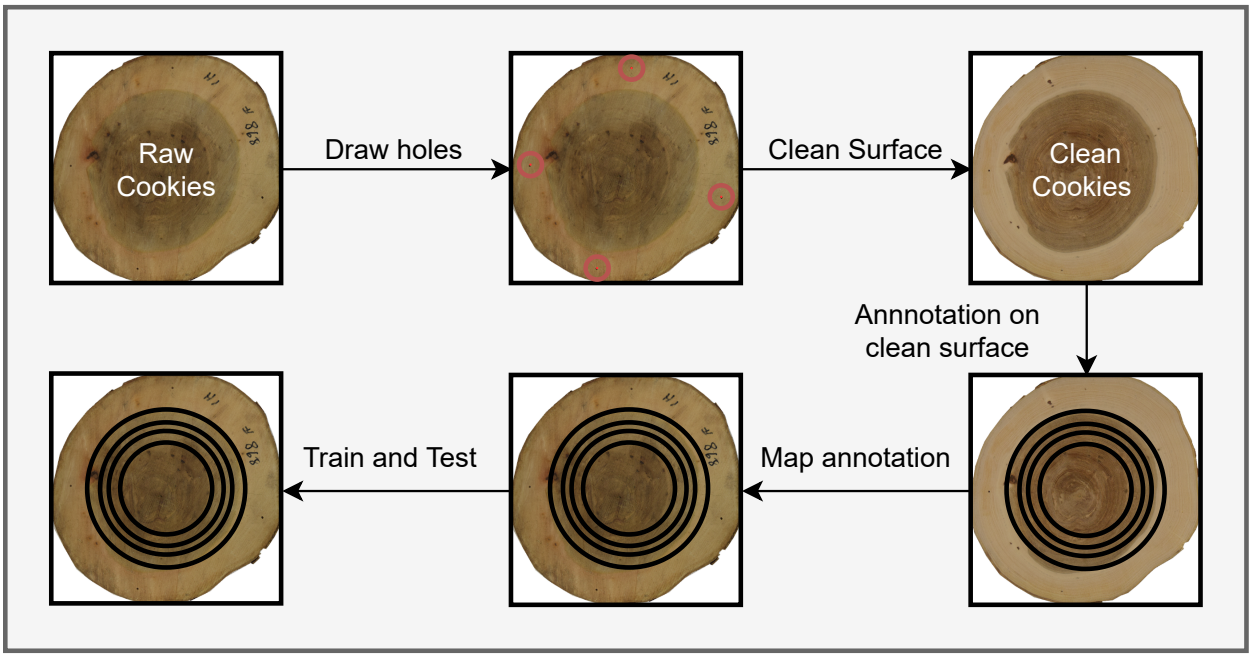


Figure 4.2. Flow chart of the whole process.

First, we drilled four holes on each rough wood surface (surface cut with a chainsaw) for later alignment purposes. Then we put the rough wood cookies inside the image-taking frame (Figure. 4.3). The area of the frame is 95 × 95 cm and the height to the Lens of the camera is 92 cm. There were in total six fill lights, four pointing downwards and two parallel to the cross-sections to reduce reflected light spots. We used autofocus (AF) and auto white balance (AWB) to obtain better quality images during the image-taking process, so the pixel area and contrast varies slightly for each wood cookie. An image was taken of each side of the cookie.

After initial rough-surface images were taken , each cookie surface was machined flat using a Model 40 Thermwood CNC router and a flycut tool until both sides were flat. The cookies that were surfaced using the Model 40 Thermwood CNC router had to meet a height requirement due to the 22.3 cm clearing on the router carriage. The cookies could be no greater than 19.4 cm tall because each cookie laid on a 1.59 cm particleboard, and on top of the particleboard was a 1.91 cm sheet of plywood. Three programs were run depending on the diameter of each cookie. Program 1 was used on cookies 30.5 cm in diameter or less. Program 2 was used for cookies 33 to 46 cm in diameter. Finally, program 3 was used for

Table 4.1. Summary of wood samples.

Species	Common name	Cookies #
Fraxinus spp.	Ash	11
Tilia americana	Basswood	11
Juglans nigra	Black walnut	14
Prunus serotina	Cherry	13
Celtis occidentalis	Hackberry	10
Acer saccharum	Hard maple	14
Carya spp.	Hickory	12
Quercus rubra	Red oak	14
Acer saccharinum	Soft maple	12
Quercus spp.	White oak	14
Liriodendron tulipifera	Yellow poplar	11
Total		**136**

cookies 48.3 cm in diameter and up. Processing any individual side of one cookie could take anywhere from 5 to 25 minutes, depending on how rough the individual cut was. Having three different programs allowed the CNC operator to save time and not waste any motions that were not cut. The maximum material that could be taken off in one pass from the router was 0.51 cm. So after finished run the Z axis height was adjusted by -0.51 cm. This process was repeated until one side of the cookie was completely flat. Once flat the cookies were turned over to begin surfacing the other side. The previous steps were repeated until the entire cookie was flat on both sides. By the time the CNC processing is finished, the cookies were flat on both sides but left with small machine marks that were going to be sanded off in the final step.

With the hardwood cookies flat on both sides, they were sanded with a 60-grit sanding belt using a TimeSaver Series 1300 wide belt sander. Using the TimeSaver belt sander the cookies were sanded until the machine marks from the flycut router bit were eliminated. Much like the Thermwood router the TimeSaver had a height requirement. No cookies could be bigger than 12.7 cm tall. So before even being considered to sand, the CNC operator machined the cookies down to less than 12.7 cm in height. With each pass through the TimeSaver using 60 grit sandpaper the maximum amount of material the operator could

Figure 4.3. Frame to take images.

take off was -0.051 cm. Once finished sanding on both sides, a brush, compressed air and shop vacuum were used remove dust from the surface as necessary to achieve clear cross sections for the clean-surface pictures.

The clean-surface picture taking was similar to the rough-surface picture taking process, with the addition that for several species, both dry and wetted surface pictures were taken. For many species, wetting the surface with water could remove dust and make the surface features more visible. However, for Black Walnut, the absorbed water will lead to a darker surface and more invisible ring edges.

4.2.2 Data Annotation

The data annotation process consisted of two steps. We first manually labeled 4 anchors (East, West, North and South) and the pith coordinates in order to align the annotation of the both rough and clean images. Then we focused on the clean images, slice one cut started from the pith of the cookie to the edge of the image, and manually annotate ring edges. Here we only annotated those ring edges with clean boundary to avoid potential false positives samples created by human considering that the ring edges are arrange tightly. Slightly different from [80] who annotated only one point per ring as ground truth, we used

two or three points to represent either an arc or a line. Figure. 4.4 shows the annotation example. This was time consuming task that typically took 5 minutes to fully label one slice of each wood cookie.

Figure 4.4. Annotation Sample of White Oak (*Quercus spp.*).

4.2.3 Methods

We define this tree-ring measurement as a semantic segmentation task. We want to obtain a model $f : I \rightarrow P$, where $I \in \mathbb{R}^{L \times W}$ is the input ring image. Here L and W are the length and width for the image. In this research, we set W to 128 since this width is also the annotation width we used. $P \in \mathbb{R}^{L \times 3}$ is the prediction heat map of ring edges and only outputs the left mid and right point of each pixel row. Slightly different from [80], [87], which tried to detect all edges of rings by using segmentation methods and output a mask of a similar shape to the input, we simplify the task since the rough images may not even have full clean growth ring edges by raw eyes. To point out, our annotation is

As the problem formed as semantic segmentation task, we could follow common encoder-decoder architecture, and used some the-state-of-art encoder e.g., ResNet [9], EfficientNet [91] and MobileNet [11] and FPN [92] as decoders in this research. Figure. 4.5 is the illustration of the architecture used in this research.

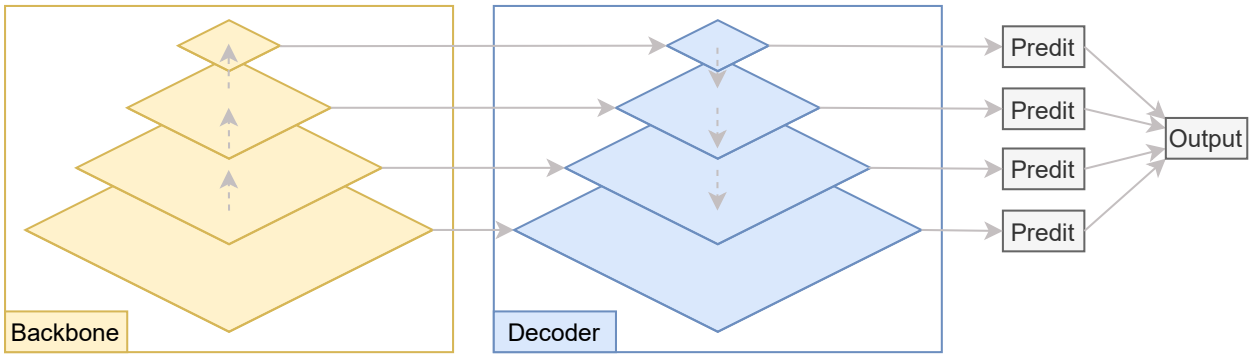


Figure 4.5. Illustration of the FPN architecture.

4.2.4 Post-processing and Evaluation

Post-processing is used to finalize rings. We used a simple local maximum once the heatmap is obtained. Also, we remove some local maximum that the likelihood is less than 0.05.

To simplify the evaluation process, we also followed the proposed evaluation methods in [80]. They defined the correctly detected tree-ring boundaries as true positives (TP), tree-ring boundaries omitted by models as false negatives (FN), and false boundaries introduced by models as false positives (FP). If the predicted ring edge is within the certain distance of the ground truth, we consider it is a true positive; otherwise, it is a false prediction. We need to point out that the certain distance is a sensitive parameter, and in [80], they use a pixel distance of 2 for evaluation. We also inherit the recall (REC), which is called sensitivity as well, and precision (PREC) metrics that Eqs (4.1) and (4.2) determine, respectively. Finally, we use the balanced F score (F_1) to evaluate the performance, which is the harmonic mean of precision and recall (Eqs. 4.3).

$$REC = \frac{TP}{TP + FN} \tag{4.1}$$

$$PREC = \frac{TP}{TP + FP} \tag{4.2}$$

$$F_1 = 2 \cdot \frac{REC \cdot PREC}{REC + PREC} \tag{4.3}$$

4.3 Results and Discussion

In this research, we used a desktop computer equipped with a quad-core × Intel Core i7-8700K running at 3.70GHz, 32 GB of memory, and × NVIDIA GeForce RTX 1080 Ti GPU. We trained all models with the same hyper parameters. We used binary cross entry loss to train the models, and Table. 4.2 shows the details of all necessary parameters in this research. We split the data into 80% from training and 20% for testing without overlapping. In the following section, we will discuss some factors that will affect the performance of our methods.

Table 4.2. Implementation Details.

Name	**Parameter**
Input size	1024×128
Output size	256×3
Optimizer	Adam
Learning rate	10^{-5}
Mini batch size	16
Max iteration	2000

First, we discuss the performance over different species in Table. 4.3. Overall, for most species, the F_1 scores range from 0.7 to 0.9, showing that the model could discover ring edges. However, for some species, e.g., White oak and Hard maple, the F_1 scores are pretty low, indicating the potential failure of the models. Since we only trained one model for all species, that failure might come from the anatomy structure. White oak and Hard maple are with abundant and banded parenchyma cells. Those features might make the model detect more false ring edge boundaries.

Table. 4.4 shows the impacts of surface cleanliness. It meets our assumption that ring edges from the rough surface are the hardest to be identity. Also, wetting the surface makes more ring edges visible both by raw human eyes and the models.

Table 4.3. The performance over different species. The backbone encoder is Efficient-b0, the surface is clean and dry and the color space is V.

Species	Common name	TP	FN	FP	REC	PREC	F_1
Fraxinus spp.	Ash	126	6	118	0.95	0.52	0.67
Tilia americana	Basswood	207	32	76	0.87	0.73	0.79
Juglans nigra	Black walnut	251	16	68	0.94	0.79	0.86
Prunus serotina	Cherry	201	26	107	0.89	0.65	0.75
Celtis occidentalis	Hackberry	68	9	18	0.88	0.79	0.83
Acer saccharum	Hard maple	36	6	36	0.86	0.50	0.63
Carya spp.	Hickory	169	18	15	0.90	0.92	0.91
Quercus rubra	Red oak	276	30	126	0.90	0.69	0.78
Acer saccharinum	Soft maple	326	98	89	0.77	0.79	0.78
Quercus spp.	White oak	149	21	145	0.88	0.51	0.64
Liriodendron tulipifera	Yellow poplar	82	11	78	0.88	0.51	0.65

Table 4.4. The impacts of surface cleanliness. Note that the sum of true positive and false positives is now a constant due to the image alignment and crop reasons.

Surface	TP	FN	FP	SEN	PREC	F_1
Rough	1450	863	1563	0.63	0.48	0.54
Clean Dry	1508	550	1072	0.73	0.58	0.65
Clean Wet	1891	273	876	0.87	0.68	0.77

As in the [80], use proper color space will enhance the accuracy of the prediction. We followed the same setup and compared the impact of utilizing RGB, HSV, and V (relative lightness or darkness of a color) color space. In Table. 4.5, our experiments once demonstrated that using V channel will somehow reduce overfitting by applying a robust prior knowledge that ring edges are often shown that when the relative lightness changes raptly.

We tested several different backbone encoders here in Table. 4.6. Since the current labeled dataset size is small, we selected relative lightweight models here. It is not surprising that Efficient-b0 is the best among all three encoders since Efficient was shown as highly-performing in many other tasks.

This approach has several limitations. First, the labeled dataset size is relatively small, and we just utilized one rectangular area per sample. Specifically for some rough samples,

Table 4.5. The impacts of Color Space. The backbone encoder is Efficient-b0, the surface is clean and wet.

Color Space	TP	FN	FP	SEN	PREC	F_1
RGB	1633	236	788	0.87	0.67	0.76
HSV	1686	421	927	0.80	0.65	0.71
V	1891	273	876	0.87	0.68	0.77

Table 4.6. Choices of Encoder. The surface is clean and wet and the color space is V.

Encoder	TP	FN	FP	SEN	PREC	F_1
MobileNet-V2	1552	634	773	0.71	0.67	0.69
ResNet-18	1449	522	812	0.74	0.64	0.68
Efficient-b0	1891	273	876	0.87	0.68	0.77

Ring edges within those areas might become invisible even for rough eyes. To better utilize these images in the future, we plan to annotate more rings edges and even try to annotate the entire cross-section surface instead of just selected sections. There are other studies, e.g., [87], [93], that try to detect pith automatically. However, we ignored the influence of pith locations and manually labeled pith. In the real world, the correct localization of the ring pith is important because the image slices should start from the pith. Detecting the pith is even more difficult for samples with multiple occurrences of pith. Last, we used a simple semantic segmentation method to solve the problem aligned with most references to demonstrate its purpose. There are several advanced ways to enhance the performance, e.g., using object detection methods with key point identification. In the future, we will investigate other methods for the ring edge detection task.

4.4 Conclusion

In this paper, we first introduce a new dataset of images of hardwood species annotated for tree ring detection. It currently consists of 136 common hardwood species from Indiana, USA. For better future industry utilization, we incorporated both clean and rough images into this study.

We applied the state-of-the-art semantic segmentation models to the dataset and achieved an overall F_1 score of 0.77 for clean images, which shows the power of the convolutional neural network. However, this method failed for rough images in alignment with our assumptions.

5. ZERO SHOT LEARNING FOR WOOD

Acknowledgement

A full version of this chapter will be submitted when ready.

Abstract

The application of Convolutional Neural Network in developing algorithms for machine wood identification has received significant attention recently. These techniques commonly use many correctly identified (labeled) images to train a neural network (CNN). This approach's limiting prerequisite is the cost of acquisition and the limited availability of a sufficient number (typically hundreds or thousands) of images per species, and the verification of species labels. In this study, we apply the concept of zero-shot learning to identify the classes of features that are not present during neural network training. A typical approach in the zero-shot learning method is to learn the attribute vector instead of the categorical label. This approach mimics the traditional way of human identification of wood by its features.

Wu, F., Liu, Y., Gazo, R., Benes, B., & Haviarova, E. (2021).

5.1 Introduction

Wood is a uneatable resource, which supplies are not infinite [94]. There are about 60,000 wood species that are known to science, and the international trade in round wood is worth around 17 billions US dollars in 2019 [95], [96]. The entire economic value of wood is often under-estimated, especially at the places of origin. There is a large amount of illegal trade mainly involving rosewood, which is a trade term for a wide range of tropical hardwoods [97]. From 2014 to 2018, rosewood accounted for 31.7% of the total monetary value of illegal wildlife trade seizures [98]. The whole supply chain of forest products is intricate, usually involving multiple countries. Practically, it is not reliable to use the existing certification and documentation methods to trace the origin of forest products [94]. Therefore, it is important to identify the wood species from the source, midpoint, and destination to meet the legislative requirement and human concerns about sustainability. Besides, commercially sourcing wood species is essential to functioning economy since different wood species have different physical, mechanical and chemical properties that affect their final uses and prices.

Genetics-based methods could help to identify wood species and have been successful in many studies [99], [100]. Using a polymerase chain reaction (PCR) to perform DNA analysis, wood scientists could determine the wood species more accurately. However, it is sometimes difficult to extract DNA from wood material since for a live tree, there is only a small amounts of DNA in the xylem and it degrades gradually after harvesting. In addition, using PCR to amplify DNA may fail, probably due to inhibitory substances such as chemical extractives, which are common in many wood species [99]. Apart from DNA-based methods, chemical methods, e.g., mass spectrometry, stable isotopes, and radio-carbon, are showing promise in many studies. However, the cost is relatively high, ranging from 100 to 300 US dollars per sample [101].

Currently, the mainstream ways of wood identification are anatomy-based methods [102]. Macroscopic wood anatomy and microscopic wood anatomy are two categories. The latter is more expensive but more accurate, because it requires microscopy preparation and observation tools [101]. Experts can define the features of macro and/or micro images to find references and determine the genus of wood. For example, the distribution of vessel ele-

ments and the arrangement of parenchyma can often help determine the hardwood genus. Generally speaking, it takes years of training to master this skill and the number of experts is declining globally [103]. To overcome the above problem, computer-aided tools for wood identification have attracted researchers' attention. *InsideWood* is one of the most beneficial online tools for wood identification, it is a database containing feature descriptions and images of wood species, most of which are microscopic images [104]. Other tools to help wood identification focusing on macroscopic images, e.g, *CITESwoodID* and *xylotron* [105], [106].

In recent decades, many studies have focused on automatic wood identification based on wood image by using Convolutional Neutral Networks (CNN) [30], [31], [103], [107]. These studies have relatively satisfactory performance, which shows the potential of deep learning in wood identification. However, all of these studies require relatively large replicates for each class, and their scopes are usually to narrow down the identification to native species. We believe these studies will raise the scientists' attention to collecting more images of each wood species, but collecting data is always time-consuming and labor-intensive. Therefore, we should seek an alternative way to automatic wood identification, which fits most wood collections' current situation. These collections own thousands of species, each containing only a few or even just a single sample and a complete feature description with multiple-entry key, as in *InsideWood* and *CITESwoodID*.

Zero-shot learning in machine learning has been proposed to identify objects from the classes not observed in the training process and predict the classes they belong to. [108]–[110]. The methodology of zero-shot learning is to learn the parameters of observed classes together with their class representations and rely on the representational similarity between class labels, so that images can be classified into new classes during the prediction process. This methodology fits the requirement of current wood collections precisely.

In this paper, we combine the observed classes and non-observed classes by distinguishing the attributes of objects, and applying zero-shot learning to microscopic wood images in *InsideWood*. To our knowledge, this research is the first of its kind to focus on large-scale wood species identification.

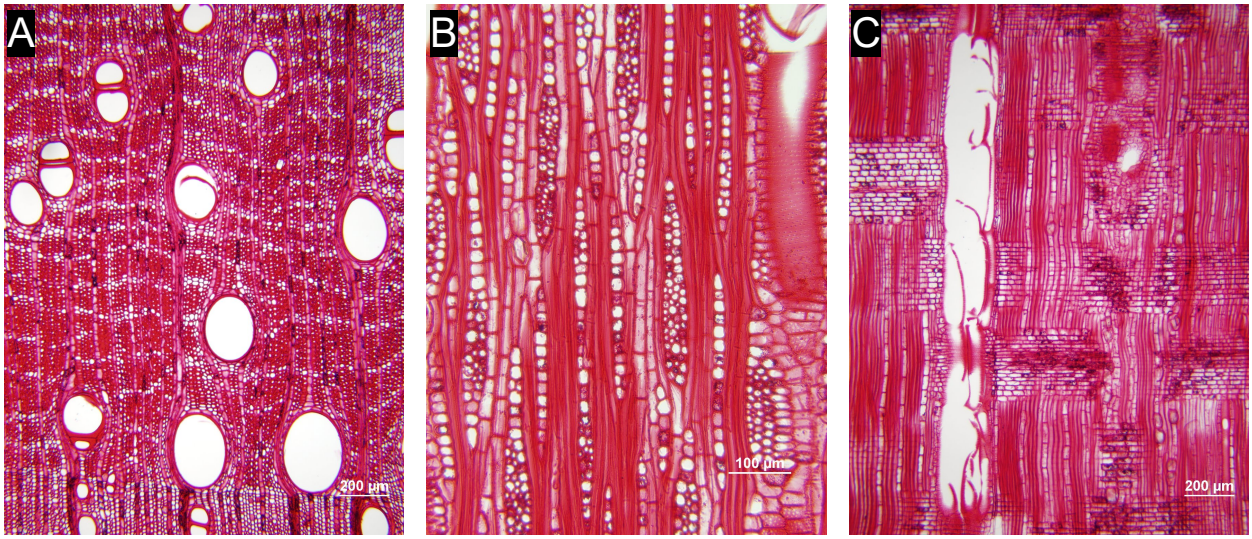


Figure 5.1. A set of sample images of True Hickory (*Carya aquatica*). A, B and C are cross, tangential and radial sections respectively.

5.2 Material and Methods

5.2.1 Data Description

InsideWood is an open resource for wood anatomy research, that can be searched, and which incorporates multiple-entry keys to help wood identification [104]. Its database contains more than 9,400 fossil and modern woody dicots descriptions, representing over 10,000 species and 200 plant families, and provides over 50,000 images [111], most of which are microscopic images. In this study, we focus on a subset of the *InsideWood* collection. Table. 5.1 lists some statistics of the data we use, all from *InsideWood*. Those microscopic images generally follow three views (cross-section, tangential-section, and radial-section). The cross-section is the most commonly used view in wood identification, with 16,778 images. There are 13,176 images of the tangential section and 8,969 images of the radial section. Figure. 5.1 shows some sample images. Those images fall into different views and magnification, and when the magnification information is not well used, the performance will be degraded.

Table 5.1. Data Descriptions.

Name		#
# of Images	**Total**	38,923
	cross section	16,778
	tangential section	13,176
	radial section	8,969
# of Family		254
# of Species		7,426
Magnification		x1 $\sim$ x300
Resuliton		1000x1500

5.2.2 Attributes

We use a subset of International Association of Wood anatomists (IAWA) Hardwood List [40] feature definitions as labeled attributes. We labeled all attributes into three categories, which implies the observation of attributes in a different section. Table. 5.3 shows summaries sub categories of each attributes. The features of growth rings are the most difficult challenging part to learn, it is also in line with the knowledge of experts.

5.2.3 Objective Function

In this research, we applied Direct Attribute Prediction (DAP) [109] to learn probabilistic attributes. Although DAP performs worse than the compatibility learning frameworks [112], it is simple and has a better diagnostic effect. The class label can be calculated by Equation 5.1.

$$f(x) = \text{argmax}_c \prod_{m=1}^{M} \frac{p\left(a_m^c \mid x\right)}{p\left(a_m^c\right)}, \tag{5.1}$$

where

$$p\left(a_m \mid x\right) = \sum_{k=1}^{K} p\left(a_m \mid y_k\right) p\left(y_k \mid x\right), \tag{5.2}$$

Table 5.2. Attribute Accuracy

Section	Feature Group	EfficientNet-b1	ResNet-18	MobileNet-V2
Transverse	Growth Rings	0.665	0.675	0.690
	Vessels	0.836	0.833	0.839
	Tracheids and fibres	0.814	0.811	0.814
	Axial parenchyma	0.798	0.792	0.800
	Rays	0.882	0.877	0.883
Longitudinal	Vessels	0.779	0.781	0.786
	Tracheids and fibres	0.732	0.731	0.743
	Rays	0.765	0.764	0.789
	Secretory elements and cambial variants	0.962	0.965	0.962
	Storied structure	0.941	0.941	0.942
	Mineral inclusions	0.931	0.931	0.928

5.3 Results and Discussion

We built a supervised classification framework based on deep transfer learning neural network. For the main part of the model, we use EfficientNet [91] since it is the state-of-the-art model based on NAS techniques [113]. In the experiment of this paper, we focus on the version with fewer parameters, which is EfficientNet-b1. It should be noted that the backbone of different versions of EfficientNet is the same, only differs in depth and bottleneck block types. In addition, we also compared the deep transfer learning model using MobileNet-V2 and ResNet as the backbone [114].

Data splitting was performed at the level of the plant family. We applied stratified five-folds cross-validations without overlapping (i.e., 20% of the plant family per fold). For each experiment, four folds of data were used during training, and the remaining fold of the data was used as the test set. We further divide the training set (70% of the plant family) and the validation set (10% of the plant family) in the training phase. The training set is used to train models, and the validation set is used to select the final model parameters. In our experiments, the final model parameters are selected based on maximizing the evaluation metrics of the validation set.

We first apply Direct Attribute Prediction (DAP) to investigate the difficulty of learning each attribute. Table. 5.2 show the result. For simplicity, we group all the results into the main categories. In general, we can conclude that zero-shot learning works as desired.

5.4 Conclusion

This preliminary study focused on investigating the potential zero-shot learning to microscopic wood images in *InsideWood*. Right now, we only applied the most straightforward Direct Attribute Prediction (DAP) and showed average accuracy of 80%. This result met our assumption that zero-shot learning could help for large-scale microscopic wood identification.

Table 5.3. Attributes Table.

Feature Group	Codes	Views
Growth Rings	1–2	
Growth ring boundaries distinct	1	cross
Growth ring boundaries indistinct or absent	2	cross
Vessels	3–51, 52–54, 56–59	
Porosity	3–5	cross
Vessel arangement	6–8	cross
Vessel groupings	9–11	cross
Solitary vessel outline	12	cross
Perforation plates	13–19	radial
Intervessel pits: arrangement and size	20–27	tangential
Vestured pits	29	tangential
Vessel - ray pitting	30–35	tangential
Helical thickenings	36–39	tangential
Tangential diameter of vessel lumina	40–45	cross
Vessels per square millimetre	46–50	cross
Mean vessel element length	52–54	tangential
Tyloses and deposits in vessels	56–58	tangential
Wood vesselless	59	cross
Tracheids and fibres	60–73	
Vascular / vasicentric tracheids present	60	cross
Ground tissue fibres	61–64	tangential
Septate fibres and parenchyma–like fibre bands	65–67	
Septate fibres present	65	tangential
Non-septate fibres present	66	tangential
Parenchyma-like fibre bands alternating with ordinary fibres	67	cross
Fibre wall thickness	68–70	cross
Mean fibre lengths	71–73	tangential
Axial parenchyma	75–95	
Axial parenchyma absent or extremely rare	75	cross
Apotracheal axial parenchyma	76–77	cross
Paratracheal axial parenchyma	78–84	cross
Banded parenchyma	85–89	cross
Axial parenchyma cell type / strand length	90–95	cross
Rays	96–117	
Ray width	96–100	tangential
Aggregate rays	101	cross
Ray height	102	tangential
Rays of two distinct sizes	103	tangential
Rays: cellular composition	104–109	radial
Sheath cells	110	tangential
Tile cells	111	cross
Perforated ray cells	112	cross
Disjunctive ray parenchyma cell walls	113	cross
Rays per millimetre	114–116	cross
Wood rayless	117	cross
Storied structure	118–122	
All rays storied	118	tangential
Low rays storied, high rays non–storied.	119	tangential
Axial parenchyma and / or vessel elements storied	120	tangential
Fibres storied	121	tangential
Rays and / or axial elements irregularly storied	122	tangential
Secretory elements and cambial variants	124–135	
Oil and mucilage cells	124–126	
Oil and / or mucilage cells associated with ray parenchyma	124	tangential
Oil and / or mucilage cells associated with axial parenchyma	125	tangential
Oil and / or mucilage cells present among fibres	126	cross
Intercellular canals	127–131	
Axial canals	127–129	cross
Radial canals	130	tangential
Intercellular canals of traumatic origin	131	cross
Tubes / tubules	132	radial
Cambial variants	133–135	cross
Mineral inclusions	136–163	
Prismatic crystals	136–143	tangential
Druses	144–148	tangential
Other crystal types	149–153	tangential
Other diagnostic crystal features	154–158	tangential
Silica	159–163	tangential

6. CONCLUSION

This dissertation demonstrated the potential of using machine learning, especially deep learning methods, in many forestry-related tasks, e.g., wood and bark identifications. Those applications help both the research community and industry in implementing digital forestry strategies.

For machine learning, data is always the first concern. If the availability, quality, and quantity of data and annotation meet specific requirements, the methods will have significance. As our community realizes the effectiveness and importance of machine learning, it is our hope that more publicly available data becomes accessible to researchers.

In the recent decade, there has been an 'All-in-AI' trend that many research areas beginning to apply deep learning methods to their fields of study. Although we admit the potential of deep learning techniques, we need to point out that deep learning is not the solution for all. The relative complexity and the lack of exploratory may limit and impede its adoption in many research areas.

REFERENCES

[1] T. M. Mitchell, *Machine Learning*. 1997, ISBN: 0070428077 9780070428072 0071154671 9780071154673.

[2] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of research and development*, vol. 3, no. 3, pp. 210–229, 1959.

[3] R. Dechter, "Learning while searching in constraint-satisfaction-problems," in *Proceedings of the Fifth AAAI National Conference on Artificial Intelligence*, ser. AAAI'86, Philadelphia, Pennsylvania, 1986, pp. 178–183.

[4] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998, ISSN: 1558-2256. DOI: 10.1109/5.726791.

[5] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[6] T. Hastie and R. Tibshirani, *Statistical learning & data mining iv*. [Online]. Available: https://github.com/ledell/sldm4-h2o.

[7] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. 32nd Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[8] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *J. Mach. Learn. Technol.*, vol. 15, no. 1, pp. 1929–1958, 2014.

[9] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, Jun. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[10] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.

[11] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4510–4520.

[12] D. E. Rumelhart and J. L. McClelland, "On learning the past tenses of english verbs," 1986.

[13] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[14] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[16] C. Olah, *Understanding LSTM Networks – colah's blog*. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

[17] S. Boudra, I. Yahiaoui, and A. Behloul, "A comparison of multi-scale local binary pattern variants for bark image retrieval," in *International Conference on Advanced Concepts for Intelligent Vision Systems*, Springer, 2015, pp. 764–775.

[18] M. Sulc and J. Matas, "Kernel-mapped histograms of multi-scale lbps for tree bark recognition," in *2013 28th International Conference on Image and Vision Computing New Zealand (IVCNZ 2013)*, IEEE, 2013, pp. 82–87.

[19] S. Fiel and R. Sablatnig, "Automated identification of tree species from images of the bark, leaves or needles," Jan. 2010.

[20] A. Bressane, J. A. Roveda, and A. C. Martins, "Statistical analysis of texture in trunk images for biometric identification of tree species," *Environmental monitoring and assessment*, vol. 187, no. 4, p. 212, 2015.

[21] A. A. Othmani, C. Jiang, N. Lomenie, J.-M. Favreau, A. Piboule, and L. F. L. Y. Voon, "A novel computer-aided tree species identification method based on burst wind segmentation of 3d bark textures," *Machine Vision and Applications*, vol. 27, no. 5, pp. 751–766, 2016.

[22] Z.-K. Huang, Z.-H. Quan, and J.-X. Du, "Bark classification based on contourlet filter features using rbpnn," in *International Conference on Intelligent Computing*, Springer, 2006, pp. 1121–1126.

[23] T. Mizoguchi, A. Ishii, H. Nakamura, T. Inoue, and H. Takamatsu, "Lidar-based individual tree species classification using convolutional neural network," in *Videometrics, Range Imaging, and Applications XIV*, International Society for Optics and Photonics, vol. 10332, 2017, 103320O.

[24] M. Cimpoi, S. Maji, and A. Vedaldi, "Deep filter banks for texture recognition and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 3828–3836.

[25] L. Sharan, R. Rosenholtz, and E. Adelson, "Material perception: What can you see in a brief glance?" *Journal of Vision*, vol. 9, no. 8, pp. 784–784, 2009.

[26] E. Yuliastuti, S. R. Sasi, *et al.*, "Compact computer vision system for tropical wood species recognition based on pores and concentric curve," in *2013 3rd International Conference on Instrumentation Control and Automation (ICA)*, IEEE, 2013, pp. 198–202.

[27] R. Yusof, N. R. Rosli, and M. Khalid, "Using gabor filters as image multiplier for tropical wood species recognition system," in *2010 12th International Conference on Computer Modelling and Simulation*, IEEE, 2010, pp. 289–294.

[28] M. Khalid, R. Yusof, and A. S. M. Khairuddin, "Tropical wood species recognition system based on multi-feature extractors and classifiers," in *2011 2nd International Conference on Instrumentation Control and Automation*, IEEE, 2011, pp. 6–11.

[29] B. Sugiarto, E. Prakasa, R. Wardoyo, R. Damayanti, L. M. Dewi, H. F. Pardede, Y. Rianto, *et al.*, "Wood identification based on histogram of oriented gradient (hog) feature and support vector machine (svm) classifier," in *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, IEEE, 2017, pp. 337–341.

[30] P. L. Paula Filho, L. S. Oliveira, S. Nisgoski, and A. S. Britto, "Forest species recognition using macroscopic images," *Mach. Vis. Appl.*, vol. 25, no. 4, pp. 1019–1031, 2014. DOI: 10.1007/s00138-014-0592-7.

[31] P. Ravindran, A. Costa, R. Soares, and A. C. Wiedenhoeft, "Classification of cites-listed and other neotropical meliaceae wood images using convolutional neural networks," *Plant methods*, vol. 14, no. 1, p. 25, 2018.

[32] M. P. E. Apolinario Lainez, S. G. Huamán Bustamante, and G. C. Orellana, "Deep learning applied to identification of commercial timber species from peru," in *2018 IEEE XXV International Conference on Electronics, Electrical Engineering and Computing (INTERCON)*, Aug. 2018, pp. 1–4. DOI: 10.1109/INTERCON.2018.8526457.

[33] G. van Rossum, "Python tutorial," Centrum voor Wiskunde en Informatica (CWI), Amsterdam, Tech. Rep. CS-R9526, May 1995.

[34] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds., Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[35] R. Gazo, L. Wells, V. Krs, and B. Benes, "Validation of automated hardwood lumber grading system," *Comput. Electron. Agric.*, vol. 155, pp. 496–500, 2018. DOI: 10.1016/j.compag.2018.06.041.

[36] L. Wells, R. Gazo, R. Del Re, V. Krs, and B. Benes, "Defect detection performance of automated hardwood lumber grading system," *Comput. Electron. Agric.*, vol. 155, pp. 487–495, Dec. 2018, ISSN: 0168-1699. DOI: 10.1016/j.compag.2018.09.025.

[37] R. M. Haralick, K. Shanmugam, and I. H. Dinstein, "Textural features for image classification," *IEEE Trans. Syst. Man Cybern. Syst.*, vol. SMC-3, no. 6, pp. 610–621, 1973. DOI: 10.1109/TSMC.1973.4309314.

[38] T. Ojala, M. Pietikäinen, and D. Harwood, "A comparative study of texture measures with classification based on featured distributions," *Pattern Recognit.*, vol. 29, no. 1, pp. 51–59, 1996, ISSN: 0031-3203. DOI: 10.1016/0031-3203(95)00067-4.

[39] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996. DOI: 10.1038/381607a0.

[40] V. Alfonso, P. Baas, S. Carlquist, J. Chimelo, V. Coradin, P. Détienne, P. Gasson, D. Grosser, J. Ilic, K. Kuroda, R. Miller, K. Ogata, H. Richter, B. Welle, and E. Wheeler, "Iawa list of microscopic features for hardwood identification: With an appendix on non-anatomical information," *IAWA J.*, vol. 10, pp. 221–358, Jan. 1989. DOI: 10.1163/22941932-90000496.

[41] J. Martins, L. Oliveira, S. Nisgoski, and R. Sabourin, "A database for automatic classification of forest species," *Mach. Vis. Appl.*, vol. 24, no. 3, pp. 567–578, 2013. DOI: 10.1007/s00138-012-0417-5.

[42] R. Geirhos, P. Rubisch, C. Michaelis, M. Bethge, F. A. Wichmann, and W. Brendel, "Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness.," in *Proc. 6th Int. Conf. Learn. Represent.*, 2019.

[43] J. Settle and C. Gonso, "2020 indiana forest products price report and trend analysis," Indiana Department of Natural Resources, Tech. Rep., 2020.

[44] M. Stone, "Cross-validatory choice and assessment of statistical predictions," *J. R. Stat. Soc. Series. B. Stat. Methodol.*, vol. 36, no. 2, pp. 111–133, 1974.

[45] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd Int. Conf. Learn. Represent.*, 2015.

[46] T. Valérie and J. Marie-Pierre, "Tree species identification on large-scale aerial photographs in a tropical rain forest, French Guiana—application for management and conservation," *Forest Ecology and Management*, vol. 225, no. 1, pp. 51–61, Apr. 2006, ISSN: 0378-1127. DOI: 10.1016/j.foreco.2005.12.046.

[47] H. L. Hadlich, F. M. Durgante, J. dos Santos, N. Higuchi, J. Q. Chambers, and A. Vicentini, "Recognizing Amazonian tree species in the field using bark tissues spectra," *Forest Ecology and Management*, vol. 427, pp. 296–304, Nov. 2018, ISSN: 0378-1127. DOI: 10.1016/j.foreco.2018.06.002.

[48] H. Liu, P. Dong, C. Wu, P. Wang, and M. Fang, "Individual tree identification using a new cluster-based approach with discrete-return airborne LiDAR data," *Remote Sensing of Environment*, vol. 258, p. 112 382, Jun. 2021, ISSN: 0034-4257. DOI: 10.1016/j.rse.2021.112382.

[49] T. Hellström, P. Lärkeryd, T. Nordfjell, and O. Ringdahl, "Autonomous Forest Vehicles: Historic, envisioned, and state-of-the-art," *International Journal of Forest Engineering*, vol. 20, no. 1, pp. 31–38, Jan. 2009, ISSN: 1494-2119. DOI: 10.1080/14942119.2009.10702573.

[50] J. Ido and T. Saitoh, "CNN-based tree species identification from bark image," in *Tenth International Conference on Graphics and Image Processing (ICGIP 2018)*, vol. 11069, International Society for Optics and Photonics, May 2019, 110693G. DOI: 10.1117/12.2524213.

[51] M. Carpentier, P. Giguère, and J. Gaudreault, "Tree Species Identification from Bark Images Using Convolutional Neural Networks," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Oct. 2018, pp. 1075–1081. DOI: 10.1109/IROS.2018.8593514.

[52] A. Wendel, S. Sternig, and M. Godec, "Automated identification of tree species from images of the bark, leaves and needles," in *16th Computer Vision Winter Workshop*, Citeseer, 2011, p. 67.

[53] M. Šulc and J. Matas, "Fine-grained recognition of plants from images," *Plant Methods*, vol. 13, no. 1, p. 115, Dec. 2017, ISSN: 1746-4811. DOI: 10.1186/s13007-017-0265-4.

[54] Zheru Chi, Houqiang Li, and Chao Wang, "Plant species recognition based on bark patterns using novel Gabor filter banks," in *International Conference on Neural Networks and Signal Processing, 2003. Proceedings of the 2003*, vol. 2, Dec. 2003, 1035–1038 Vol.2. DOI: 10.1109/ICNNSP.2003.1281045.

[55] Yuan-Yuan Wan, Ji-Xiang Du, De-Shuang Huang, Zheru Chi, Yiu-Ming Cheung, Xiao-Feng Wang, and Guo-Jun Zhang, "Bark texture feature extraction based on statistical texture analysis," in *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004.*, Oct. 2004, pp. 482–485. DOI: 10.1109/ISIMP.2004.1434106.

[56] Jiatao Song, Zheru Chi, Jilin Liu, and Hong Fu, "Bark classification by combining grayscale and binary texture features," in *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing, 2004.*, Oct. 2004, pp. 450–453. DOI: 10.1109/ISIMP.2004.1434097.

[57] S. Bertrand, R. B. Ameur, G. Cerutti, D. Coquin, L. Valet, and L. Tougne, "Bark and leaf fusion systems to improve automatic tree species recognition," *Ecological Informatics*, vol. 46, pp. 57–73, 2018.

[58] S. Boudra, I. Yahiaoui, and A. Behloul, "Plant identification from bark: A texture description based on statistical macro binary pattern," in *2018 24th International Conference on Pattern Recognition (ICPR)*, IEEE, 2018, pp. 1530–1535.

[59] S. Fekri-Ershad, "Bark texture classification using improved local ternary patterns and multilayer neural network," *Expert Systems with Applications*, vol. 158, p. 113 509, 2020.

[60] V. Remeš and M. Haindl, "Bark recognition using novel rotationally invariant multispectral textural features," *Pattern Recognition Letters*, vol. 125, pp. 612–617, 2019.

[61] J. Ido and T. Saitoh, "Automatic tree species identification from natural bark image," in *Eleventh International Conference on Graphics and Image Processing (ICGIP 2019)*, vol. 11373, International Society for Optics and Photonics, Jan. 2020, p. 1 137 305. DOI: 10.1117/12.2557187.

[62] D. Misra, C. Crispim-Junior, and L. Tougne, "Patch-based cnn evaluation for bark classification," in *European Conference on Computer Vision*, Springer, 2020, pp. 197–212.

[63] M. Robert, P. Dallaire, and P. Giguère, "Tree bark re-identification using a deep-learning feature descriptor," in *2020 17th Conference on Computer and Robot Vision (CRV)*, IEEE, 2020, pp. 25–32.

[64] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

[65] G. Hinton, O. Vinyals, and J. Dean, "Distilling the Knowledge in a Neural Network," *arXiv:1503.02531 [cs, stat]*, Mar. 2015.

[66] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge Distillation: A Survey," *International Journal of Computer Vision*, vol. 129, no. 6, pp. 1789–1819, Jun. 2021, ISSN: 1573-1405. DOI: 10.1007/s11263-021-01453-z.

[67] R. Müller, S. Kornblith, and G. E. Hinton, "When does label smoothing help?" In *Advances in Neural Information Processing Systems*, 2019, pp. 4694–4703.

[68] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, Mar. 1951, ISSN: 0003-4851, 2168-8990. DOI: 10.1214/aoms/1177729694.

[69] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[70] S. Boudra, I. Yahiaoui, and A. Behloul, "A set of statistical radial binary patterns for tree species identification based on bark images," *Multimedia Tools and Applications*, pp. 1–32, 2020.

[71] M. Švab, "Computer-vision-based tree trunk recognition," Ph.D. dissertation, Bsc Thesis,(Mentor: doc. dr. Matej Kristan), Fakulteta za racunalništvo in, 2014.

[72] R. Lakmann, "Barktex benchmark database of color textured images," *Koblenz-Landau University*, 1998.

[73] R. Ratajczak, S. Bertrand, C. Crispim-Junior, and L. Tougne, "Efficient bark recognition in the wild," in *International conference on computer vision theory and applications (VISAPP 2019)*, 2019.

[74] A. E. Douglass, *Climatic cycles and tree-growth*. Carnegie Institution of Washington, 1919, vol. 2.

[75] C. W. Ferguson, "Bristlecone pine: Science and esthetics: A 7100-year tree-ring chronology aids scientists; old trees draw visitors to california mountains," *Science*, vol. 159, no. 3817, pp. 839–846, 1968, ISSN: 0036-8075. DOI: 10.1126/science.159.3817.839.

[76] V. C. LaMarche, "Tree-ring evidence of past climatic variability," *Nature*, vol. 276, no. 5686, pp. 334–338, 1978, ISSN: 1476-4687. DOI: 10.1038/276334a0.

[77] J. Ballesteros, J. Bodoque, A. Díez-Herrero, M. Sanchez-Silva, and M. Stoffel, "Calibration of floodplain roughness and estimation of flood discharge based on tree-ring evidence and hydraulic modelling," *Journal of Hydrology*, vol. 403, no. 1-2, pp. 103–115, 2011, ISSN: 0022-1694. DOI: 10.1016/j.jhydrol.2011.03.045.

[78] G. C. Wiles, P. E. Calkin, and G. C. Jacoby, "Tree-ring analysis and quaternary geology: Principles and recent applications," *Geomorphology*, vol. 16, no. 3, pp. 259–272, 1996, ISSN: 0169-555X. DOI: 10.1016/s0169-555x(96)80005-5.

[79] K. Norell, "Automatic counting of annual rings on pinus sylvestris end faces in sawmill industry," *Computers and Electronics in Agriculture*, vol. 75, no. 2, pp. 231–237, 2011, ISSN: 0168-1699. DOI: 10.1016/j.compag.2010.11.005.

[80] A. Fabijańska and M. Danek, "DeepDendro – a tree rings detector based on a deep convolutional neural network," *Computers and Electronics in Agriculture*, vol. 150, pp. 353–363, 2018, ISSN: 0168-1699. DOI: 10.1016/j.compag.2018.05.005.

[81] W. Lara, F. Bravo, and C. Sierra, "MeasuRing: An R package to measure tree-ring widths from scanned images," *Dendrochronologia*, vol. 34, pp. 43–50, 2015, ISSN: 1125-7865. DOI: 10.1016/j.dendro.2019.125644.

[82] J. Shi, W. Xiang, Q. Liu, and S. Shah, "MtreeRing: An R package with graphical user interface for automatic measurement of tree ring widths using image processing techniques," *Dendrochronologia*, vol. 58, p. 125 644, 2019, ISSN: 1125-7865. DOI: 10.1016/j.dendro.2015.04.002.

[83] P. Soille and L. Misson, "Tree ring area measurements using morphological image analysis," *Canadian Journal of Forest Research*, vol. 31, no. 6, pp. 1074–1083, 2001, ISSN: 0045-5067. DOI: 10.1139/x01-025.

[84] M. Cerda, N. Hitschfeld-Kahler, and D. Mery, "Robust tree-ring detection," in *Advances in Image and Video Technology*, Springer Berlin Heidelberg, 2007, pp. 575–585. DOI: 10.1007/978-3-540-77129-6_50.

[85] A. Fabijańska, M. Danek, J. Barniak, and A. Piórkowski, "Towards automatic tree rings detection in images of scanned wood samples," *Computers and Electronics in Agriculture*, vol. 140, pp. 279–289, 2017, ISSN: 0168-1699. DOI: 10.1016/j.compag.2017.06.006.

[86] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Lecture Notes in Computer Science*. Springer International Publishing, 2015, pp. 234–241. DOI: 10.1007/978-3-319-24574-4_28.

[87] T. Habite, O. Abdeljaber, and A. Olsson, "Automatic detection of annual rings and pith location along norway spruce timber boards using conditional adversarial networks," *Wood Science and Technology*, vol. 55, no. 2, pp. 461–488, 2021, ISSN: 0043-7719. DOI: 10.1007/s00226-021-01266-w.

[88] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017. DOI: 10.1109/cvpr.2017.632.

[89] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, MIT Press, 2014, pp. 2672–2680.

[90] A. Fabijańska, M. Danek, J. Barniak, and A. Piórkowski, "A comparative study of image enhancement methods in tree-ring analysis," in *Image Processing and Communications Challenges 8*, Springer International Publishing, 2017, pp. 69–78. DOI: 10.1007/978-3-319-47274-4_8.

[91] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proceedings of the 36th International Conference on Machine Learning*, C. Kamalika and S. Ruslan, Eds., vol. 97, PMLR, 2019, pp. 6105–6114.

[92] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2017. DOI: 10.1109/cvpr.2017.106.

[93] R. Gazo, J. Vanek, M. Abdul-Massih, and B. Benes, "A fast pith detection for computed tomography scanned hardwood logs," *Computers and Electronics in Agriculture*, vol. 170, p. 105 107, 2020, ISSN: 0168-1699. DOI: 10.1016/j.compag.2019.105107.

[94] P. E. Gasson, C. A. Lancaster, R. Young, S. Redstone, I. A. Miles-Bunch, G. Rees, R. P. Guillery, M. Parker-Forney, and E. T. Lebow, "WorldForestID: Addressing the need for standardized wood reference collections to support authentication analysis technologies; a way forward for checking the origin and identity of traded timber," *Plants, People, Planet*, vol. 3, no. 2, pp. 130–141, 2021, ISSN: 2572-2611. DOI: 10.1002/ppp3.10164.

[95] E. Beech, M. Rivers, S. Oldfield, and P. Smith, "GlobalTreeSearch: The first complete global database of tree species and country distributions," *Journal of Sustainable Forestry*, vol. 36, no. 5, pp. 454–489, Jul. 2017, ISSN: 1054-9811. DOI: 10.1080/10549811.2017.1310049.

[96] FAOSTAT, *Faostat database*, 2021.

[97] A. Irwin, "Tree sleuths are using DNA tests and machine vision to crack timber crimes," *Nature*, vol. 568, no. 7750, pp. 19–21, Apr. 2019. DOI: 10.1038/d41586-019-01035-7.

[98] UNODC, *World wildlife crime report 2020: Trafficking in protected species*, 2020.

[99] H. Abe, U. Watanabe, K. Yoshida, K. Kuroda, and C. Zhang1, "Changes in organelle and DNA Quality, Quantity, and distribution in the wood of Cryptomeria Japonica over long-term storage," *IAWA Journal*, vol. 32, no. 2, pp. 263–272, Jan. 2011, ISSN: 0928-1541, 2294-1932. DOI: 10.1163/22941932-90000056.

[100] L. Jiao, Y. Yin, F. Xiao, Q. Sun, K. Song, and X. Jiang, "Comparative Analysis of two DNA Extraction Protocols from Fresh and Dried wood of Cunninghamia Lanceolata (Taxodiaceae)," *IAWA Journal*, vol. 33, no. 4, pp. 441–456, Jan. 2012, ISSN: 0928-1541, 2294-1932. DOI: 10.1163/22941932-90000106.

[101] E. E. Dormontt, M. Boner, B. Braun, G. Breulmann, B. Degen, E. Espinoza, S. Gardner, P. Guillery, J. C. Hermanson, G. Koch, S. L. Lee, M. Kanashiro, A. Rimbawanto, D. Thomas, A. C. Wiedenhoeft, Y. Yin, J. Zahnen, and A. J. Lowe, "Forensic timber identification: It's time to integrate disciplines to combat illegal logging," *Biological Conservation*, vol. 191, pp. 790–798, Nov. 2015, ISSN: 0006-3207. DOI: 10.1016/j.biocon.2015.06.038.

[102] UNODC, *Best practice guide for forensic timber identification*, 2016.

[103] F. Lens, C. Liang, Y. Guo, X. Tang, M. Jahanbanifard, F. S. C. d. Silva, G. Ceccantini, and F. J. Verbeek, "Computer-assisted timber identification based on features extracted from microscopic wood sections," *IAWA Journal*, vol. 41, no. 4, pp. 660–680, Jul. 2020, ISSN: 0928-1541, 2294-1932. DOI: 10.1163/22941932-bja10029.

[104] E. A. Wheeler, "Inside wood–a web resource for hardwood anatomy," *Iawa Journal*, vol. 32, no. 2, pp. 199–211, 2011.

[105] G. Koch, H.-G. Richter, and U. Schmitt, "Design and application of CITESwoodID Computer-aided identification and description of CITES-protected timbers," *IAWA Journal*, vol. 32, no. 2, pp. 213–220, Jan. 2011, ISSN: 0928-1541, 2294-1932. DOI: 10.1163/22941932-90000052.

[106] J. C. Hermanson and A. C. Wiedenhoeft, "A brief review of machine vision in the context of automated wood identification systems," *IAWA Journal*, vol. 32, no. 2, pp. 233–250, Jan. 2011, ISSN: 0928-1541, 2294-1932. DOI: 10.1163/22941932-90000054.

[107] F. Wu, R. Gazo, E. Haviarova, and B. Benes, "Wood identification based on longitudinal section images by using deep learning," *Wood Science and Technology*, vol. 55, no. 2, pp. 553–563, 2021. DOI: 10.1007/s00226-021-01261-1.

[108] H. Larochelle, D. Erhan, and Y. Bengio, "Zero-data learning of new tasks.," in *AAAI*, vol. 1, 2008, p. 3.

[109] C. H. Lampert, H. Nickisch, and S. Harmeling, "Attribute-based classification for zero-shot visual object categorization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 3, pp. 453–465, Mar. 2013, ISSN: 1939-3539. DOI: 10.1109/TPAMI.2013.140.

[110] M. Rohrbach, M. Stark, and B. Schiele, "Evaluating knowledge transfer and zero-shot learning in a large-scale setting," in *CVPR 2011*, Jun. 2011, pp. 1641–1648. DOI: 10.1109/CVPR.2011.5995627.

[111] E. A. Wheeler, P. E. Gasson, and P. Baas, "Using the insidewood web site: Potentials and pitfalls," *IAWA Journal*, vol. 1, no. aop, pp. 1–51, 2020. DOI: 10.1163/22941932-bja10032.

[112] Z. Akata, F. Perronnin, Z. Harchaoui, and C. Schmid, "Label-embedding for image classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 38, no. 7, pp. 1425–1438, 2015. DOI: 10.1109/TPAMI.2015.2487986.

[113] T. Elsken, J. H. Metzen, F. Hutter, *et al.*, "Neural architecture search: A survey," *Journal of Machine Learning Research*, vol. 20, no. 55, pp. 1–21, 2019.

[114] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *NIPS'32*, 2019, pp. 8024–8035.

VITA

Education

Purdue University *Jan 2018 - Dec 2021*

Ph.D in Forestry

University of Eastern Finland *Sep 2015 - Apr 2018*

M.S. in Wood Material

Nanjing Forestry University *Sep 2011 - Jun 2015*

B.E in Wood Material

University of British Columbia *Aug 2013 - May 2014*

Exchange student in Wood Products Processing

Selected Publications

Wu, F., Gazo, R., Benes, B., & Haviarova, E. (2021). Deep BarkID - A Portable Tree Bark Identification System by Knowledge Distillation. *Eur. J. For. Res.*

Wu, F., Gazo, R., Haviarova, E., & Benes, B. (2021). Wood identification based on longitudinal section images by using deep learning. *Wood Sci. Technol.*

Wu, F., Gazo, R., Haviarova, E., & Benes, B. (2019). Efficient Project Gradient Descent for Ensemble Adversarial Attack. *IJCAJ 19 AIBS workshop*

Wu, F., & Kärenlampi, P.P. (2017). Phase Transition in A Growing Network. *J. Complex Netw.*

Liu, Y., **Wu, F.**, Lyu, C., Liu, X., & Liu, Z. (2021). Behavior2vector: Embedding Users' Personalized Travel Behavior to Vector. *IEEE Trans. Intell. Transp. Syst.*

Selected Competitions with Cash Prize

CVRP 2021 The 2nd Agriculture-Vision Prize Challenge *2021*

2nd place in two tracks 4200 USD

KDD CUP 2020 Reinforcement Learning Competition Track *2020*

1st place in Reposition Track 8000 USD

IJCAI - 19 Alibaba Adversarial AI Challenge-Competition *2019*

1st place in Traget Attack Track 5000 USD