European Journal of Forest Research
https://doi.org/10.1007/s10342-021-01407-7

ORIGINAL PAPER

Deep BarkID: a portable tree bark identification system by knowledge distillation

Fanyou Wu[1] · Rado Gazo[1] · Bedrich Benes[2] · Eva Haviarova[1]

Received: 14 May 2021 / Revised: 14 July 2021 / Accepted: 1 August 2021

Abstract
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.

Keywords Deep learning · Convolutional neural network · Knowledge distillation · Tree bark · Tree identification

Introduction

Species recognition is one of the key steps in management and conservation planning of many forest ecosystems (Valérie and Marie-Pierre 2006; Hadlich et al. 2018; Liu et al. 2021). 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 (Hellström et al. 2009). 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 (Ido and Saitoh 2019), and in public education.

Communicated by Martina Meincken.

✉ Rado Gazo
gazo@purdue.edu

1 Department of Forestry and Natural Resources, Purdue University, West Lafayette, IN 47906, USA

2 Department of Computer Graphics Technology and Computer Science, Purdue University, West Lafayette, IN 47906, USA

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 (Carpentier et al. 2018). 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 (Wendel et al. 2011).

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 (Šulc and Matas 2017). 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 nonlinear (e.g., Multilayer Perceptron (MLP)) classifiers. Chi et al. (2003) proposed a method using Gabor filter banks, and Wan et al. (2004) applied the co-occurrence matrices, histogram, and autocorrelation methods to bark identification. Yuan-Yuan

Published online: 09 August 2021

Wan et al. (2004) reported that adding color features could improve performance, and Song et al. (2004) employed the Grey-Level Co-occurrence Matrix (GLCM) assisted by Long Connection Length Emphasis (LCLE) for bark classification. Bertrand et al. (2018) used handcraft features, considering shape, color, structure, and orientation of bark by using Canny filters, hue histogram, and Gabor filters. Boudra et al. (2018) introduced Termed Statistical Macro-Binary Pattern (SMBP), a variant of Local Binary Pattern that represents the intensity distribution within the macro-structure of large spatial support by one macro-pattern code. Fekri-Ershad (2020) used Local Ternary Patterns (LTP) and then fed them to the Multilayer Perceptron (MLP). Remeš and Haindl (2019) introduced rotationally invariant multi-spectral textural features and reported 90.4% accuracy on BarkNet (Carpentier et al. 2018) 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) (Lecun et al. 1998). CNNs were successfully used for bark identification in several studies (Carpentier et al. 2018; Ido and Saitoh 2019, 2020; Šulc and Matas 2017; Misra et al. 2020; Robert et al. 2020). 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 (Simonyan and Zisserman 2015) or ResNet (He et al. 2016).

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 (Hinton et al. 2015) 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.

Material and methods

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. Figure 1 is the map for Martell Forest. It has a total area of 193 ha, of which 70% is covered by deciduous forest.

Bark image data

We collected 309 images from 61 trees of 10 different species (see Fig. 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 and 60 cm away from the trunk. The Diameter at Breast Height (DBH) varied between 20 and 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 2000 images representing each tree species (see Table 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 (Carpentier et al. 2018) that includes 20 different tree species ranging from 24 to 109 trees per species (see Table 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.

Knowledge distillation

In this paper, 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 even superior performance. Here 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 simplistic yet effective for model compression and has been broadly used in various tasks and applications (Gou et al. 2021).

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, (Gou et al. 2021; Hinton et al. 2015). 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 Müller et al. (2019).

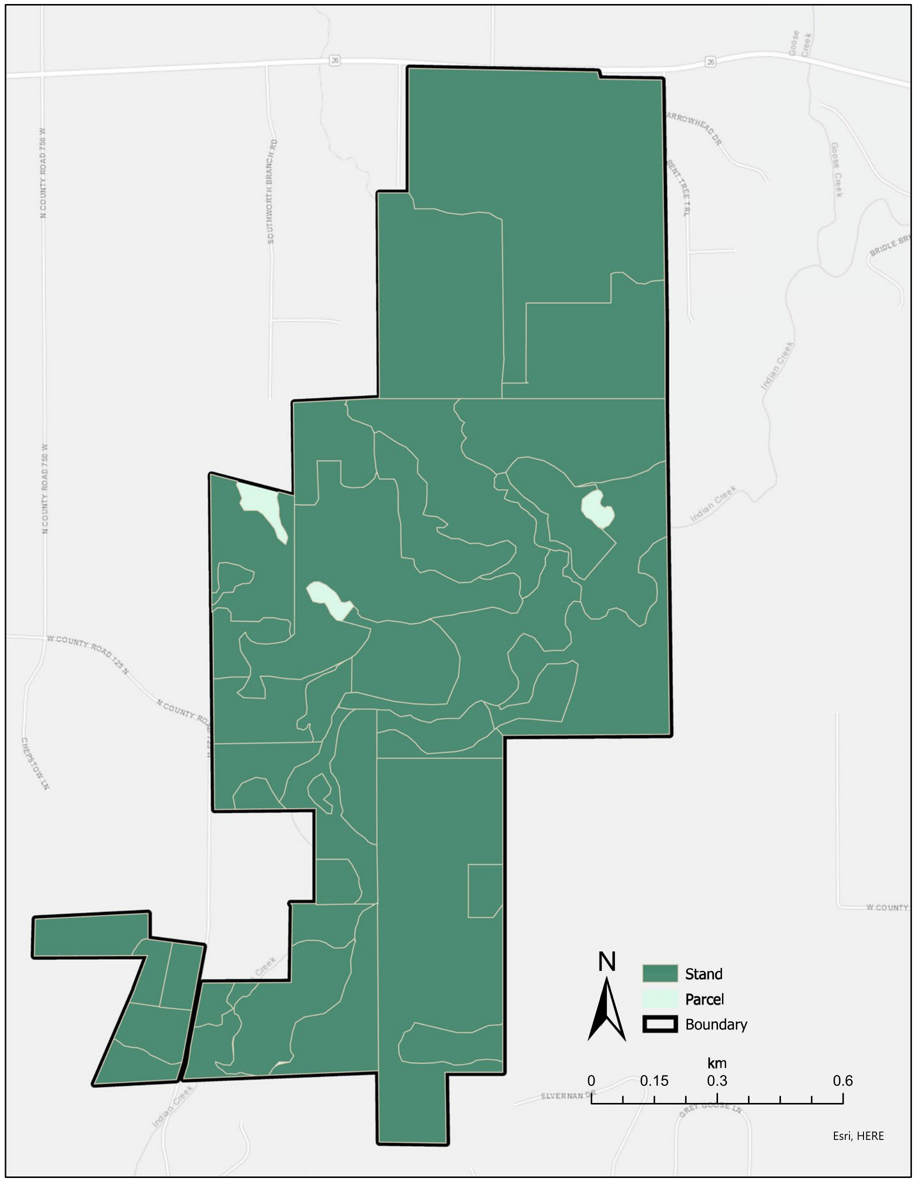

Fig. 1 Map of study area at Martell Forest near West Lafayette, Indiana, USA (40°25′ N; 87°2′ W)

Figure 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 (Kullback and Leibler 1951) and cross-entropy (CE) loss (Goodfellow et al. 2016), 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:

$$q_i = \frac{exp(z_i/T)}{\sum_j exp(z_j/T)}, \tag{1}$$

where z_i and z_j and are the *ith* and *jth* 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 (Hinton et al. 2015).

We use the following loss function to apply knowledge distillation:

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

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

Fig. 2 Sample images for Indiana Bark Dataset

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). KL divergence is a classic metric to measure the difference between two distributions. Sometimes it is also called relative entropy. Formally:

$$\mathcal{L}_{KL}(P, Q) = \sum_i P_i \log\left(\frac{P_i}{Q_i}\right), \tag{3}$$

here P_i and Q_i are the probability of two difference distribution P and Q for i classes.

Deep BarkID

We used two state-of-the-art CNN architectures: ResNet (He et al. 2016) and MobileNet (Sandler et al. 2018). Each is provided in several versions, which differ in the number of layers. Table 2 summarizes those architectures that we used in this paper. We selected shallow versions of these CNNs: ResNet-34 and MobileNet-V2. Since in the original BarkNet Carpentier et al. (2018) 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 (Hinton et al. 2015; Gou et al. 2021).

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`.

Implementation

Inspired by the implementation from (Carpentier et al. 2018), we first downsampled the whole image to the half size of its original resolution to speed up the image reading process. Table 3 lists all details of our implementation. Most hyperparameter are the same in the Carpentier et al. (2018). 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.

Table 1 Species list for BarkNet and Indiana Bark Dataset

Dataset	Species	Common name	Trees	Img	SubImgs
Indiana	*Acer saccharum*	Sugar Maple	6	31	1860
Bark	*Carpinus caroliniana*	American hornbeam	6	30	1800
	Fagus grandifolia	American Beech	5	24	1440
	Liriodendron tulipifera	Yellow Poplar	7	35	2100
	Juglans nigra	Black Walnut	6	30	1800
	Platanus occidentalis	American Sycamore	6	30	1800
	Prunus serotina	Black Cherry	6	30	1800
	Quercus alba	White Oak	6	32	1920
	Quercus rubra	Northern Red Oak	7	35	2100
	Robinia pseudoacacia	Black Locust	6	32	1920
	Total Indiana Bark		61	309	18,540
BarkNet	*Abies balsamea*	Balsam Fir	41	922	28,235
	Acer rubrum	Red Maple	64	1676	48,925
	Acer saccharum	Sugar Maple	81	1999	68,040
	Betula alleghaniensis	Yellow Birch	43	1255	37,325
	Betula papyrifera	White Birch	32	1285	33,892
	Fagus grandifolia	American Beech	41	840	2,3904
	Fraxinus americana	White Ash	61	1472	5,3995
	Larix laricina	Tamarack	77	1902	11,4956
	Ostrya virginiana	American Hophornbeam	29	612	29,723
	Picea abies	Norway Spruce	72	1324	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	1023	25,621
	Populus tremuloides	Quaking Aspen	58	1037	63,247
	Quercus rubra	Northern Red Oak	109	2724	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

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

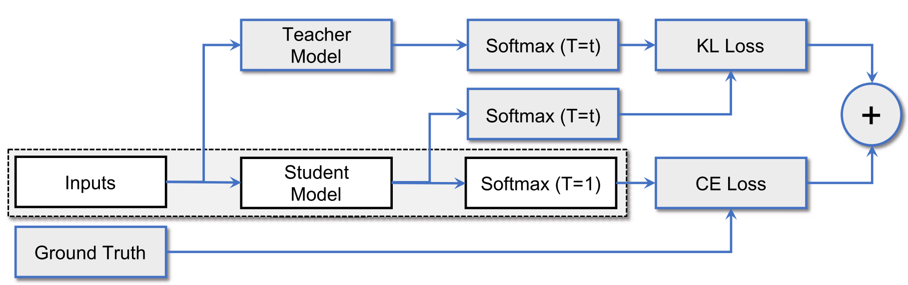

Fig. 3 Visualization of our implementation of knowledge distillation during training. The black box contains the steps performed during the inference. The KL loss and CE loss are the standard Kullback–Leibler divergence and cross-entropy loss, respectively. t will set to 5 in this study as the parameter of temperature T

Table 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
	$64 \times 56 \times 56$	(3×3, 64 residual block)×2	0.148
	$128 \times 28 \times 28$	(3×3, 128 residual block)×4	1.116
	$256 \times 14 \times 14$	(3×3, 256 residual block)×6	6.822
	$512 \times 7 \times 7$	(3×3, 512 residual block)×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
	$16 \times 112 \times 112$	(3×3, 16 inverted residual block)×1	0.001
	$24 \times 56 \times 56$	(3×3, 24 inverted residual block)×2	0.014
	$32 \times 28 \times 28$	(3×3, 32 inverted residual block)×3	0.040
	$64 \times 14 \times 14$	(3×3, 64 inverted residual block)×4	0.184
	$96 \times 14 \times 14$	(3×3, 96 inverted residual block)×3	0.303
	$160 \times 7 \times 7$	(3×3, 160 inverted residual block)×3	0.795
	$320 \times 7 \times 7$	(3×3, 320 inverted residual block)×1	0.474
	$1280 \times 7 \times 7$	(3×3, 1280 inverted residual block)×1	0.412
	$1000 \times 1 \times 1$	average pool, 1000-d fc, softmax	1.281
	Total		3.505

Table 3 Implementation Details

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

Those pretrained models are obtained from TORCHHUB[1]. We employed a slightly different fine-tuning strategy than the original BarkNet paper: we did not freeze the first layer (7×7 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.

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 every single 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.

Results and discussion

The objective of this research was to explore the effectiveness of knowledge distillation. To fully evaluate CNN's performance, we used fivefold cross-validation, which is the same as in Carpentier et al. (2018). We applied fivefold (20% each) cross-validation without overlapping. For each model, fourfold of data (80%) were used during training, and the remaining 20% was used for testing. Data splitting was performed tree-level for BarkNet that no same tree will be used during 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 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 Table 4. Generally, using multiple crops as an ensemble technique will increase the accuracy.

[1] https://pytorch.org/hub/

Table 4 Model accuracy comparison

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 Carpentier et al. (2018)	87.04%	93.88%
	Boudra et al. (2020)	79.10%	-
	Remeš and Haindl (2019)	90.04%	-
	ResNet-34 (ours)	90.02%	94.62%
	MoblieNet-V2	88.45%	93.51%
	`Deep BarkID`	88.75%	94.36%

'-' 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

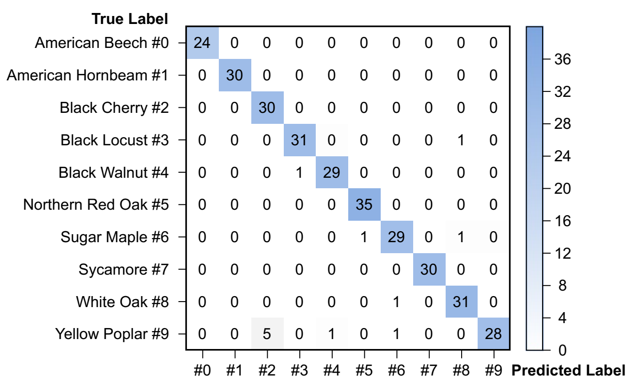

Fig. 4 The confusion matrix for Multiple Crop of `Deep BarkID` using Indianan Bark Dataset

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. Figure 4 shows the confusion matrix of our `Deep BarkID`. It shows that Yellow Poplar (*Liriodendron tulipifera*) was hard to identify and might be confused with Black Cherry (*Prunus serotina*).

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 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 (2019) 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.

Table 5 ResNet-34 model accuracy for different weight initialization using the IBD

Weight initialization	Single crop (%)	Multiple crop (%)
He et al. (2016)	58.60	66.6
ImageNet Pretrained	91.20	97.09%
BarkNet Pretrained	92.24	97.41

Validity of transfer learning

Many studies, e.g., Carpentier et al. (2018); Ido and Saitoh (2019, 2020); Ravindran et al. (2018) 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.

Table 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 Table 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.

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 (Wendel et al. 2011), Trunk12 Dataset (Švab 2014), BarkTex dataset (Lakmann 1998) and Bark101 (Ratajczak et al. 2019). The AFF bark dataset is a collection of the most common Austrian trees. It contains 1,182 bark samples (960 × 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 × 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 × 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)×(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.

Potential of deep learning methods

Several studies confirm the advantage of using deep learning methods for bark identification (Šulc and Matas 2017; Ido and Saitoh 2020; Carpentier et al. 2018). However, some of them point to several drawbacks. Šulc and Matas (2017) 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.

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. 5. However, its user interface could be improved. We also plan to deploy it on Android devices.

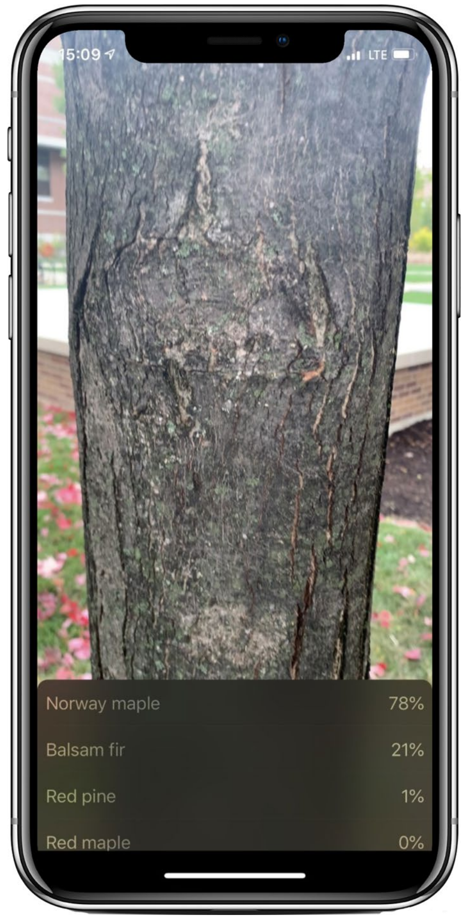

Fig. 5 A snapshot of `Deep BarkID` deployed on an iPhone X

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`.

Funding This research was supported by the Foundation for Food and Agriculture Research Grant ID: 602757 to Benes and McIntire Stennis grant accession no. 1012928 to Gazo from the USDA National Institute of Food and Agriculture. The content of this publication is solely the responsibility of the authors and does not necessarily represent the official views of the respective funding agencies.

Availability of data and material The datasets generated during and/or analyzed during the current study are available in the github repository, https://github.com/wufanyou/DBID.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Code availability The codes generated during and/or analyzed during the current study are available in the github repository, https://github.com/wufanyou/DBID.

References

Bertrand S, Ameur RB, Cerutti G, Coquin D, Valet L, Tougne L (2018) Bark and leaf fusion systems to improve automatic tree species recognition. Ecol Inform 46:57–73. https://doi.org/10.1016/j.ecoinf.2018.05.007

Boudra S, Yahiaoui I, Behloul A (2018) Plant identification from bark: a texture description based on statistical macro binary pattern. In: Proc. 24th Int. Conf. Pattern Recognit. (ICPR 2018), pp 1530–1535. https://doi.org/10.1109/ICPR.2018.8545798

Boudra S, Yahiaoui I, Behloul A (2020) A set of statistical radial binary patterns for tree species identification based on bark images. Multimed Tools Appl. https://doi.org/10.1007/s11042-020-08874-x

Carpentier M, Giguère P, Gaudreault J (2018) Tree species identification from bark images using convolutional neural networks. In: Proc. 2018 IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS 2018), pp 1075–1081. https://doi.org/10.1109/IROS.2018.8593514

Fekri-Ershad S (2020) Bark texture classification using improved local ternary patterns and multilayer neural network. Expert Syst Appl 158:113509. https://doi.org/10.1016/j.eswa.2020.113509

Goodfellow I, Bengio Y, Courville A (2016) Deep Learning. MIT Press, Camberidge

Gou J, Yu B, Maybank SJ, Tao D (2021) Knowledge distillation: a survey. Int J Comput Vis 129(6):1789–1819. https://doi.org/10.1007/s11263-021-01453-z

Hadlich HL, Durgante FM, dos Santos J, Higuchi N, Chambers JQ, Vicentini A (2018) Recognizing Amazonian tree species in the field using bark tissues spectra. For Ecol Manag 427:296–304. https://doi.org/10.1016/j.foreco.2018.06.002

He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: Proc. 2016 IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR 2016), pp 770–778. https://doi.org/10.1109/CVPR.2016.90

Hellström T, Lärkeryd P, Nordfjell T, Ringdahl O (2009) Autonomous forest vehicles: historic, envisioned, and state-of-the-art. Int J For Eng 20(1):31–38. https://doi.org/10.1080/14942119.2009.10702573

Hinton G, Vinyals O, Dean J (2015) Distilling the knowledge in a neural network. arXiv:1503.02531

Ido J, Saitoh T (2019) CNN-based tree species identification from bark image. In: Proc. 10th Int. Conf. Graph. Image Process. (ICGIP 2018). https://doi.org/10.1117/12.2524213

Ido J, Saitoh T (2020) Automatic tree species identification from natural bark image. In: Proc. 11th Int. Conf. Graph. Image Process. (ICGIP 2019). https://doi.org/10.1117/12.2557187

Song J, Chi Z, Liu J, Fu H (2004) Bark classification by combining grayscale and binary texture features. In: Proc. 2004 Int. Symp. Intell. Multimed Video Speech Process., pp 450–453. https://doi.org/10.1109/ISIMP.2004.1434097

Kullback S, Leibler RA (1951) On information and sufficiency. The Annals Math Stat 22(1):79–86. https://doi.org/10.1214/aoms/1177729694

Lakmann R (1998) Barktex benchmark database of color textured images. Koblenz-Landau University

Lecun Y, Bottou L, Bengio Y, Haffner P (1998) Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11):2278–2324. https://doi.org/10.1109/5.726791

Liu H, Dong P, Wu C, Wang P, Fang M (2021) Individual tree identification using a new cluster-based approach with discrete-return airborne LiDAR data. Remote Sens Environ 258:112382. https://doi.org/10.1016/j.rse.2021.112382

Misra D, Crispim-Junior C, Tougne L (2020) Patch-based CNN evaluation for bark classification. In: Workshop 2020 Eur. Conf. Computer Vis. (ECCV 2020). https://doi.org/10.1007/978-3-030-65414-6_15

Müller R, Kornblith S, Hinton GE (2019) When does label smoothing help? In: Proc. 2019 Adv. Neural Inf. Process. Syst. (NeurIPS 2019), pp 4694–4703

Ratajczak R, Bertrand S, Crispim-Junior C, Tougne L (2019) Efficient bark recognition in the wild. In: Proc. 14th Int. Jt. Conf. Computer Vis. Imaging Computer Graph. Theory Appl. (VISAPP 2019). https://doi.org/10.5220/0007361902400248

Ravindran P, Costa A, Soares R, Wiedenhoeft AC (2018) Classification of CITES-listed and other neotropical Meliaceae wood images using convolutional neural networks. Plant Methods 14(1):25. https://doi.org/10.1186/s13007-018-0292-9

Remeš V, Haindl M (2019) Bark recognition using novel rotationally invariant multispectral textural features. Pattern Recognit Lett 125:612–617. https://doi.org/10.1016/j.patrec.2019.06.027

Robert M, Dallaire P, Giguère P (2020) Tree bark re-identification using a deep-learning feature descriptor. In: Proc. 17th Conf. Computer Robot Vis. (CRV 2020), pp 25–32. https://doi.org/10.1109/CRV50864.2020.00012

Sandler M, Howard A, Zhu M, Zhmoginov A, Chen L (2018) MobileNetV2: inverted residuals and linear bottlenecks. In: Proc. 2018 IEEE/CVF Conf. Computer Vis. Pattern Recognit. (CVPR 2018), pp 4510–4520. https://doi.org/10.1109/CVPR.2018.00474

Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: Proc. 3rd Int. Conf. Learn. Represent. (ICLR 2015)

Šulc M, Matas J (2017) Fine-grained recognition of plants from images. Plant Methods 13(1):115. https://doi.org/10.1186/s13007-017-0265-4

Švab M (2014) Computer-vision-based tree trunk recognition. Bsc Thesis (Mentor: Dr. Matej Kristan). Fakulteta za racunalništvo in

Valérie T, Marie-Pierre J (2006) Tree species identification on large-scale aerial photographs in a tropical rain forest, French Guiana-application for management and conservation. For Ecol Manag 225(1):51–61. https://doi.org/10.1016/j.foreco.2005.12.046

Wendel A, Sternig S, Godec M (2011) Automated identification of tree species from images of the bark, leaves and needles. In: Proc. 16th Computer Vis. Winter Workshop, pp 67–74

Wan Y, Du J, Huang D, Chi Z, Cheung Y, Wang X, Zhang G (2004) Bark texture feature extraction based on statistical texture analysis. In: Proc. 2004 Int. Symp. Intell. Multimed Video Speech Process., pp 482–485. https://doi.org/10.1109/ISIMP.2004.1434106

Chi Z, Li H, Wang C (2003) Plant species recognition based on bark patterns using novel Gabor filter banks. In: Proc. 2003 Int. Conf. Neural Netw. Signal Process., pp 1035–1038. https://doi.org/10.1109/ICNNSP.2003.1281045