Efficient Bark Recognition in the Wild

Rémi Ratajczak[1,2,3], Sarah Bertrand[1], Carlos Crispim-Junior[1] and Laure Tougne[1]
[1]*Univ Lyon, Lyon 2, LIRIS, F-69676 Lyon, France*
[2]*Unité Cancer et Environnement, Centre Léon Bérard, Lyon, France*
[3]*Agence de l'Environnement et de la Maîtrise de l'Energie, Angers, France*

Keywords: Bark Recognition, Texture Classification, Color Quantification, Dimensionality Reduction, Data Fusion.

Abstract: In this study, we propose to address the difficult task of bark recognition in the wild using computationally efficient and compact feature vectors. We introduce two novel generic methods to significantly reduce the dimensions of existing texture and color histograms with few losses in accuracy. Specifically, we propose a straightforward yet efficient way to compute Late Statistics from texture histograms and an approach to iteratively quantify the color space based on domain priors. We further combine the reduced histograms in a late fusion manner to benefit from both texture and color cues. Results outperform state-of-the-art methods by a large margin on four public datasets respectively composed of 6 bark classes (*BarkTex*, *NewBarkTex*), 11 bark classes (*AFF*) and 12 bark classes (*Trunk12*). In addition to these experiments, we propose a baseline study on Bark-101, a new challenging dataset including manually segmented images of 101 bark classes that we release publicly.

1 INTRODUCTION

Automatic bark recognition applied to tree species classification is an important problematic that has gained interest in the computer vision community. It is an interesting challenge to evaluate texture classification algorithms on color images acquired in the wild. Due to their low inter-class variability and high intra-class variability, bark images are indeed considered as difficult to classify for a machine as for a human.

In the context of tree species classification, barks have interesting properties compared to other commonly used attributes (*e.g.* leaves, fruits, flowers, *etc.*). Firstly, they are non-seasonal: Whatever the season, the texture of a bark will not change. This property is particularly important to classify a tree during winter, when no leaves or fruits are present. Secondly, bark textures rarely change through short time periods (*i.e.* a year basis), but they do change over long periods (*i.e.* tenth year basis). This property enables the use of age priors for bark classification, but it makes the recognition even more challenging when these priors are unavailable. Finally, barks are easier to isolate and to photograph compared to fruits and leaves that may be unreachable on tall trees. In result, bark images have been used to recognize tree species either alone (Bertrand et al., 2017), or in combination with other tree's attributes (Bertrand et al., 2018). In order to recognize trees, mobile applications like Folia[1] (Cerutti et al., 2013) have been developed. These applications assume that users do not necessarily have an Internet connection, which is a very common situation in the wild. They should work on embedded devices by seeking a trade-off between accuracy, time complexity and space complexity to ensure state-of-the-art recognition rates while avoiding unnecessary energy consumption. Consequently, most of these applications are based on efficient handcrafted filters.

In this context, we propose two novel approaches to drastically reduce the dimensionality of both texture and color feature vectors, while preserving accuracy. We focused our work on handcrafted methods and decided not to use end-to-end Deep Convolutional Neural Networks (DCNNs) because existing datasets are not well suited to both train and evaluate deep learning algorithms: They contain relatively few images, up to 1632 for 6 classes in (Porebski et al., 2014). Moreover, DCNNs have both high time and space complexity that make them unsuitable for embedded usage.

To overcome the few numbers of segmented bark classes in existing datasets; up to 12 in (Švab, 2014);

[1]http://liris.univ-lyon2.fr/reves/folia/public/

Ratajczak, R., Bertrand, S., Crispim-Junior, C. and Tougne, L.
Efficient Bark Recognition in the Wild.
DOI: 10.5220/0007361902400248
In *Proceedings of the 14th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2019)*, pages 240-248
ISBN: 978-989-758-354-4

Figure 1: Examples of bark images from Bark-101 dataset.

we also release a novel and even more challenging dataset made of 101 segmented bark classes in high resolution: Bark-101[2] (see Section 2.1). This dataset has been conceived with a clear focus on the number of classes, involving high intra-class and low inter-class variabilities (see Figure 1).

The remainder of this study is organized as follow. Section 2 presents previous work related to bark recognition in the wild. Section 3 introduces and details the proposed algorithms for efficient dimensionality reduction of texture and color histograms. Section 4 presents the experiments we carried out and discusses the results.

2 RELATED WORK

This section presents existing bark datasets and state-of-the-art methods for bark recognition.

2.1 Datasets

We propose to describe four state-of-the-art datasets that are publicly available and commonly used in the literature. In complement, we release an even more challenging dataset: Bark-101. The characteristics of these datasets are detailed below and summarized on Table 1.

BarkTex and NewBarkTex. BarkTex was the first dataset specialized on bark recognition. It was introduced by R. Lakmann (Lakmann, 1998). It is composed of 6 classes, each corresponding to a different tree species. Each class contains 68 color images of trunks for a total of 408 images. The trunks are spatially centered in the images, but some background can appear depending on the width of the trunk. NewBarkTex is derived from BarkTex. It was proposed in (Porebski et al., 2014) and it is also composed of 6 classes. A region of interest (ROI) of size 128x128 pixels was cropped from the center of the images of BarkTex. Then the ROI was separated in 4 sub-images of 64x64 pixels. Half of the sub-images were kept for training and the second half for testing. Therefore, NewBarkTex is made of 272 images per class for a total of 1632 images.

Trunk12. It is a bark dataset created in 2014 by Matic Švab (Švab, 2014). It consists of 360 color images of barks corresponding to 12 different species of tree found in Slovenia. Each class consists of about 30 images. All images were taken with the same camera in the same conditions (20 cm distance, multiple trees per class, avoiding moss, same light conditions, taken in upright position).

AFF. This bark dataset was presented in (Wendel et al., 2011). It has 11 classes of bark from the most common Austrian trees. It is composed of 1082 color images of bark. In AFF, tree species are separated in sub-classes according to the age of the tree. The texture of the trunk changes during the lifetime of a tree, starting from a smooth to a more and more coarse texture. In this study, we have chosen to not separate the classes according to the age of the trees.

Bark-101. We built the Bark-101 dataset among the PlantCLEF[3] identification task, part of the ImageCLEF challenge, designed to compare plant recognition algorithms. PlantCLEF contains photos of multiple yet not segmented plant organs (leaf, flower, branch, steam, *etc.*) taken by people in various unsupervised shooting conditions and gathered through the mobile application Pl@ntNet[4]. More than 500 herb, tree and fern species centered on France are present in PlantCLEF (Goëau et al., 2014). To construct Bark-101, we kept the tree stems available from PlantCLEF (*i.e.* the barks), and have manually segmented them to remove undesirable background information. We decided to follow the authors of (Wendel et al., 2011) by not constraining the size of the segmented images in Bark-101. This choice was made to simulate real world conditions assuming a perfect stem segmentation. As a matter of fact, in a real world setting not all trees would have the same diameter and not all users would photograph them at a pre-defined distance. The Bark-101 dataset is therefore composed of 101 classes of tree barks from various age and size for a total of 2592 images. Images in Bark-101 contain noisy data like shadows, mosses or illumi-

[2]http://eidolon.univ-lyon2.fr/~remi1/Bark-101/

[3]https://www.imageclef.org/lifeclef/2017/plant

[4]https://identify.plantnet-project.org/

Table 1: Characteristics of bark datasets.

Dataset information	BarkTex	NewBarkTex	Trunk12	AFF	Bark-101
Classes	6	6	12	11	101
Total images	408	1632	393	1082	2592
Images per classes	68	272	30-45	16-213	2-138
Image size	256x384	64x64	1000x1334	1000x(478-1812)	(69-800)x(112-804)
Illumination change	✓	✓	✗	✓	✓
Scale change	✓	✓	✗	✓	✓
Noise (shadows, lichen)	✗	✗	✗	✓	✓
Train / Test splits	✗	50/50	✗	✗	50/50

nation changes (see Figure 1). Due to the unsupervised acquisition of the images and the variation of bark textures over the lifespan of the tree, there is a high intra-class variability in Bark-101. Furthermore, a large number of classes naturally leads to a small inter-class variability since the number of visually similar species increases with the number of species. Consequently, Bark-101 can be considered as a challenging dataset in the context of bark recognition. We further demonstrate this statement in the experiments presented in Section 4.

2.2 Existing Methods

Bark recognition is often considered as a texture classification problem. In (Wan et al., 2004) the authors compared different statistical analysis tools for tree bark recognition, like co-occurrence matrices, grayscale histogram analysis and run-length method (RLM). The use of co-occurrence matrices on grayscale bark images is also present in (Huang et al., 2006b). The authors combined them with fractal dimensions to characterize the self-similarity of bark textures at different scales. Spectral methods, such as Gabor filters, are also used. In (Huang et al., 2006a), the authors demonstrated that only four wave orientations and 6 scales are sufficient to identify tree species by their bark. To avoid losing the information provided by color, Wan *et al.* (Wan et al., 2004) applied their grayscale method individually to each channel of the *RGB* space. In (Bakić et al., 2013), different color spaces were used to characterize color information, including *RGB* and *HSV* spaces. The authors of (Bertrand et al., 2017) proposed to combine texture and color hues in a late fusion manner. First, they extracted orientation features using Gabor filters. Secondly, they combined these features with a sparse representation of bark contours by encoding the intersections of Canny edges with a regular grid. Lastly, they described bark colors using the hue histogram from the *HSV* color space. The resulting descriptor proved to increase the classification rate of tree recognition when combined with leaves (Bertrand et al., 2018).

Other commonly used descriptors for bark classification are Local Binary Pattern-like (LBP-like) descriptors that were inspired by the original LBP filter proposed by (Ojala et al., 2001). LBP-like filters are generic local texture descriptors parametrized over a (P,R) neighborhood, with P the number of pixel neighbors and R the radii. In the case of a multiscale filter, the number R_s of radii R is strictly greater than 1. LBP-like filters encode textural patterns with binary codes, whose aggregation result in a texture histogram of high dimension (*e.g.* $R_s \times 2^P$ for the original LBP). Recently, the authors of (Boudra et al., 2018) proposed a texture descriptor called Statistical Macro Binary Pattern (SMBP) inspired by LBP. SMBP encodes the information between macrostructural scales with a representation using statistical characteristics for each scale. The descriptor increases the classification rate on two barks datasets compared to the state of the art. In (Alice Porebski, 2018), the authors demonstrated the accuracy of color LBP-like descriptors with different color spaces, achieving above state-of-the-art performance, but with a very high dimensional feature vector. Recent works on texture classification in the wild are also of interest for bark recognition. In particular, the Light Combination of Local Binary Patterns (*LCoLBP*), proposed by (Ratajczak et al., 2019), and the Completed Local Binary Pattern (*CLBP*), proposed by (Guo, Z. et al., 2010), obtained equivalent results to popular DCNNs on historical aerial images classification for a small computational cost. Since the CLBP filter is often used as a baseline for bark recognition based on LBP-like filters (Junmin Wang, 2017) the LCoLBP may be a suitable candidate for bark recognition in the wild.

3 PROPOSED METHODS

Bark images acquired in the wild represent objects with discriminative texture patterns and colors. To represent these characteristics considering a trade-off between accuracy and feature space complexity, we propose two novel complementary methods to efficiently reduce the number of texture and color features. We considered these cues individually before combining them in a late fusion manner (concatenation). For the texture cues, we followed the com-

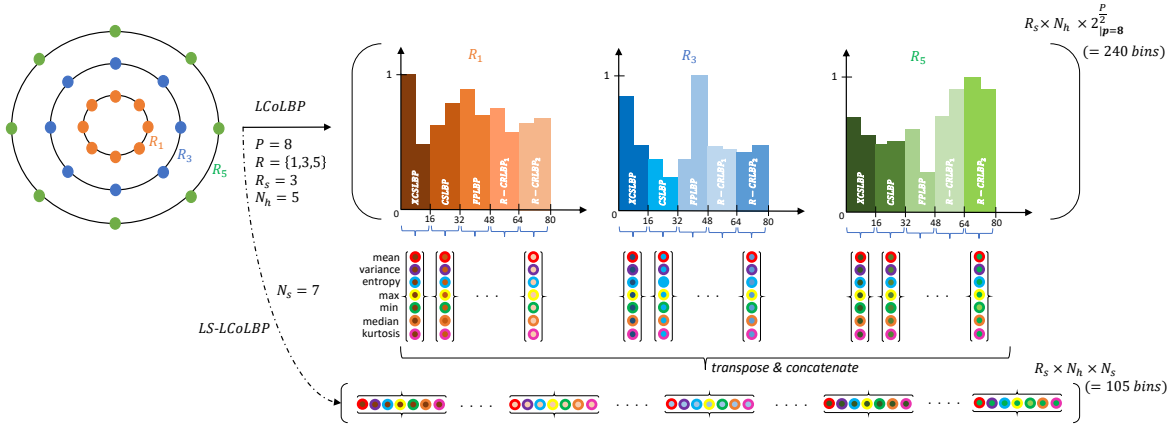

Figure 2: Late Statistics (*LS*) from a Light Combination of Local Binary Patterns (*LCoLBP*) with a 3-radii neighborhood ($R_s = 3$), 8 neighbors P, and 7 statistics N_s. Colored dots denote the statistics (outer colors) obtained from the $N_h = 5$ sub-histograms included in the LCoLBP histogram for each radius (inner colors). This configuration yields a 2.3 reduction factor.

mon use of LBP-like filters. Based on these filters, we developed a generic yet efficient statistical representation which preserves filters' properties and does not require re-sampling or mapping operations (see Section 3.1). For the color cues, we were inspired by (Bertrand et al., 2017) and built a task-guided low dimensional histogram representation upon the *HSV* color space using bark priors (see Section 3.2).

3.1 From Textures to Late Statistics

We define the Late Statistics as a combination of statistical features calculated from LBP-like texture histograms. Considering a texture histogram H_t, itself made of the concatenation of N_h known and ordered sub-histograms $\{h_1, h_2, ..., h_{N_h}\}$, we calculate N_s Late Statistics independently for each h_i with $i \in \{1, ..., N_h\}$. Late Statistics are then concatenated in the same order as the h_i sub-histograms. Assuming a single H_t per LBP-like radius, this process results in a vector made of $R_s \times N_s \times N_h$ features, where R_s is the number of radii (*i.e.* scales). It is represented with $N_s = 7$ statistics and a grayscale 3-radii ($R_s = 3$) LCoLBP filter on Figure 2. Note that each rotation of the R-CRLBP (Ratajczak et al., 2018), a sub-filter of the LCoLBP, is considered as an independent filter in this study.

Late statistics are defined as *late* in opposition to the *early* statistics proposed by (Boudra et al., 2018). In (Boudra et al., 2018), the authors calculated statistics *before* calculating the texture histogram by re-sampling the local textural patterns of a LBP-like filter. In consequence, the statistical approach proposed by Boudra *et al.* would require new implementations with error-prone re-sampling operations to be applied to other LBP-like filters. The Late Statistics evade this constraint by considering texture histograms that have been *already calculated*. They do not need to operate on the local textural patterns but rather on the global histogram representations, so that they do not require any modification of the descriptor implementations (*i.e.* no re-sampling).

Due to their nature, Late Statistics are expected to preserve common descriptor properties, like rotation and global illumination invariance. This point should make the Late Statistics at least as robust to condition changes as the descriptors themselves. Furthermore, it should be observed that, similarly to the *early* statistics used by (Boudra et al., 2018), the Late Statistics naturally behave as a spatial normalization algorithm: A histogram will be summarized with a fixed number of statistics N_s whatever its number of bins. This property is particularly useful to combine different histograms in a balanced feature vector that contains the same quantity of information for each texture pattern.

Finally, one may observe that Late Statistics are an efficient way to summarize textural information in a very similar manner as Haralick features (Haralick et al., 1973). However, while Haralick features represent statistics calculated directly from an image, Late Statistics benefit from the efficient LBP-like representation of textures.

3.2 Efficient Colors

We discussed in Section 2 that several bark descriptors work on grayscale images and that a few of them also include color data. Among these descriptors, hue information is commonly used (Bakić et al., 2013; Bertrand et al., 2017). The size of a hue histogram H

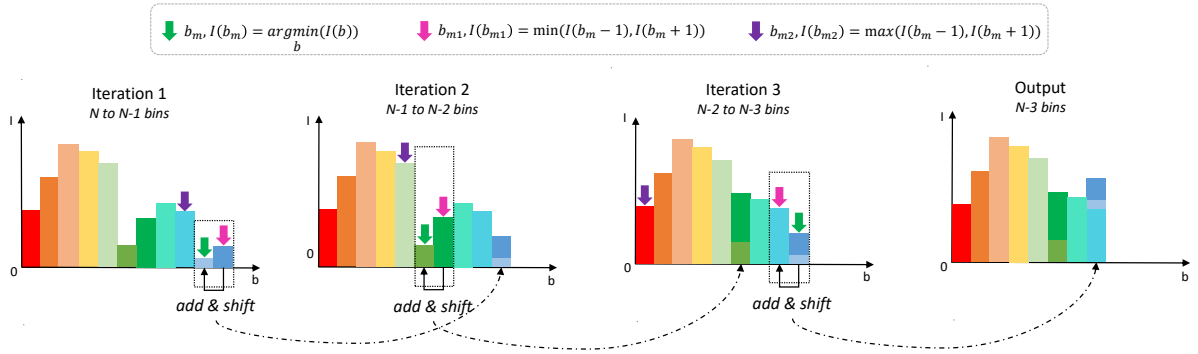

Figure 3: Schematic of the color histogram reduction applied with 3 iterations. The number of bins in the histogram is reduced by 1 after each iteration. The reduction is performed using a local add and shift strategy on the bin b with the smallest intensity $I(b)$. A look-up table is used to map the modified bins.

is usually of 360 bins. However, this value (360) cannot be stored on a single byte. For sampling reasons, it is preferable to transpose the hue color range *[0;359]* into the range *[0;179]* than into the range *[0;255]*. Therefore, the size of the hue histogram is 180 bins, which represents a feature vector of a quite high dimensionality for task-specific applications with color priors, like bark recognition in the wild. As visible on Figure 1, barks have dominant variations of brown, green and yellow colors. Based on this observation, we may expect that other colors like blue or purple may not provide significant contribution to the hue signature of a bark. However, completely removing these colors may result in less discriminant histograms, making the bark classification process even more difficult.

To reduce the size of the hue histogram, we propose to merge the least represented colors through a non-destructive iterative process. For a given dataset D made of k color images splitted into a train set T_r and a test set T_e, we first calculate and sum all the hue histograms of the images in T_r. We obtain a *summed* hue histogram H_s from the train set $T_r \subset D$. This *summed* histogram H_s is supposed to represent the color prior on the whole dataset D. Secondly, on the *summed* histogram H_s, we iteratively add the population of the bin b having the smallest intensity value (*i.e.* the smallest population) to the population of its neighbor of minimal population. After adding these bins, we shift the histogram to the left in order to reduce its dimension. The iteration process is stopped when the desired size, fixed by the user, is reached (see Section 4). Attention should be brought to the circularity of the hue channel in the *HSV* color space. The order and position of the add and shift operations are stored in a look-up table M. On the test set T_e, the look-up table M is then used to calculate the reduced histograms of the input images L_{T_e} by applying the add and shift operations in the same order as in T_r. Consequently, the reduced histogram for an image $l_{T_e} \in L_{T_e}$ is generated in regard to the *summed* histogram, thus taking into account the color prior of the data. The add and shift process is illustrated on Figure 3.

4 EXPERIMENTS AND RESULTS

4.1 Experimental Setup

For our experiments, we considered two state-of-the-art LBP-like filters made of complementary sub-descriptors to assess the efficiency of the Late Statistics: The Light Combination of Local Binary Patterns (*LCoLBP*) and the Completed Local Binary Pattern (*CLBP*). As explained in Section 2, these descriptors are efficient texture filters which obtained DCNNs like accuracy rates on texture in the wild datasets (Ratajczak et al., 2019). Since these filters may result in very high dimensional histograms, we considered a constant number of neighbors P on a 3-radii ($R_s = 3$) neighborhood: ($P = 8, R = \{1,3,5\}$). To assess the effectiveness of the Late Statistics on both mapped and unmapped LBP-like representations, we followed (Ratajczak et al., 2019) and applied the LCoLBP without any mapping, resulting in a histogram of $R_s \times 5 \times 2^4$ bins ($N_h = 5$) with a ($P = 8, R = \{1,3,5\}$) neighborhood. On the other hand, we applied the rotational and uniform (riu^2) mapping introduced by (Ojala et al., 2001) with the CLBP filter, resulting in a histogram of $3 \times (2 + 2 \times 10)$ bins instead of $3 \times (2 + 2 \times 256)$ bins without mapping for the same neighborhood.

We considered commonly used statistics in our experiments: Mean, variance, entropy, minimum, maximum, median and kurtosis. Among them, we carried out an ablation study through gridsearch to determine the best combination of Late Statistics for each tex-

Table 2: Ablation study for the Late Statistics of the LCoLBP and CLBP filters on the BarkTex dataset.

Late Statistics							Accuracy (%)	
mean	variance	entropy	minimum	maximum	median	kurtosis	*LS-LCOLBP*	*LS-CLBP*
✓	–	–	–	–	–	–	81.9	71.8
✓	✓	–	–	–	–	–	82.8	59.6
✓	✓	✓	–	–	–	–	78.4	64.7
✓	✓	✓	✓	–	–	–	82.8	63.2
✓	✓	✓	✓	✓	–	–	83.1	69.4
✓	✓	✓	✓	✓	✓	–	86.3	72.1
✓	✓	✓	✓	✓	✓	✓	89.5	62.8
✓	✓	–	✓	✓	✓	✓	88.2	60.1
✓	–	✓	✓	✓	✓	✓	**89.5**	62.5
✓	–	–	✓	✓	✓	✓	88.2	59.6
✓	–	–	✓	✓	✓	–	88.2	**75.3**

ture filter as presented on Table 2 for BarkTex. Ablation results obtained on other datasets were in concordance with Table 2 as well as with the following observations. On Table 2, we can observe that naively adding more Late Statistics (first seven lines) may decrease the accuracy of the texture representations. We also demonstrate that the Late Statistics should be carefully and individually selected for each texture filter in order to maximize the accuracy rate (up to a difference of $\tilde{1}5\%$ on Table 2). This phenomenon highlight a minor drawback of the Late Statistics: While they are effective and easy to implement, they increase the number of parameters to tune. Therefore, the number of statistics N_s was set to 6 for the LCoLBP and 4 to CLBP. We did not apply the Late Statistics on the first sub-histogram of the CLBP filter because it is made of only two bins. Late Statistics of the LCoLBP filter ($LS-LCoLBP$) result in $R_s \times 5 \times N_s$ features. Late Statistics of the CLBP filter ($LS-CLBP$) result in $R_s \times (2+2 \times N_s)$ features.

Concerning the bark colors, we verified that bark images are actually made of dominant colors on the extensive train set of the Bark-101 dataset because of the large variability of bark images and classes available (see Section 2.1). The *summed* hue histogram H_s for this dataset is visible on Figure 4. Based on this figure, we can observe that the majority of barks present brownish, yellowish and greenish hues. In order to define a suitable dimension for the reduced color histogram, we observed the evolution of the classification rate according to the strategy defined in

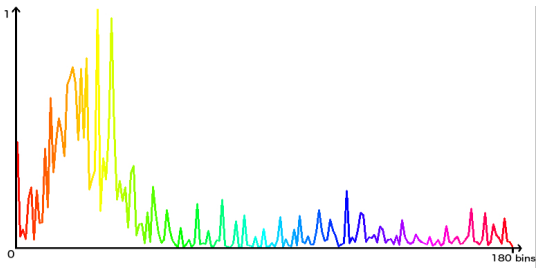

Figure 4: Summed histogram of the hue channels from the train set of the Bark-101 dataset.

Section 4.2 on Bark-101 dataset (Figure 5). Despite small variations in the species classification rate when quantifying the histogram, we can see that the accuracy remains approximately constant over 30 bins. As the other bark datasets present smaller variance and are thus less representative of real world conditions, we decided to set the size of the reduced hue histogram to 30 bins regardless of the bark dataset. We remind that a look-up table, to obtain the reduced histograms, is calculated for each dataset independently: Only the size of 30 bins has been fixed using Bark-101.

4.2 Evaluation and Metrics

We separated our evaluation into two classification strategies, c_1 and c_2, depending on the dataset organizations, visible on Table 1, and previous state-of-the-art experiments. Strategy c_1 stands for the classical train/test strategy. In c_1, a part of the dataset is used for training and the rest of the dataset is used for testing. We performed c_1 on NewBarkTex and Bark-101 following the train and test splits (50%/50%) provided by the authors. Strategy c_2 is the leave-one-out strategy (LOO). In c_2, we considered an ensemble S of N samples and we performed N iterations. At each iteration $i \in \{1,...,N\}$, we used a different sample $s(i)$ of S for evaluation (*i.e.* testing) and all the other samples of $S-\{s(i)\}$ for training. If the evaluation sample $s(i)$ was successfully classified, the result of the corresponding iteration was set to 1, and to 0 otherwise. The final accuracy was obtained by averaging the results of all iterations. In accordance with (Boudra et al., 2018), we performed c_2 on BarkTex, Trunk12 and AFF. For both c_1 and c_2 we used the top-1 accuracy and a K-Nearest Neighbor classifier (KNN) with $K=1$ and the $L1$ distance. The 1-NN is the most commonly used classifier in the context of bark and texture recognition. The $L1$ distance has been chosen arbitrarily. Additionally, for c_1, and in accordance with (Alice Porebski, 2018), we used a multi-class non-linear Support Vector Machine (SVM) with radial ba-

sis kernel. The parameters of the SVM classifier have been optimized by grid search for each dataset and for each feature vector.

4.3 Results and Discussion

Results are visible on Table 3 and Table 4. They have been obtained considering the evaluation strategies described in Section 4.2. Highest accuracy rates from other studies have been reported and marked with a right-top star symbol. Specifically, for AFF, Trunk12 and BarkTex, we reported the results obtained with *MSLBP** and *SLBP** from (Boudra et al., 2018). The results for NewBarkTex were reported for *Wang17** (Junmin Wang, 2017), *Sandid16** (Sandid and Douik, 2016), and *Porebski18** (Alice Porebski, 2018). We also considered the results proposed by the methods of (Bertrand et al., 2017) that we renamed *GWs* and *GWs/H180*. All other results correspond to our own implementations using OpenCV 3.4 in C++ for the texture and color descriptors. Scikit-learn in Python has been used for the Late Statistics and the classifiers. The texture cues were calculated on grayscale images. The color cues were calculated on bark images in *HSV* color space. The following sections discuss the results obtained.

4.3.1 Color Cues

On Tables 3 and 4, we can observe that all the texture filters obtained higher accuracy rates when combined with color histograms of both 30 bins (H30) and 180 bins (H180). As a reminder, H30 is the reduced hue histogram and H180 is the complete hue histogram. When used alone, H180 is, in average, only 3.3% more accurate than H30 on AFF, Trunk12 and BarkTex. However, we found that when combined with grayscale textures, the contribution of both color representations is equivalent. These results demonstrate the efficiency of the color histogram reduction algorithm presented in Section 3.2. Moreover, these results confirm that color cues seem to be non-negligible features for bark recognition, in opposition to the assumption made by (Boudra et al., 2018) but in accordance with (Junmin Wang, 2017).

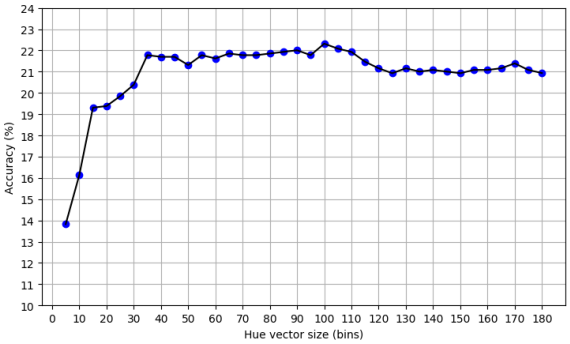

Figure 5: Accuracy of the reduced hue histogram on the Bark-101 dataset depending on the final number of bins.

Table 3: Results of Late Statistics and reduced color histograms with KNN and leave-one-out strategy. Blue: Highest results of the literature. Green: highest results overall. Red: Highest late statistics results.

Descriptor	size	Top-1 Accuracy / Dataset (%)		
		AFF	Trunk12	BarkTex
*MSLBP**	2 816	63.3	63.3	**86.8**
*SMBP**	10 240	**71.7**	**71.0**	84.3
H30	30	50.5	64.4	55.4
H180	180	55.6	69.0	61.3
LCoLBP	240	75.3	77.1	92.1
LCoLBP / H30	270	**80.7**	**84.2**	**92.4**
LCoLBP / H180	420	80.7	84.2	91.7
CLBP	66	68.1	70.0	78.7
CLBP / H30	96	72.9	77.4	83.8
CLBP / H180	246	73.5	78.1	84.3
GWs	121	48.2	39.9	56.1
GWs / H30	151	64.7	74.3	66.2
GWs / H180	301	66.5	76.1	69.6
LS-LCoLBP	90	69.4	74.6	89.5
LS-LCoLBP / H30	120	**76.9**	**80.7**	90.2
LS-LCoLBP / H180	270	76.9	80.7	**91.2**
LS-CLBP	30	59.1	70.0	75.3
LS-CLBP / H30	60	65.4	77.4	78.2
LS-CLBP / H180	210	67.9	78.1	79.4

Table 4: Results achieved on NewBarkTex and Bark-101.

Descriptor	size	Top-1 Accuracy / Dataset(%)			
		NewBarkTex		Bark-101	
		KNN	SVM	KNN	SVM
*Porebski18**	10 752	–	**92.6**	–	–
*Wang17**	267	84.3	–	–	–
*Sandid16**	3 072	–	82.1	–	–
H30	30	48.0	50.6	19.1	20.4
H180	180	48.5	53.6	22.2	20.9
LCoLBP	240	78.8	**89.3**	34.2	**41.9**
LS-LCoLBP	90	66.5	79.4	28.3	30.1
LS-LCoLBP / H30	120	71.9	82.0	27.6	**32.1**
LS-LCoLBP / H180	270	72.3	**82.2**	27.8	31.0
GWs / H30	151	60.4	74.1	28.2	**31.7**
GWs / H180	301	54.1	63.6	31.8	**32.2**

4.3.2 Late Statistics

The Late Statistics decreased the size of the evaluated texture features with a multiplicative factor between 2.7 for the LCoLBP and 2.2 for the CLBP with an averaged reduction in accuracy of only 5.5% overall. The Late Statistics seem particularly efficient with a leave-one-out strategy (Table 3) with an averaged difference of 3.7% between the LCoLBP and the LS-LCoLBP, and an averaged difference of 4.1% between the CLBP and the LS-CLBP. On the other hand, they are slightly less effective on the Train/Test strategies. These results may be explained by a lack of training data resulting in an under accurate statistical description of the per-class texture histograms.

4.3.3 Overall Performances

We can observe that the Late Statistics combined with the reduced hue histograms H30 outperform prior works on the AFF, Trunk12 and BarkTex datasets. LS-LCoLBP/H30 is in averaged 6.3% more accurate than the methods compared in (Boudra et al., 2018). Moreover, it is about 100 times smaller than *SMBP**. On NewBarkTex, Late Statistics combined with hue histograms and a SVM classifier achieve similar results to *Sandid16** with an accuracy of 82.0%. We observe that our method ($LS-LCoLBP/H30$) obtained slightly lower results than the most accurate algorithms from the literature on this dataset, but it does have a significantly smaller feature vector which is about 30 times smaller than *Sandid16** and 100 times smaller than *Porebski18**. On Bark-101, we can observe the lowest accuracy rates for all compared methods over all the datasets. These results are explained by the higher number of classes in Bark-101 compared to existing datasets. It also demonstrates the challenge proposed by Bark-101. However, most methods including GWs/H30 achieved a top-1 recognition rate about 30%, which is far above the random guess of 0.9%.

5 CONCLUSION

In this study, we compared recent state-of-the-art descriptors in the context of tree bark recognition in the wild. We proposed two novel algorithms to reduce the dimensionality of texture and color features vectors. We showed that the proposed algorithms outperform state-of-the-art methods on four bark datasets with a considerable gain in space complexity. We believe that these methods can be generalized on other histogram-like feature vectors. Furthermore, we released a new dataset made of 101 bark classes of segmented images with high intra-class variability. We demonstrated that the proposed dataset is particularly challenging for existing methods, enforcing the need for future prospects on bark recognition. Future work will investigate the proposed methods as a lightweight representation with multiple color spaces. We will also evaluate the proposed algorithms on mobile platforms, such as smartphones, to assess their performances on real-world settings.

ACKNOWLEDGEMENT

This work is part of ReVeRIES project (Reconnaissance de Végétaux Récréative, Intéractive et Educative sur Smartphone) supported by the French National Agency for Research with the reference ANR-15-CE38-004-01, and part of the French Environment and Energy Management Agency, Grant TEZ17-42.

REFERENCES

Alice Porebski, Vinh Truong Hoang, N. V. D. H. (2018). Multi-color space local binary pattern-based feature selection for texture classification. *Journal of Electronic Imaging*, 27:27 – 27 – 15.

Bakić, V., Mouine, S., Ouertani-Litayem, S., Verroust-Blondet, A., Yahiaoui, I., Goëau, H., and Joly, A. (2013). Inria's participation at imageclef 2013 plant identification task. In *CLEF Working Notes*.

Bertrand, S., Ameur, R. B., Cerutti, G., Coquin, D., Valet, L., and Tougne, L. (2018). Bark and leaf fusion systems to improve automatic tree species recognition. *Ecological Informatics*.

Bertrand, S., Cerutti, G., and Tougne, L. (2017). Bark recognition to improve leaf-based classification in didactic tree species identification. In *VISAPP*.

Boudra, S., Yahiaoui, I., and Behloul, A. (2018). Plant identification from bark: A texture description based on statistical macro binary pattern. In *ICPR*.

Cerutti, G., Tougne, L., Mille, J., Vacavant, A., and Coquin, D. (2013). Understanding leaves in natural images - a model-based approach for tree species identification. *Computer Vision and Image Understanding*.

Goëau, H., Joly, A., Bonnet, P., Selmi, S., Molino, J.-F., Barthélémy, D., and Boujemaa, N. (2014). Lifeclef plant identification task 2014. In *CLEF2014 Working Notes. Working Notes for CLEF 2014 Conference, Sheffield, UK, September 15-18, 2014*, pages 598–615. CEUR-WS.

Guo, Z., Zhang, L., and Zhang, D. (2010). A Completed Modeling of Local Binary Pattern Operator for Texture Classification. *IEEE TIP*, 19(6).

Haralick, R. M., Shanmugam, K., and Dinstein, I. (1973). Textural Features for Image Classification. *IEEE Transactions on Systems, Man, and Cybernetics*.

Huang, Z.-K., Huang, D.-S., Du, J.-X., Quan, Z.-H., and Guo, S.-B. (2006a). Bark classification based on gabor filter features using rbpnn neural network. In *International conference on neural information processing*, pages 80–87. Springer.

Huang, Z.-K., Zheng, C.-H., Du, J.-X., and Wan, Y.-y. (2006b). Bark classification based on textural features using artificial neural networks. In *International Symposium on Neural Networks*. Springer.

Junmin Wang, Yangyu Fan, N. L. (2017). Combining fine texture and coarse color features for color texture classification. *Journal of Electronic Imaging*.

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

Ojala, T., Pietikäinen, M., and Mäenpää, T. (2001). A Generalized Local Binary Pattern Operator for Multiresolution Gray Scale and Rotation Invariant Texture Classi-

fication. In *Advances in Pattern Recognition, ICAPR 2001*, pages 399–408. Springer Berlin Heidelberg.

Porebski, A., Vandenbroucke, N., Macaire, L., and Hamad, D. (2014). A new benchmark image test suite for evaluating colour texture classification schemes. *Multimedia Tools and Applications*, 70(1):543–556.

Ratajczak, R., Crispim-Junior, C., Faure, É., Fervers, B., and Tougne, L. (2019). Automatic land cover reconstruction from historical aerial images: An evaluation of features extraction and classification algorithms. *IEEE TIP (accepted with minor revision)*.

Ratajczak, R., Crispim-Junior, C. F., Faure, É., Fervers, B., and Tougne, L. (2018). Reconstruction automatique de l'occupation du sol à partir d'images aériennes historiques monochromes : une étude comparative. In *Conférence Française de Photogrammétrie et de Télédétection (CFPT)*, Marne-la-Vallée, France.

Sandid, F. and Douik, A. (2016). Robust color texture descriptor for material recognition. *PRL*.

Švab, M. (2014). *Computer-vision-based tree trunk recognition*. PhD thesis, Fakulteta za raunalnitvo in informatiko, Univerza v Ljubljani.

Wan, Y.-Y., Du, J.-X., Huang, D.-S., Chi, Z., Cheung, Y.-M., Wang, X.-F., and Zhang, G.-J. (2004). Bark texture feature extraction based on statistical texture analysis. In *International Symposium on Intelligent Multimedia, Video and Speech Processing*. IEEE.

Wendel, A., Sternig, S., and Godec, M. (2011). Automated identification of tree species from images of the bark, leaves and needles. In *16th Computer Vision Winter Workshop*.