Forest species recognition based on ensembles of classifiers

Jefferson G. Martins  
Department of Informatics  
Federal University of Technology - Paraná (UTFPR)  
Toledo (PR), Brazil  
martins@ufpr.edu.br

Luiz E. S. Oliveira  
Department of Informatics  
Federal University of Paraná (UFPR)  
Curitiba (PR), Brazil  
lesoliveira@inf.ufpr.br

Robert Sabourin  
Département de génie de la production automatisée  
École de Technologie Supérieure (ÉTS)  
Montreal (QC), Canada  
sabourin@etsmtl.ca

Alceu S. Britto Jr  
Department of Computer Science  
Pontifical Catholic University of Paraná (PUCPR)  
Curitiba (PR), Brazil  
alceu@ppgia.pucpr.br

Abstract—Recognition of forest species is a very challenging task thanks to the great intra-class variability. To cope with such a variability, we propose a multiple classifier system based on a two-level classification strategy and microscopic images. By using a divide-and-conquer approach, an image is first divided into several sub-images which are classified independently by each classifier. In a first fusion level, partial decisions for the sub-images are combined to generate a new partial decision for the original image. Then, the second fusion level combines all these new partial decisions to produce the final classification of the original image. To generate the pool of diverse classifiers, we used classical texture-based features as well as keypoint-based features. A series of experiments shows that the proposed strategy achieves compelling results. Compared to the best single classifier, a Support Vector Machine (SVM) trained with a keypoint-based feature set, the divide-and-conquer strategy improves the recognition rate in about 4 and 6 percentage points in the first and second fusion levels, respectively. The best recognition rate achieved by this proposed method is 98.47%.

Index Terms—textural descriptor, multiple classifier system, fusion rules

I. INTRODUCTION

Identifying forest species only from wood lumbers is a task that generally requires an expert, so that an automatic system consists of an alternative to reduce the costs for hiring and training human experts and maybe a way to improve the speed and accuracy of this task. Recently automatic forest species recognition has been drawing the attention of the machine learning community, given its both commercial and environment-preserving value [1]–[5].

A common problem in most of works is that in their experimental protocol, they consider databases containing only a few classes. Just recently, more representative datasets have been built and made available. An interesting alternative that has successfully been used to face the large number of classes is the Multiple Classifier System (MCS). In light of this, researchers have investigated strategies to create pools of classifiers [6]–[8] and combine them [9].

In this paper, we address the problem of forest species recognition using microscopic images. The proposed strategy takes into account multiple classifiers. The contribution of this work is twofold. First, we assess different families of textural descriptors for the problem of forest species classification and investigate the use of keypoint detectors. Secondly, we combine these different classifiers by using a set of fusion rules in a two-level strategy [9]–[12].

Through a set of comprehensive experiments, we show that the keypoint descriptors are a good alternative. The best result of a single descriptor was achieved by the classifier trained with the Speeded-Up Robust Features (SURF) [13], which achieved 92.46% of recognition rate. For the MCS experiments, they showed that combining the classifiers trained with whole images improved the results to 96.63% and the proposed two-levels strategy improved it to 98.47%, which was the best result in this study.

This paper is structured as follows: Section II briefly describes the used database. Section III presents a brief explanation of the descriptors and feature vectors. Section V presents the proposed method. Section VI reports our experiments. Finally, Section VII concludes the work.

II. DATABASE

The database used in this work contains 112 different forest species which were catalogued by the Laboratory of Wood Anatomy at the Federal University of Parana in Curitiba, Brazil. The protocol adopted to acquire the images comprises five steps. In the first step, the wood is boiled to make it softer. Then, the wood sample is cut with a sliding microtome to a thickness of about 25 microns (1 micron = $1 \times 10^{-6}$ meters). In the third step, the veneer is colored using the triple staining technique, which uses acridine red, chrysoidine, and astra blue.
In the fourth step, the sample is dehydrated in an ascending alcohol series. Finally, the images are acquired from the sheets of wood using an Olympus Cx40 microscope equipped with a 100× zoom. The resulting images are then saved in PNG (Portable Network Graphics) format with no compression and a resolution of 1024×768 pixels. More details about the database can be found in [14].

Each species has 20 images, for a total of 2240 microscopic images. Of the 112 available species, 37 are Softwoods and 75 are Hardwoods (Fig. 1). Looking at these samples, we can see that they have different structures. Softwoods have a more homogeneous texture and/or present smaller holes, known as resiniferous channels (Fig. 1a), whereas Hardwoods usually present some large holes, known as vessels (Fig. 1b).

It is worth noting that color cannot be used in this database, since its hue depends on the dyeing substance used to produce contrast in the microscopic images. All the images were therefore converted to gray scale (256 levels) in our experiments.

### III. Features

In this section we describe all the descriptors used to create the classifiers. These include the most commonly textural descriptors found in the literature, such as Local Binary Patterns (LBP) [15], Grey-level Co-Occurrence Matrix [16], Gabor Filters [17], and two keypoint descriptors, namely, Scale-Invariant Feature Transform (SIFT) [18] and Speed-Up Robust Feature (SURF) [13]. Keypoint descriptors, which are usually used for object recognition, are used mainly because of the nature of the microscopic images. As mentioned earlier, we believe that keypoints extracted from resiniferous channels and vessels (Fig. 1a and 1b) may be good descriptors for discriminating textures. In this study, we adopt a sparse feature extraction, where features are computed only at the keypoint pixels generated by the algorithms.

#### A. Scale Invariant Feature Transform (SIFT)

Keypoint descriptors are created by first computing the gradient magnitude and orientation at each image sample point in a 16×16 pixels region around the keypoint location. These keypoints are weighted by a Gaussian window, and then accumulated into 8-orientation histograms summarizing the contents over 4×4 subregions. This results in a vector with 128 dimensions (4×4×8) that is normalized to unit length. After computing a 128-dimensional feature vector for each identified feature point, the statistical moments average, variance, skewness and kurtosis were extracted, generating a 128-dimensional vector for each one. Additionally, we used the number of detected points as a feature and tested different arrangements of it and the four statistical moment vectors. The best results were achieved by using the number of detected points and the 128-dimensional vector for the variance moment.

#### B. Speed-Up Robust Feature (SURF)

SURF detects blob-like structures at locations where the determinant is at a maximum. To that end, the region is regularly split up into smaller 4×4 square subregions, which preserves important spatial information. For each subregion, Haar wavelet responses are computed in horizontal (dx and |dx|) and in vertical (dy and |dy|) directions, forming a descriptor vector of length 64. A SURF variant called SURF-128 uses the same sums as stated earlier, but the sums of dx and |dx| are computed separately for dy < 0 and dy ≥ 0. Similarly, the sums of dy and |dy| take the sign of dx into account, thereby doubling the number of features. This version is said to be more distinctive, and not much slower to compute, but slower to match due to its higher dimensionality.

After computing both possibilities to SURF, 64- and 128-dimensional feature vectors for each identified feature point, we extract the statistical moments average, variance, skewness, and kurtosis, generating a 64- or 128-dimensional vector for each statistical moment. As for SIFT, we used the number of detected points as a feature and tested different arrangements of it and the four statistical moment vectors. The best results were achieved by using the number of detected points and the 128-dimensional vectors for average, variance and skewness moments, for a total of 385 elements in the final feature vectors.

#### C. Maximally Stable Extremal Regions (MSER)

In 2002, Matas et al. [19] presented the Extremal Regions (ER) concept, and proposed the MSER algorithm to detect ERs. An MSER detector finds regions that are stable over a wide range of thresholds t of a gray-scale image I to a binary image E_t. An ER is thus a connected region in E_t with little size change across several thresholds (t−Δ < t < t+Δ). As the t increases, the MSER detects only dark regions (called MSER+), whereas bright regions (called MSER-) are obtained by inverting the intensity image.

The regions are defined solely by an extremal property of the intensity function in the region and on its outer boundary. MSER do not seek a global or “optimal” threshold, but all thresholds are tested, and the stability of the connected components evaluated. In some parts of images, multiple stable thresholds exist, and a system of nested subsets is the MSER output [19].

After using the MSER detector, we use SURF as a descriptor. 64- and a 128-dimensional feature vectors for each identified region were tried out. As the earlier cases, statistical moments average, variance, skewness and kurtosis is extracted,
generating a 64- or 128-dimensional vector for each statistical moment. As for SIFT and SURF, we used the number of detected regions as a feature, and assessed the possible arrangements of it and the four statistical moment vectors. The best results were achieved by using the number of detected regions and the 64-dimensional vectors for average, variance and skewness moments, for a total of 193 elements in the final feature vectors.

D. Local Binary Patterns (LBP)

The original LBP proposed by Ojala et al. [15] in 1996 labels the pixels of an image by thresholding a 3×3 neighborhood of each pixel with the center value. Then, considering the results as a binary number and the 256-bin histogram of the LBP labels computed over a region, they use this LBP as a texture descriptor. The LBP operator \( \text{LBP}_{P,R} \) produces \( 2^P \) different binary patterns that can be formed by the \( P \) pixels in the neighbor set on a circle of radius \( R \). However, certain bins contain more information than others, and as a result, it is possible to use only a subset of the \( 2^P \) LBPs. These fundamental patterns are known as uniform patterns.

Accumulating patterns having more than two transitions into a single bin yields an LBP operator, denoted \( \text{LBP}_{P,R}^{u1} \), with fewer than \( 2^P \) bins. For example, the number of labels for a neighborhood of 8 pixels is 256 for the standard LBP, but 59 for \( \text{LBP}_{P,R}^{u2} \). Then, a histogram of the frequency of the different labels produced by the LBP operator can be built [15]. We tried out different configurations of LBP operators, but the one that produced the best results was the \( \text{LBP}_{P,R}^{u2} \) with a feature vector of 59 components.

In 2002, LBP variants were proposed in [20]. \( \text{LBP}^{r1}_{P,R} \) and \( \text{LBP}^{r1v2}_{P,R} \) have the same \( \text{LBP}_{P,R} \) definition, but they have only 36 and 10 patterns, respectively. \( \text{LBP}^{r1}_{P,R} \) accumulates all binary patterns in only one bin, which keep the same minimum decimal value \( \text{LBP}_{P,R}^{r1} \), when their \( P \) bits are rotated. \( \text{LBP}^{r1v2}_{P,R} \) combines the definitions of \( \text{LBP}^{u2}_{P,R} \) and \( \text{LBP}^{r1}_{P,R} \). Thus, it uses only the uniform binary patterns and accumulates those that keep the same minimum decimal value \( \text{LBP}_{P,R}^{r1v2} \) in only one bin when their \( P \) bits are rotated.

E. Local Phase Quantization (LPQ)

Proposed by Ojansivu and Heikkila [21] in 2008, LPQ is based on quantized phase information of the Discrete Fourier Transform (DFT). It uses the local phase information extracted using the 2-D DFT or, more precisely, a Short-Term Fourier Transform (STFT) computed over a rectangular \( M \times M \) neighborhood \( N_x \), at each pixel position \( x \) of the image \( f(x) \). The quantized coefficients are represented as integer values between 0-255 using binary coding. These binary codes will be generated and accumulated in a 256-bin histogram, similar to the LBP method [20]. The accumulated values in the histogram will be used as the LPQ 256-dimensional feature vector.

Similar to the Local Binary Pattern from Three Orthogonal Planes (LBP-TOP) presented by Zhao and Pietikainen [22] for LBP, in 2011, Paivarinta et al. [23] proposed Local Phase Quantization from Three Orthogonal Planes (LPQ-TOP) for LBP. LPQ-TOP was also used here. Actually, LPQ-TOP applies the original LPQ version on the XY, XZ and YZ plans of dynamic images and concatenates the LPQ histograms, for a total of 768 elements. As we have static images, we use only the 256 elements for the XY plan. The main difference here consists in the fact that LPQ and LPQ-TOP variants use different default values for their parameters, and thus complement each other.

F. Gray Level Co-Occurrence Matrix (GLCM)

By definition, a GLCM is the probability of the joint occurrence of gray-level \( i \) and \( j \) within a defined spatial relation in an image. That spatial relation is defined in terms of a distance \( d \) and an angle \( \theta \). From this GLCM, some statistical information can be extracted. In our experiments, we tried different values of \( d \), as well as different angles. The best setup we found was \( d = 6 \) and \( \theta = [0, 45, 90, 135] \). In our experiments, we considered the following six descriptors: Contrast, Energy, Entropy, Homogeneity, 3rd Order Moment, and Maximum Likelihood. With that, we arrived at a feature vector with 24 components.

G. Gabor filters

In this study we have used the same setup proposed in [24]. The Gabor wavelet transform is applied on the image with 10 scales (0.4) and 8 (0.7) orientations through the use of a mask with \( 64 \times 64 \) pixels, which results in 80 sub-images. For each sub-image, 3 moments are calculated: mean, variance, and skewness. Thus, a 240-dimensional vector is used for Gabor textural features.

H. Summary of the descriptors

Table I summarizes the descriptors used to create the classifiers. It includes the dimensionality of the feature vectors from which we achieved the best recognition rates, and the average time to compute them for a single image and the computational complexity. This time was measured using a computer with an Intel Core i7 processor (2.2GHz), 8GB (1333MHz DDR3) RAM memory, and a Mac OS X (version 10.8.4) operating system.

<table>
<thead>
<tr>
<th>Descriptor</th>
<th># Features</th>
<th>Time (secs)</th>
<th>Comp. complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>SURF</td>
<td>385</td>
<td>0.38</td>
<td>( O(N \log_2 N) )</td>
</tr>
<tr>
<td>MSER-SURF</td>
<td>193</td>
<td>1.66</td>
<td>( O(N \log_2 N) )</td>
</tr>
<tr>
<td>SIFT</td>
<td>129</td>
<td>1.57</td>
<td>( O(N) )</td>
</tr>
<tr>
<td>LPQ</td>
<td>256</td>
<td>0.34</td>
<td>( O(N^2 \log_2 N) )</td>
</tr>
<tr>
<td>LPQ-TOP</td>
<td>256</td>
<td>0.93</td>
<td>( O(N^2 \log_2 N) )</td>
</tr>
<tr>
<td>( \text{LBP}_{P,R}^{u2} )</td>
<td>59</td>
<td>0.30</td>
<td>( O(N) )</td>
</tr>
<tr>
<td>( \text{LBP}_{P,R}^{r1v2} )</td>
<td>36</td>
<td>0.33</td>
<td>( O(N) )</td>
</tr>
<tr>
<td>( \text{LBP}_{P,R}^{r1u2} )</td>
<td>10</td>
<td>0.31</td>
<td>( O(N) )</td>
</tr>
<tr>
<td>GLCM</td>
<td>24</td>
<td>1.2e-5</td>
<td>( O(N) )</td>
</tr>
<tr>
<td>GABOR</td>
<td>240</td>
<td>0.9e-4</td>
<td>( O(N) )</td>
</tr>
</tbody>
</table>

Time is measured in seconds. N means the total of image pixels.
IV. FUSION RULES

As our proposal applies a two-levels fusion strategy, we use the well-known fusion rules majority voting, borda count, sum, average, product, maximum, and minimum. Each of them produces different results under diverse contexts. It is worth to notice majority voting and borda count use only the predicted class label while all of the other need probabilities for each possible class [9], [11].

For equations 2 to 8, \( s_q \) is the questioned sample, \( V_q \) is the feature vector for \( s_q \) used by the \( K \) classifiers \( C_k \) \((k = [1, K])\), \( C_l \) is one of the \( m \) possible classes \((l = [1, m])\), \( P(V_q|C_l) \) is the probability density function, \( P(C_l) \) is the probability of occurrence \( a \ priori \) of \( P(V_q|C_l) \), and \( P(C_l|V_q) \) is the probability \( a \ posteriori \). All of these fusion rules take into account the Bayesiana theory and assigns \( s_q \) to the predict class \( C_l \) with the higher probability \( a \ posteriori \) among all of classes \( C_l \) (Eq. 1) [9], [11].

\[
s_q \rightarrow C_l \quad \text{if} \quad P(C_l|V_q) = \max_{l=1}^{m} P(C_l|V_q) \tag{1}
\]

Majority voting is the simplest rule for which each classifier has a vote (Eq. 2). For each sample \( s_q \), the total of votes for each possible class \( C_l \) is calculated. Then, the class with the higher number of votes is taken as the predicted class \( C_l \) for \( s_q \). From Eq. 3, the vote of each classifier \( C_k \) is given to class \( C_l \) with the highest \( a \ posteriori \) probability [9], [11].

\[
s_q \rightarrow C_l \quad \text{if} \quad \Delta_{lk} = \max_{k=1}^{K} \Delta_{lk}, \text{ com} \quad \Delta_{lk} = \begin{cases} 1, & \text{if } P(C_{lk}|V_q) = \max_{j=1}^{m} P(C_{lj}|V_q) \\ 0, & \text{otherwise.} \end{cases} \tag{2}
\]

Borda count was proposed by Black [10] and needs a raking of the possible classes provided by each classifier. Each ranking position has a weight that will be assigned to the class for which a classifier ranked it in that place. The predicted class \( C_l \) is defined by the higher value for the sum of weights assigned by all of classifiers [10], [25]. Fig. 2 presents an example to Borda count rule for which we have five classes and three classifiers. After summing the weights (column Rank in Fig. 2 (a-c)) assigned by the three classifiers, the class two is taken as the winner (Fig. 2 (d)) [10], [25].

Product combines the classifiers probabilities output for each class \( C_l \) by multiplying them (Eq. 4), while the sum and average rules does the same by adding (Eq. 5) and averaging (Eq. 6) the probabilities, respectively [9], [11].

\[
s_q \rightarrow C_l \quad \text{if} \quad \prod_{k=1}^{K} P(C_{lk}|V_q) = \max_{l=1}^{m} \prod_{k=1}^{K} P(C_{lk}|V_q) \tag{4}
\]

\[
s_q \rightarrow C_l \quad \text{if} \quad \sum_{k=1}^{K} P(C_{lk}|V_q) = \max_{l=1}^{m} \sum_{k=1}^{K} P(C_{lk}|V_q) \tag{5}
\]

\[
s_q \rightarrow C_l \quad \text{if} \quad \prod_{k=1}^{K} P(C_{lk}|V_q) = \max_{l=1}^{m} \prod_{k=1}^{K} P(C_{lk}|V_q) \tag{6}
\]

From the presented fusion rules which use probabilities in a rage \([0,1]\), sum, average, maximum and minimum can be ranked based on the way the fusion is done and their severity degree (Eq. 9) [9], [11].

\[
\prod_{k=1}^{K} P(C_{lk}|V_q) \leq \max_{k=1}^{K} P(C_{lk}|V_q) \leq ... \leq \frac{1}{K} \sum_{k=1}^{K} P(C_{lk}|V_q) \leq \min_{k=1}^{K} P(C_{lk}|V_q) \tag{9}
\]

V. PROPOSED METHOD

A general overview of our proposed method for forest species classification based on a two-levels fusion strategy is depicted in Fig. 3 and 4, in which a divide-and-conquer idea takes place. Fig. 3 illustrates the process in which a pool of classifiers is created. Each classifier \( C_k \) \((k = 1..N)\) is built on different descriptors (set of features \( F_k, k = 1..N \)).
presented in Table I. Samples $s_i$ in the training set are divided into $M$ non-overlapping sub-images $s_{ij}$ ($j = 1..M$) with identical sizes from which the features vectors $u_{ij}$ ($j = 1..M$) are extracted to be used to train the model $C_k$. The same happens to each questioned sample $s_q$ to be classified (Fig. 4). These pieces are themselves smaller but similar to the original instances.

Such a strategy is an interesting way to cope with such a variability since smaller images tend to be more homogeneous by them selves, hence, an easier classification problem. At the same time, we can also observe that different sets of homogeneous (sub)images are created, but the classifiers are able to cope with them. A good example of that is shown in Fig. 5. By splitting Fig. 5(b) clearly we can notice that (sub)Fig. 5(b.1) is more similar to (sub)Fig. 5(a.1) and 5(a.2), as well as (sub)Fig. 5(b.2) is more similar to (sub)Fig. 5(c.1) and 5(c.2). The same happens to (sub)Fig. 5(e.1) and 5(e.3) with all of the (sub)Fig. 5(d), and (sub)Fig. 5(e.2) and 5(e.4) with all of the (sub)Fig. 5(f).

After classifying a sub-image $s_{qj}$ ($j = 1..M$) based on its features vector $v_{qj}$ ($j = 1..M$), each classifier $C_k$ produces a individual decision $D_{qj}$ ($j = 1..M$). All individual decisions $D_{qj}$ are combined to generating a new partial decision $D_k$ for the classifier $C_k$ to the original questioned image $s_q$ at the Fragment Fusion Level. After that, at the second phase, we combine the previous $D_k$ predictions at the Classifier Fusion Level and produce a final decision $D'$ for the original questioned image $s_q$.

The literature has shown that such a combination can considerably improve the results when complementary classifiers are used [26]. Basically, the proposed method classifies the whole image sample throughout a fusion scheme based on two levels. In the lower level, by combining the results for the different sub-images we are able to deal with the inherent intra-class problem variability. In the higher level, the method explores the combination of diverse classifiers trained on complementar statistical, structural and spectral information presented in Section III. In both levels we have combined the previous output using the well-known fusion rules presented in Section IV.

**VI. Experimental Results**

While developing our experiments, we evaluated a diverse set of parameters, such those mentioned in Section III for each descriptor. Furthermore, the best number of sub-images, and the more effective set of classifiers and the fusion rules are parameters of the system which were investigated. Our experiments are divided into three parts. First, we assess each descriptor independently; next, we present the results for all
the 1023 possible combination of our 10 classifiers; and finally, in the third part, we discuss the two-level classifier fusion strategy and the improvement of the results achieved.

In all the experiments, Support Vector Machines (SVM) were used as classifier since it had achieved the highest recognition rates among those machine learning models tested in our preliminary evaluation. Moreover, the use of other machine learning models would imply in an additional level and a more complex system architecture. For SVM, various kernels were tried, and the best results were achieved using a Gaussian kernel. Parameters $C$ and $\gamma$ were determined through a grid search. The Recognition Rate that we used were $FP$, $FN$, $TP$, and $TN$ represent False Positive, False Negative, True Positive, and True Negative, respectively. These statistics are defined in the $2 \times 2$ confusion matrix depicted in Fig. 6.

To perform the experiments, we divided the 2240 images into two disjoint sets for training (60%) and testing (40%). The training set were composed by 12 out of the 20 images per class, in a total of 1344 images, and the testing set were composed by 8 images, in a total of 896 images. In order to show that the choice of images used in each subset does not have a significant impact on the recognition rates, each experiment was performed five times with different subsets (randomly selected) for training and testing. The small standard deviation ($\sigma$) values show that the choice of the images for each dataset is not an issue.

One of the limitations with SVMs is that they do not work in a probabilistic framework. There are several situations where it would be very useful to have a classifier producing an a posteriori probability $P(class|input)$. In our case, we were interested in estimating the probabilities of combining partial decisions in both fusion levels. In this work, we adopt the strategy proposed by Platt in [27].

### A. Single classifiers

As stated previously, our first experiment consisted in performing forest species identification using the single classifiers on the whole images. Table II reports the average performance on five folds for each individual classifier. As we can see, the best results achieved $92.46\%$ ($\sigma=0.73$) using the sum fusion rule and classifiers SURF, MSER-SURF, LPQ-TOP, and LBP. The kind of texture we are dealing with in this study favors the keypoint-based descriptors since the images contain several visible gradient points that are invariant to any illumination changes, such as color, intensity, or shift (Fig. 1).

### B. Combination of single classifiers

The combination of classifiers is an active area of research, and many studies have been published, both theoretical and empirical, demonstrating the advantages of the combination paradigm over the individual classifier models [28]. With that in mind, and considering the reduced number of classifiers, the second part of our experiments consisted in combining the 10 classifiers built on the whole images presented in the previous section. While doing it, we used the brute force strategy to explore all possible combinations among our 10 classifiers. The best results achieved $96.63\%$ ($\sigma=0.96$) by using the Product rule for the classifiers SURF, MSER-SURF, LPQ-TOP, SIFT, Gabor, LBP, LBP, GLCM. It is worth to mention that these results represent a possible single level fusion strategy and rates inferior to those presented in the following section.

### C. Two-level-fusion Strategy

Following with our strategy described in Section V, we tried to increase each individual classifier rates to improve their collective rates. Our strategy can be reinforced by the example presented in Fig. 5. This picture shows that smaller fragments generate more similar sub(images), but it also generates different groups of them. As expected, in general, recognition rates for smaller pieces are worse than those achieved for the whole image (Fig. 7), but the final rates after the fusion of their individual decision usually are improved (Fig. 8). Taking it into account, we proceeded with the slicing process and analysed different sizes of sub-images, while the final recognition rates were improved.

Finally, the best results achieved for the proposed strategy was $98.47\%$ ($\sigma=1.06$) using the sum fusion rule and classifiers SURF, MSER-SURF, LPQ-TOP, LBP, and LBP, only $1.48$
Fig. 7. Best classifiers and their recognition rates while classifying image fragments for different tested sub-image sizes.

Fig. 8. Best classifiers and their recognition rates achieved for the whole image, after combining the fragment results for the tested sub-image sizes.

### TABLE III

<table>
<thead>
<tr>
<th>Author</th>
<th># Classes</th>
<th>Features/Recognition Rates (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>[31]</td>
<td>52/52</td>
<td>Gray Level A. Matrix 93.0</td>
</tr>
<tr>
<td>[1]</td>
<td>112/112</td>
<td>LPQ+LBP 86.5</td>
</tr>
<tr>
<td>[32]</td>
<td>112/112</td>
<td>LPQ+GLCM 93.2</td>
</tr>
<tr>
<td>[33]</td>
<td>112/112</td>
<td>LPQ-Blackman 95.0</td>
</tr>
<tr>
<td>[29]</td>
<td>112/112</td>
<td>CNN 97.5</td>
</tr>
<tr>
<td>[2]</td>
<td>68/44</td>
<td>DSC 93.2</td>
</tr>
<tr>
<td>This work</td>
<td>112/112</td>
<td>Two-level fusion 98.5</td>
</tr>
</tbody>
</table>

**D. Analysing results**

In order to get a better insight of these results we analyze some classes that had the higher error rates. Fig. 9 shows some samples of these classes after applying the sum fusion rule. Fig. 9(a) presents the single instance from *Grevillea robusta* which was classified as *Grevillea sp*, both of them belonging to the same gender (*Grevillea*). Fig. 9(c) shows samples of *Eucalyptus grandis*. This class of forest species presented the higher number of misclassified instances, a total of three out of 20 images (15%). In this case, there were confusion with specie *Eucalyptus saligna*, which is also from the same gender *Eucalyptus*.

Even taking the presented Oracle definition, three out of 2,240 instances (0.14%) from the Softwood species would be misclassified, as showed in Fig. 10. Two out of 20 instances (10%) from *Taxodium distichum* would be classified as *Sequoia sempervirens*, both belonging to the family *Taxodiaceae*. A single instance from *Cupressus lindleyi* would be classified as *Ginkgo biloba*.

**VII. CONCLUSION**

In this work, we have presented a two-level-fusion strategy to identify microscopic images of forest species. To build the pool of classifiers, we used 10 different descriptors, including the classical texture-based and the keypoint-based features. The latter, which are successfully applied for object tracking and recognition, have been proven useful in recognizing such a texture. Results reported in Section VI indicate that the proposal surpass the previous works in terms of recognition rates and computational cost.

Regarding the two-level-fusion strategy, the recognition rate was 98.47% ($\sigma=1.06$) using the sum fusion rule to combine the classifiers SURF, MSER-SURF, LPQ-TOP, LBP$^u$ and LBP$^r$. There was a gain of 6.01% percentage points relating to the percentage points inferior to the oracle rate (99.95%, $\sigma=0.07$)$^1$. This small difference shows the effectiveness of combining strategy. Table III summarizes the recent results published on the literature using microscopic images of forest species. Comparing our results with the literature we outperformed all of them and improve the recognition rates in 1.2 percentual points. It is worth to note that the previous best results from Hafemann et al. [29] used CNN which is very more expensive in terms of computational cost.

---

$^1$Oracle rate indicates the upper limit for each set of combined classifiers. It is computed considering that the better classifier is always selected.
of generating pools of classifiers. For future works, we also plan to investigate automatic methods relating to the previous best results. For the present problem, even the oracle did not achieve 100% accuracy. More research should be done because the best single classifier improved by 1.84 percentage points relating to the previous best results. In spite of the improvement produced by the proposed method when compared to the single classifiers and also their combination, it is clear that we are closer to the oracle, but certainly, more research should be done because even the oracle did not achieve 100% for the present problem. For future works, we also plan to investigate automatic methods of generating pools of classifiers.

REFERENCES


