Machine Vision and Applications (2015) 26:279–293
DOI 10.1007/s00138-015-0659-0

ORIGINAL PAPER

Forest species recognition based on dynamic classifier selection and dissimilarity feature vector representation

J. G. Martins · L. S. Oliveira · A. S. Britto Jr. · R. Sabourin

Received: 10 April 2014 / Revised: 25 November 2014 / Accepted: 30 December 2014 / Published online: 29 January 2015

Abstract Multiple classifiers on the dissimilarity space are proposed to address the problem of forest species recognition from microscopic images. To that end, classical texture-based features such as Gabor filters, local binary patterns (LBP) and local phase quantization (LPQ), as well as two keypoint-based features, the scale-invariant feature transform (SIFT) and the speeded up robust features (SURF), are used to generate a pool of diverse classifiers on the dissimilarity space. A comprehensive set of experiments on a database composed of 2,240 microscopic images from 112 different forest species was used to evaluate the performance of each individual classifier of the generated pool, the combination of all classifiers, and different dynamic selection of classifiers (DSC) methods. The best result (93.03 %) was observed by incorporating probabilistic information in a DSC method based on multiple classifier behavior.

Keywords Forest species recognition · Microscopic images · Dynamic classifier selection · Dissimilarity representation · Texture analysis

J. G. Martins · L. S. Oliveira (✉)
Federal University of Parana (UFPR), R. Rua Cel. Francisco H. dos Santos, 100, Curitiba, PR 81531-990, Brazil
e-mail: lesoliveira@inf.ufpr.br

J. G. Martins
e-mail: martins@utfpr.edu.br

A. S. Britto Jr.
Pontifical Catholic University of Parana (PUCPR), R. Imaculada Conceição, 1155, Curitiba, PR 80215-901, Brazil
e-mail: alceu@ppgia.pucpr.br

R. Sabourin
Ecole de Technologie Superieure, 1100 rue Notre Dame Ouest, Montreal, QC, Canada
e-mail: robert.sabourin@etsmtl.ca

1 Introduction

Wood is a feedstock used to manufacture a wide range of products. It may be categorized by its different applications, and according to their physical, chemical, and anatomical characteristics, which leads to great variations in their prices. The safe trade of log and timber has become a significant challenge since buyers must certify they are buying the correct material, while supervising agencies have to certify that wood has been not extracted irregularly from forests. All this implies millions of dollars in costs, and aims to prevent fraud involving a wood trader attempting to mix noble species with cheaper ones, or even trying to export wood species listed as endangered.

In this context, researchers from different fields have perceived a vast area for research, and have proposed different alternatives to deal with the problem of forest species recognition. Tou et al. [40,41] report two forest species classification experiments in which texture features are used to train a neural network classifier. They report recognition rates ranging from 60 to 72 %, for five different forest species.

Khalid et al. [18] present a system for recognizing 20 different Malaysian forest species. Similarly to Tou et al., the classification process is based on a neural network trained with textural features. The database used in their experiments is composed of 1,753 images for training, and only 196 for testing. They report a recognition rate of 95 %. Paula et al. [33] have investigated the use of GLCM and color-based features to recognize 22 different species of Brazilian flora. They propose a segmentation strategy to deal with large intra-class variability. Experimental results show that when color and textural features are used together, results can be improved considerably.

A common problem in most of the aforementioned works is that in their experimental protocol, they consider databases containing only a few classes, without readily available information about their taxonomy. Just recently, more representative datasets have been built and made available for research purposes. Martins et al. [25] introduced a database of microscopic images composed of 112 forest species, and Yusof et al. [46] proposed a dataset with 52 Malaysian forest species. Paula Filho et al. [34] proposed a database of macroscopic images with 42 Brazilian forest species, and in the same study, reported a 96.2 % recognition rate using a single SVM classifier trained with completed local binary pattern (CLPB).

These databases motivated other researchers to investigate different approaches to tackling the problem of forest species recognition. Kapp et al. [17] assessed multiple feature sets using a quadtree-based approach and reported recognition rates of 95 and 88 % for the microscopic [25] and macroscopic [34] databases, respectively. Hafemann et al. [13] took a different approach, and instead of using textural descriptors, they used the images to train a convolutional neural network (CNN). Using the datasets proposed in [25] and [34], they reported accuracies of 95 and 97 %, respectively. Yada et al. [44] reported some experiments using only the hardwood species of the database proposed in [25]. Using 25 species, they achieved a 92 % recognition rate.

Similar techniques have been described in the literature to solve related problems. Yanikoglu et al. [45] proposed a feature set based on shape, color, and textural descriptors to deal with 126 different plant species available in the ImageCLEF'12 database. In this challenging problem, the authors reported an accuracy of 61 %.

All aforementioned studies are based on image-based processing systems, i.e., the acquisition module captures either a macroscopic or a microscopic image used as input for the classification system. Other researchers have explored different devices to acquire the input signal, e.g., a proper source of radiation to excite the wood surface to analyze the emitted spectrum. These techniques are based mainly on vibrational spectroscopy methods, such as Near Infra-Red (NIR) [42], Mid-IR [28], Fourier-Transform Raman Spectroscopies [23], and fluorescence spectroscopy [36]. Such methods acquisition systems are usually composed of a spectrometer, a laser source, and an optical filter. All these devices should be placed at an angle α and a distance z capable of maximizing the overall power of the signal captured by the spectrometer according to the focal length of its objective lens.

Despite the inherent difficulties in recognizing forest species, such as a huge intra-class variability, the elevated number of classes also imposes an extra challenge. An interesting alternative that has successfully been used to face the large number of classes is the dissimilarity approach [3,21]. This strategy takes into account a dichotomy transformation which makes it possible to transform any n-class pattern classification problem into a 2-class problem. This property allows us to design a classification system, even just with a limited number of samples from a large number of classes. Another advantage of the dissimilarity approach is that it does not require a learning model to be retrained each time a new forest species is presented to the system.

Much like the dissimilarity approach, another concept that has been successfully deployed by the pattern recognition community is the multiple classifier system (MCS). After years of research, results lead to the conclusion that creating a monolithic classifier to cover all the variability inherent in most classification problems is somewhat unfeasible. In light of this, researchers have investigated strategies to create pools of classifiers [5,15,39], how to combine these classifiers [19], and how to dynamically select the best classifier or ensemble of classifiers for a given problem [10,16,20,43].

In this paper, we address the problem of forest species recognition using microscopic images. The proposed strategy takes into account multiple classifiers on the dissimilarity space. The contribution of this work is twofold. First, we assess different families of textural descriptors for the problem of forest species classification. Besides the most used textural descriptors, such as the local binary pattern (LBP) and the local phase quantization (LPQ), we investigate the use of keypoint detectors, such as the scale-invariant feature transform (SIFT) and the speeded up robust features (SURF), as well. Our motivation lies in the fact that the microscopic images feature certain structures (resiniferous channels and vessels) that may be described by keypoint detectors.

Secondly, we explore the use of these different classifiers in the dissimilarity space. As earlier stated, the dissimilarity approach can deal with a limited number of samples per class, and the machine learning model does not need to be trained each time a new class is presented to the system. This is an important characteristic for this application since the number of species available in nature is very large. Besides the traditional combination of classifiers, in this work, we explore the concept of dynamic selection of classifiers (DSC). We assess most of the strategies available in the literature [16], such as those based on accuracy [43], probabilistic information [9], classifier behavior [10], and oracle information [20]. After analyzing the results produced by all DSC methods, we realized that the probabilistic information used in the method proposed by Giacinto and Roli [9] was promising; with that in mind, we modified another interesting method, the multiple classifier behavior (MCB) [10], and incorporated this probabilistic information instead of using only the local overall accuracy. This modification compared favorably against all other strategies we tested.

Through a set of comprehensive experiments on a recently proposed database composed of 2,240 microscopic images from 112 different forest species [25], we show that the keypoint descriptors are a good alternative for this kind of image. Both SIFT and SURF exceed all the other textural descriptors. The best result of a single descriptor was achieved by the classifier trained with SURF, which achieved an 89.14 % recognition rate. For the DSC experiments, they showed that the modified version of the method proposed by Giacinto and Roli [10] achieved 93.03 % recognition rate, which was the best result in this study.

This paper is structured as follows: Section 2 introduces the concepts related to the dissimilarity-based representation. Section 3 introduces the DCS techniques. The database used in our experiments is briefly described in Sect. 4. A brief explanation of the descriptors and feature vectors is presented in Sect. 5. Section 6 presents the proposed method, while Sect. 7 reports our experiments. Finally, Sect. 8 concludes the work.

2 Dissimilarity-based representation

The concepts of similarity, dissimilarity, and proximity have been discussed in the literature from different perspectives [11,35,38]. Pekalska and Duin [35] introduced the idea of representing relations between objects through dissimilarity, which they call dissimilarity representation. This concept describes each object by its dissimilarities to a set of prototype objects, called the representation set R. Each object χ is represented by a vector of dissimilarities $D(\chi, R) = [d(\chi, r_1), d(\chi, r_2), \ldots, d(\chi, r_n)]$ to the objects $r_j \in R$.

Let R be a representation set composed of n objects. A training set T of m objects is represented as the $m \times n$ dissimilarity matrix $D(T, R)$. In this context, the usual method of classifying a new object χ represented by $D(\chi, R)$ involves using the nearest neighbor rule. The object χ is classified into the class of its nearest neighbor, that is the class of the representation object r_j given by $d(\chi, r_j) = \min_{r \in R} D(\chi, R)$. The key point here is that the dissimilarities should be small for similar objects (belonging to the same class) and large for distinct objects.

As pointed out in [35], the idea of dissimilarity is quite interesting when a feasible feature-based description of objects might be difficult to obtain or inefficient for learning purposes. In the case of textures, several different descriptors have been proposed allowing the modeling of intra- and extra-class variation.

In that context, in this work we adopt the strategy proposed by Bertolini et al. [2] which combines feature-based descriptions with the concept of dissimilarity. The idea is to extract the feature vectors from both questioned and reference forest species to then compute a dissimilarity feature vector. If both samples come from the same species (genuine), then all the components of such a vector should be close to 0, otherwise (forgery), the components should be far from 0. Of course, this is totally true under favorable conditions. As with any other feature representation, the dissimilarity feature vector can be affected by the intra-class variability. Such a variability could generate values far from zero when measuring the dissimilarity of samples taken from the same forest species.

Before deploying a dissimilarity-based system, we must first create the dataset to train a machine learning model. Algorithm 1 summarises the training procedure.

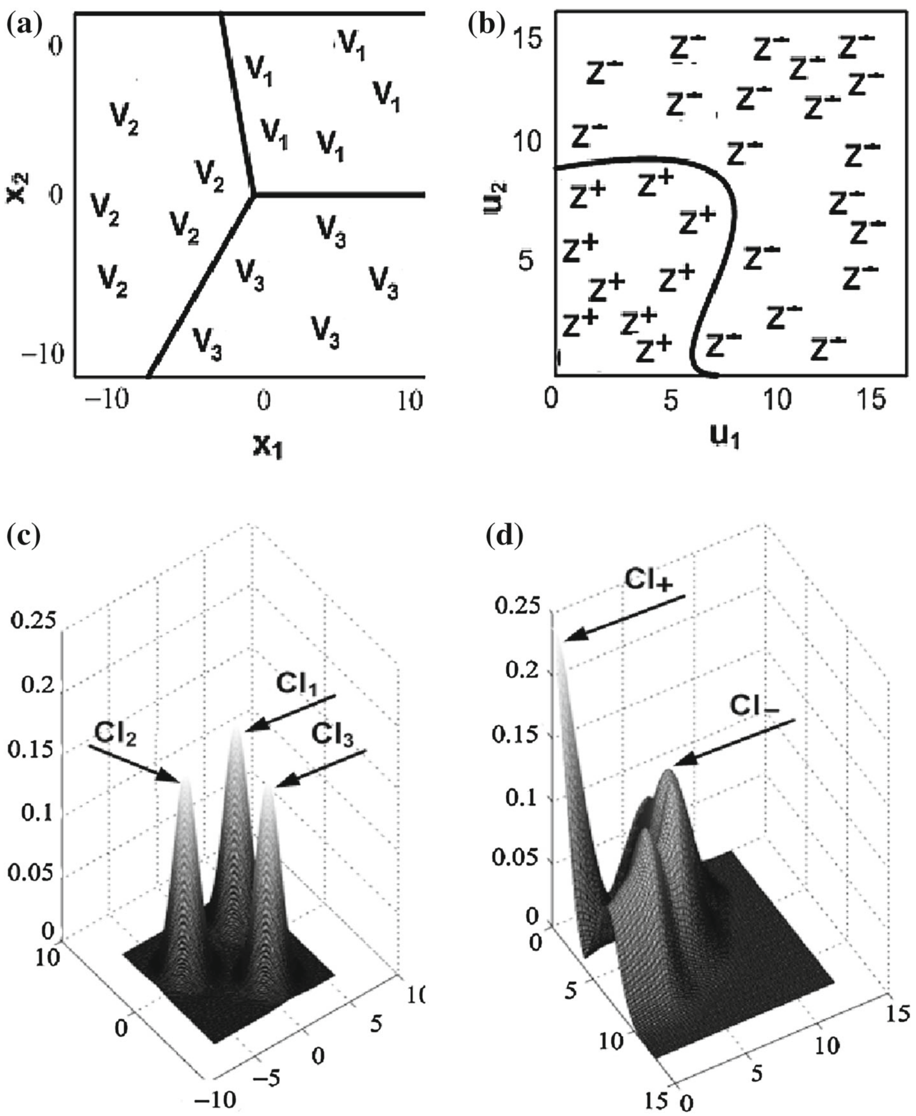
Algorithm 1 Training

```
Input: Classes that will be used for training the machine learning
  model. {As stated before, an advantage of this approach is that we
  can use disjoint sets for training and testing.}
Output: Machine Learning model $\Phi$ trained to discriminate between
  positive and negative dissimilarity feature vectors.
PostiveSamples = 0
for i = 1 to NumberTrainingClasses do
  Choose randomly a set (R) of $n$ images to be used as references.
  //Extract textural features from the reference images
  for j = 1 to $n$ do
    $V_{ij}$ = Feature vector extracted from $R_j$
  end for
  //Compute the positive dissimilarity samples. Difference among
  the vectors of the same class
  for j = 1 to $n$ do
    for k = j+1 to $n$ do
      $Z^+ = |V_{ij} - V_{ik}|$
      PositiveSamples++
    end for
  end for
end for
//Compute the negative dissimilarity samples. Difference among the
vectors of different classes
NegativeSamples = 0
while (NegativeSample < PositiveSamples) do
  Let $a$ = Feature vector draw randomly from $V$.
  Let $b$ = Feature vector draw randomly from $V$, but from a different
    class than $a$
  $Z^- = |a - b|$
  NegativeSamples++
end while
$\Phi$ = Machine Learning Training($Z^+$, $Z^-$)
return $\Phi$
```

The result of this procedure can be visualized in Fig. 1. We can see that the dichotomy transformation affects the geometry of the distribution. In the feature space, multiple boundaries are needed to separate all the writers. In the dissimilarity space, by contrast, only one boundary is necessary, since the problem is reduced to a 2-class classification problem.

In this case, we have used three classes in the feature space (Fig. 1a, c), which were transformed into the dissimilarity space (Fig. 1b, d). As will be discussed later, the machine learning model used in this work is a support vector machine (SVM).

Fig. 1 The dichotomy transformation from the feature space (**a**, **c**) to the dissimilarity space (**b**, **d**)



After the dissimilarity classifiers are trained, testing is done using the Algorithm 2. In line 6 of Algorithm 2, several functions can be used to combine the partial decision of the classifier. In our experiments, the function that provided the best results was the Max rule [19].

Algorithm 2 Testing

```
1: Input: feature vector of the testing pattern ($Q$), $k$ references ($S$) of
   all testing classes enrolled into the system.
2: Output: The class to which $Q$ should be assigned.
3: for i = 1 to NumberTestingClasses do
4:     $Z_k = |Q - S_{ik}|$ {Compute the dissimilarity vectors}
5:     Outputs = $\Phi(Z_k)$ {Classify the $k$ dissimilarity vectors using the
       previously trained classifier}
6:     PartialDecisions = Combine(Outputs) {Combine the $k$ outputs
       into a single partial decision}
7: end for
8: return max(PartialDecisions) {Returns the class that maximizes
   the partial decisions}
```

3 Dynamic selection of classifiers

By definition, an MCS is composed of a pool of base classifiers (C) that may be created through different methods, such as, Bagging [5], Boosting [39], or Random Subspaces [15]. In the case of dissimilarity-based classifiers, they can be created using different dissimilarity spaces. The goal of dynamic selection then is to find a subset of classifiers C^* ($C^* \subseteq C$) that correctly classify a given unknown pattern Q. In the literature [16], the subset C^* may be composed of a single classifier [9,43] or an ensemble of classifiers [7,20].

In general, selection is performed by estimating the competence of the classifier available in the pool on local regions of the feature space. The feature space is divided into different partitions, and the most capable classifiers for a given unknown pattern Q, are determined. Regarding the competence measures, the literature shows that they may be based on accuracy (overall local accuracy or local class accuracy) [43], probabilistic information [9], classifier behavior computed on the output profiles [7,10], and oracle information [20,22].

In our study, we investigated all the aforementioned strategies. It is worth mentioning that the oracle-based methods, such as KNORA [20], which usually show good performance in several pattern recognition tasks, did not behave well in our tests. It would appear that these methods depend on larger pools of base classifiers. As stated earlier, our best results were achieved using the concepts of accuracy presented by Woods et al. [43] and the probabilistic- and behavior-based measures introduced by Giacinto and Roli [9,10]. For readers interested in DSC, a recent review can be found in [16].

4 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 μ ($1\mu = 1 \times 10^{-}6\,m$). 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 1,024 × 768 pixels. More details about the database can be found in [25].

Each species has 20 images, for a total of 2,240 microscopic images. Of the 112 available species, 37 are Softwoods and 75 are Hardwoods (Fig. 2). 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. 2a), whereas Hardwoods usually present some large holes, known as vessels (Fig. 2b).

Another visual characteristic of the Softwood species is the growth ring, which is defined as the difference in the thickness of the cell walls resulting from the annual development of the tree. We can see this feature in Fig. 2a. The coarse cells at the bottom and top of the image indicate more intense physiological activity during spring and summer. The smaller cells in the middle of the image (highlighted in light red) represent the minimum physiological activity that occurs during autumn and winter.

It is worth noting that color cannot be used as an identifying feature 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.

5 Features

In this section we briefly describe all the descriptors we used to create the dissimilarity classifiers. These include the most commonly textural descriptors found in the literature, such as local binary patterns (LBP) [29], local phase quantization (LPQ) [31], grey-level co-occurrence matrix [14], Gabor filters [12], and two keypoint descriptors, namely, scale-invariant feature transform (SIFT) [24] and speed-up robust feature (SURF) [1]. Keypoint descriptors, which are usually used for object recognition, are used mainly because

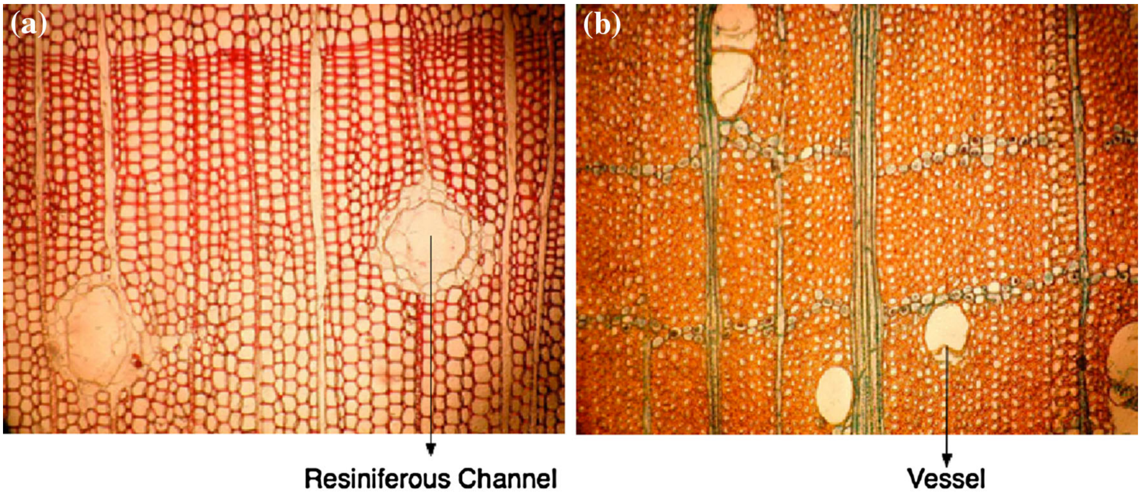


Fig. 2 Samples of the database **a** softwoods and **b** hardwoods

of the nature of the microscopic images. As mentioned earlier, we believe that keypoints extracted from resiniferous channels and vessels (Fig. 2a, b) 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.

5.1 Scale invariant feature transform (SIFT)

A keypoint descriptor is 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, we extracted the statistical moments average, variance, skewness and kurtosis, generating a 128-dimensional vector for each statistical moment. 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, for a total of 129 components in the final feature vectors.

5.2 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 (d_x and $|d_x|$) and in vertical (d_y and $|d_y|$) 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 d_x and $|d_x|$ are computed separately for $d_y < 0$ and $d_y \geq 0$. Similarly, the sums of d_y and $|d_y|$ take the sign of d_x 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 extracted 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.

5.3 Maximally stable extremal regions (MSER)

In 2002, Matas et al. [27] 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 ($t - \Delta < t < t + \Delta$). As 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 [27].

After using the MSER detector, the SURF was used as a descriptor. 64- and a 128-dimensional feature vectors for each identified region were tried out. As was the case earlier, we extracted the statistical moments average, variance, skewness and kurtosis, 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.

5.4 Local binary patterns (LBP)

The original LBP proposed by Ojala et al. [29] 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 used this LBP as a texture descriptor. The LPB 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 $\text{LBP}^{u2}_{P,R}$, 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 LBP^{u2}. Then, a histogram of the frequency of the different labels produced by the LBP operator can be built [29]. We tried out different configurations of LBP operators, but the one that produced the best results was the $\mathrm{LBP}^{u2}_{8,2}$, with a feature vector of 59 components.

In 2002, LBP variants were proposed in [30]. LBP^{ri} and LBP^{riu2} have the same $\mathrm{LBP}_{P,R}$ definition, but they have only 36 and 10 patterns, respectively. LBP^{ri} accumulates all binary patterns in only one bin, which keep the same minimum decimal value $LBP^{ri}_{P,R}$, when their P bits are rotated. LBP^{riu2} combines the definitions of LBP^{u2} and LBP^{ri}. Thus, it uses only the uniform binary patterns and accumulates those that keep the same minimum decimal value $LBP^{riu2}_{P,R}$ in only one bin when their P bits are rotated.

5.5 Local phase quantization (LPQ)

Proposed by Ojansivu and Heikkila [31] 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 and 255 using binary coding. These binary codes will be generated and accumulated in a 256-bin histogram, similar to the LBP method [30]. 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) and the volume local binary pattern (VLBP) presented by Zhao and Pietikainen [47] for LBP, in 2011, Paivarinta et al. [32] proposed local phase quantization from three orthogonal planes (LPQ-TOP) and volume local phase quantization (VLPQ) for LPQ. 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 used 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.

5.6 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 θ. From the 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.

5.7 Gabor filters

In this study we have used the same setup proposed in [34]. The Gabor wavelet transform is applied on the image with 5 scales (0..4) and 8 (0..7) orientations through the use of a mask with 64 × 64 pixels (Fig. 4), which results in 40 sub-images. For each sub-image, 3 moments are calculated: mean, variance, and skewness. Thus, a 120-dimensional vector is used for Gabor textural features.

5.8 Summary of the descriptors

Table 1 summarizes the 10 descriptors used to create the dissimilarity 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. This time was measured using a computer with an Intel Core i7 processor (2.2 GHz), 8 GB (1,333 MHz DDR3) RAM memory, and a Mac OS X (version 10.8.4) operating system.

6 Proposed method

Figure 3 presents a general overview of the proposed method. It receives the input pattern to be classified together with the references of all classes enrolled into the system. The feature extraction module is then responsible for extracting all descriptors listed in Table 1. With all feature vectors on hand, the next step consists in computing the dichotomy transformation for all 10 feature spaces in order to create the dissimilarity feature vectors.

These vectors are sent to their respective classifiers, trained according to Algorithm 1. Taking a close look at the dissimilarity-based classification described in Algorithm 2, we may note the presence of a combination step required to combine the results of all references for each class. In our case, the combination rule that produced the best results was the Max rule. Finally, the outputs of all 10 classifiers are used in an MCS strategy. In this work, we evaluated the traditional combination of classifiers, which combines the outputs of all classifiers, as well as DSC strategies. Despite the fact that in this work we used a limited number of classifiers, the proposed framework can handle any number of classifiers in the pool.

Regarding DSC strategies, different strategies reported in the literature were considered in this study. The multiple classifier behavior (MCB) proposed by Giacinto and Roli [10] produced the best results after a slight modification of the original algorithm. To allow greater insight into this method

Fig. 3 The block diagram of the proposed system

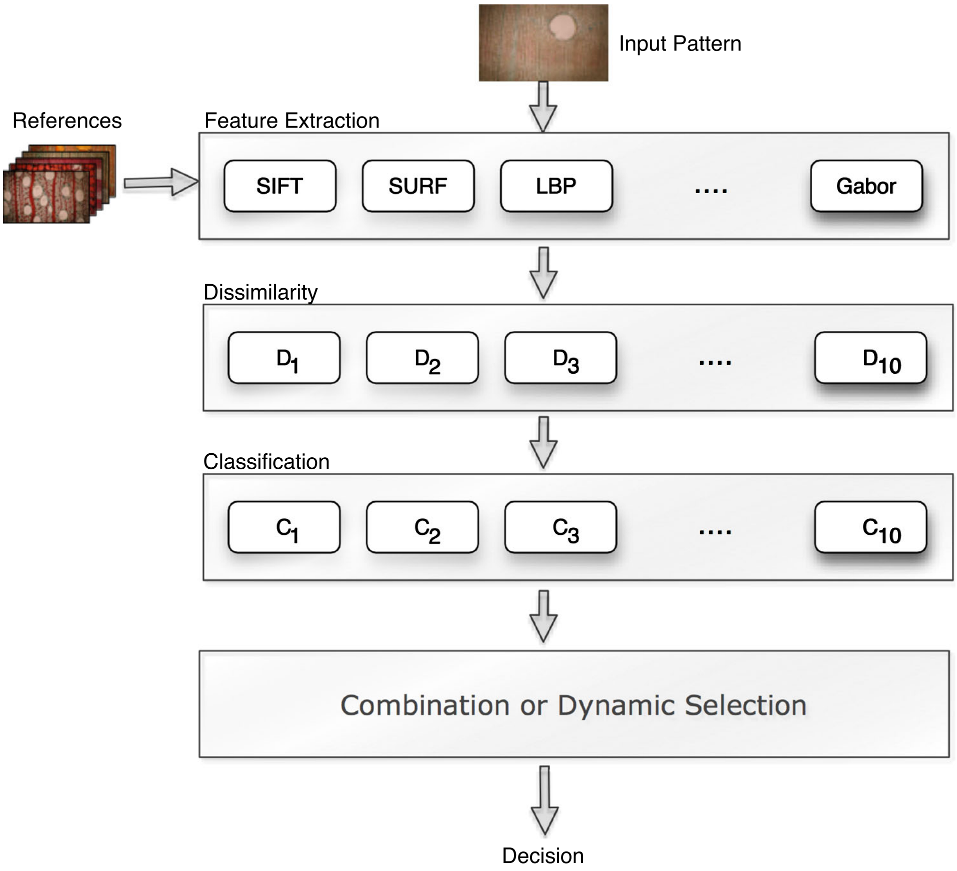


Table 1 Summary of the descriptors

Classifier	# Features	Time (s)
SURF	385	0.38
MSER-SURF	193	1.66
SIFT	129	1.57
LPQ	256	0.34
LPQ-TOP	256	0.93
LBP_{u2}	59	0.30
LBP_{ri}	36	0.33
LBP_{riu2}	10	0.31
GLCM	24	1.2e^{-4}
GABOR	120	0.9e^{-4}

and the proposed modification, we describe the MCB in detail in the next paragraphs.

The MCB is estimated by using a similarity function to measure the degree of similarity of the output profiles of all base classifiers. First, a local region Ψ is defined as the k-nearest neighbors of the unknown pattern in the training set. Then, the similarity function is used as a filter to preselect from Ψ the samples for which the classifiers present similar behavior to that observed for the unknown pattern Q. The remaining samples are employed to select the most accurate classifier by using OLA. Finally, if the selected classifier is significantly better than the others in the pool, based on a defined threshold value, it is then used to classify the unknown pattern Q. Otherwise, all the classifiers are combined using the majority voting rule.

The underpinning concept of the MCB is the vector $\text{MCB}_{\psi} = \{C_1(\psi), C_2(\psi), \ldots, C_M(\psi)\}$. It contains the class labels assigned to the sample ψ but the M classifiers in the pool. The measure of similarity is define in Eq. 1.

$$\text{Similarity } (\psi_1, \psi_2) = \frac{1}{M} \sum_{i=1}^{M} T_i(\psi_1, \psi_2) \tag{1}$$

where

$$T_i(\psi_1, \psi_2) = \begin{cases} 1 & \text{if } C_i(\psi_1) = C_i(psi_2) \\ 0 & \text{if } C_i(\psi_1) \neq C_i(psi_2) \end{cases} \tag{2}$$

Algorithm 3 presents the original MCB method. An important feature of this algorithm can be seen on line 13, where the overall local accuracy of each classifier available in the pool is estimated in the local region of the training set represented by the samples with similar behavior as that observed for the unknown pattern Q. Besides this original

Algorithm 3 MCB OLA method

```
Input: the pool of classifiers $C$; the unknown pattern $Q$; the neighborhood size $K$;
Output: $c^*_Q$, the most promising classifier for the unknown pattern $Q$;
Compute the vector $MCB_Q$ as the class labels assigned to $Q$ by all classifiers in $C$;
Find $\Psi$ as the $K$ nearest neighbors of the test sample $Q$ in the $TrainingSet$;
for each sample $\psi_j$ in $\Psi$ do
    Compute $MCB_{\psi_j}$ as the class label assigned to $\psi_j$ by all classifiers in $C$;
    Compute $Sim$ as the similarity between $MCB_Q$ and $MCB_{\psi_j}$;
    if $(Sim > SimilarityThreshold)$ then
        $\Psi' = \Psi' \cup \psi_j$;
    end if
end for
for each classifier $c_i$ in $C$ do
    Calculate $OLA_i$ as the local class accuracy of $c_i$ on $\Psi'$;
end for
Select the best classifier $c^*_Q = \arg\max_i \{OLA_i\}$;
if ($c^*_Q$ is significantly better than the other classifiers on $\Psi'$) then
    Use the classifier $c^*_Q$ to classify $Q$
else
    Use all classifiers in C to classify $Q$;
end if
```

version, we propose a modified MCB by considering LCA, “a priori” and “a posteriori” schemes to estimate the competence of the classifiers at that point of the algorithm.

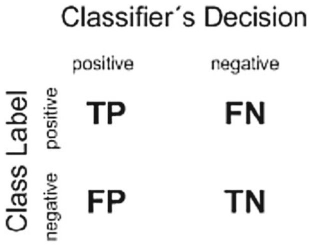
7 Experimental results

Our experiments are divided into three parts. First, we assess each descriptor independently; next, we present the results of the combination of all descriptors, and finally, in the third part, we discuss the different strategies used for the dynamic selection of classifiers; we further show that the modifications we proposed in the original Giacinto method are able to further improve the results achieved by the combination of classifiers.

In all the experiments, support vector machines (SVM) were used as classifiers. Various kernels were tried, and the best results were achieved using a Gaussian kernel. Parameters C and γ were determined through a grid search. The Recognition Rate that we used for evaluation purposes in this work is given by Eq. 3 and is always computed on the testing set.

$$\text{Recognition rate} = 100 - \left(\left(\frac{\text{FP} + \text{FN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}\right) \times 100\right) \quad (3)$$

where FP, FN, TP, and TN represent false positive, false negative, true positive, and true negative, respectively. These statistics are defined in the 2×2 confusion matrix depicted in Fig. 4.



Classifier´s Decision

Class Label	positive	negative
positive	TP	FN
negative	FP	TN

Fig. 4 2×2 confusion matrix

To perform the experiments, we divided the 112 forest species into two disjoint sets for training (60 %, or 68 species) and testing (40 %, or 44 species). Then, the 20 images per class of the training set were further divided into training (12 images) and validation (8 images). Training and validation thus contained the same 68 species. The 60–40 % relation was used because we were interested in evaluating the influence of the number of classes on the training in terms of recognition rates.

In order to show that the choice of classes and 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 (σ) 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 $P(\text{class}|\text{input})$. In our case, we were interested in estimating the probabilities of combining partial decisions in the dissimilarity framework as well as in the DSC context. In this work, we adopt the strategy proposed by Platt in [37].

7.1 Single classifiers

As stated previously, our first experiment consisted in performing forest species identification using the single classifiers. The identification problem consisted of identifying a species I among all the species enrolled in the system. Given an input feature vector x from a texture image S, we determine the species $I_c, c \in 1, 2, \ldots, N$, where N is the number of species enrolled in the system. Hence, $S \in I_c$, if $\max_c\{D_{\text{model}}(\chi, R_c)\}$, where D_{model} is the dissimilarity model trained to return an estimation of posterior probability, which indicates that S and the reference R_c belong to the same species.

When using dissimilarity, one important aspect is to define the number of references that will be used for training (R) and testing (S). In our experiments, the best results were achieved

Table 2 Performance of the single classifiers

Classifier	Identification rate (%)	σ
SURF	89.14	2.39
SIFT	88.47	1.64
MSER-SURF	87.80	2.17
LPQ	86.74	2.07
LPQ-TOP	86.41	1.36
LBP$_{u2}$	66.25	4.67
LBP$_{ri}$	50.74	9.54
GABOR	25.67	2.53
LBP$_{riu2}$	16.49	17.57
GLCM	4.09	1.13

using 12 images per species as references ($R = 11$) to generate positive and negative samples, and 19 images ($S = 19$) for identification. The fusion rule applied to combine the classifier's output was the Max rule, which produced the best results among all the combination rules described by Kittler et al. [19].

Table 2 reports the average performance on five folds for each individual classifier. As we can see, the top 3 classifiers, which are trained using keypoint-based features, surpassed the traditional textural descriptors for this dataset. The kind of texture we are dealing with in this study favors the keypoint-based descriptors since the images contain several visible vessels that can be used as features (Fig. 2). The poor performance of LBP could appear surprising but it is less uncommon that one may expect. Some works in the literature, e.g., [4,8] also show that LBP may be surpassed by other descriptors. Since LBP represents an input image by building statistics on the local micro pattern variation, such as bright/dark spots, edges, and flat areas, we believe that the presence of larger patterns, such as resiniferous channels and vessels, may compromise the performance of this descriptor.

Regarding the number of species used for training, Fig. 5 shows that the performance of the dissimilarity-based classifier increases as the number of classes gets bigger. This curve was produced by the classifier trained with LPQ, but all the others mimic this behavior as well. This is somewhat different from what is seen in other works based on dissimilarity, where the classifier achieves the best results using few classes for training [2]. In our problem, this can be explained by the large intra-class variability.

7.2 Combination of 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 [22]. With that in mind, the second part of our experiments consisted in combining the 10 classifiers presented in the previous section. The reduced number of classifiers allows us to explore all possible combinations among them. Thus, Table 3 shows the best results, i.e., all the combinations that surpassed the best single classifier. Although we used eight different fusion rules, the results in Table 3 were achieved by using the Max rule.

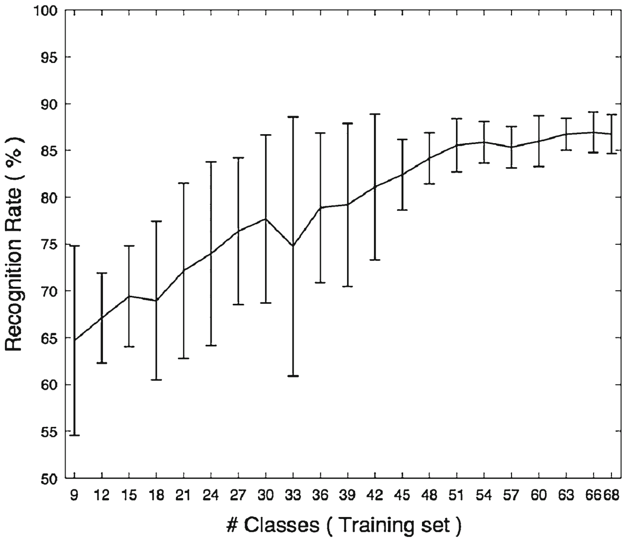


Fig. 5 Performance of the dissimilarity classifier as the number of classes available for training increases

From these results, we may notice a higher influence of the best classifiers. Except for Gabor filters, all of them were included in the pool of classifiers according to the sequence in Table 2. In general, the recognition rates for the different combinations were close to each other, and the maximum improvement seen from the combination of classifiers was 1.57 percentage points. This was achieved by combining SURF, MSER-SURF, SIFT, LPQ and LPQ-TOP classifiers. Figure 6 compares the ROC curves for the best single classifier (SURF) with the best ensemble produced by the combination.

The ultimate goal of combining classifiers is to increase the classification performance of the pattern recognition system. This scheme works well when the sets of patterns misclassified by different classifiers do not overlap. By analyzing the misclassified samples in our experiments, we noticed that the classifiers very often make the same mistakes. That is the reason why the combination brought only a slight improvement.

7.3 Dynamic selection of classifiers

When discussing DSC, the concept of oracle performance is usually present in the evaluation of the proposed methods.

Table 3 Best recognition rates among ensembles composed by k ($k = 1 \dots 10$) classifiers

Ensembles	%	σ
SURF	89.14	2.39
SURF and MSER-SURF	90.10	1.65
SURF, MSER-SURF and SIFT	90.51	2.38
SURF, MSER-SURF, SIFT and LPQ	90.57	2.26
SURF, MSER-SURF, SIFT, LPQ and LPQ-TOP	90.71	2.45
SURF, MSER-SURF, SIFT, LPQ, LPQ-TOP and Gabor	90.66	2.44
SURF, MSER-SURF, SIFT, LPQ, LPQ-TOP, LBP$_{u2}$ and Gabor	90.61	2.33
SURF, MSER-SURF, SIFT, LPQ, LPQ-TOP, LBP$_{u2}$, LBP$_{ri}$ and Gabor	90.48	2.34

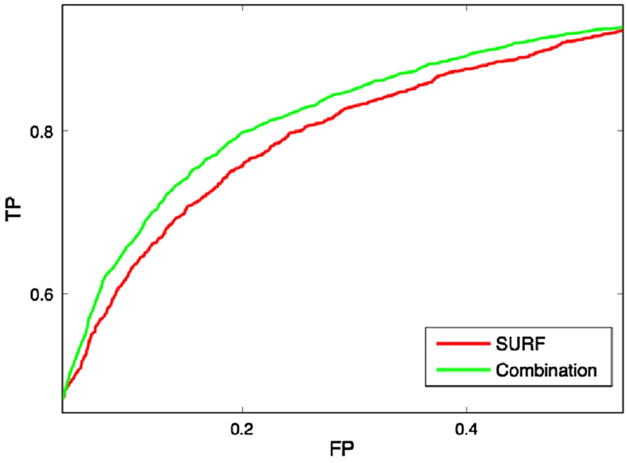


Fig. 6 ROC curves for the best single classifier and the best ensemble

This means that the proposed method is compared against the upper limit in terms of performance of the pool of classifiers. The oracle performance is estimated by considering that if at least one classifier can correctly classify a particular test sample, then the pool can do so as well. Considering our pool of 10 dissimilarity-based classifiers, the oracle is 99.54 %. The challenge is then to select the right classifier given a test sample.

As stated in Sect. 3, several DSC methods have been proposed in the literature to solve this problem. We have tested most of them and our best results were achieved by those methods that select dynamically a single classifier. Methods selecting ensemble of classifiers, such as kNORA [20], did not show good performance due the reduced number of classifiers in the pool.

In our first experiment we have tested the seminal methods proposed by Woods et al. in 1997 [43]. The best result using OLA, 86.91 % ($\sigma = 1.76$), was achieved using 300 neighbours ($k = 300$), i.e., the competence region of the classifier is composed of 300 samples. According to the literature, LCA in general produce better results than OLA. In our experiments, however, LCA reached a lower recognition rate, 82.96 % ($\sigma = 3.51$) using $k = 13$. In both cases, the number of neighbors ranged from 1 to 300 and the combination rule that produced the best results was the Max rule.

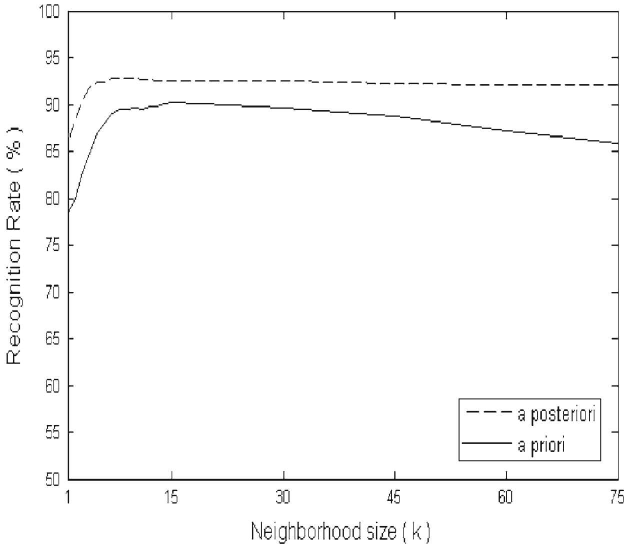


Fig. 7 Performance comparison between “A posteriori” and “A priori” for different neighborhood sizes

Table 4 Summary of the modified MCB methods

Method	Rec. rate (%)	σ	k
MCB OLA	84.36	2.52	4
MCB LCA	88.40	2.49	60
MCB “A priori”	87.20	2.11	195
MCB “A posteriori”	93.03	1.36	8

In the second experiment the probabilistic-based measures introduced by Giacinto and Roli [9] were evaluated. Here, instead of calculating the percentage of validation samples correctly classified in the input pattern neighborhood, we combined the a posteriori probabilities of input pattern neighbors in the validation set. The “a posteriori” method surpassed all the results presented so far, achieving a recognition rate of 92.86 % ($\sigma = 2.29$) for $k = 8$. The best result of the “a priori” technique was 90.28 % ($\sigma = 1.19$) for $k = 15$. Similarly to the previous experiments, the best fusion rule was the Max. Figure 7 compares the performance of both methods for different neighborhood sizes. It shows that the “a posteriori”

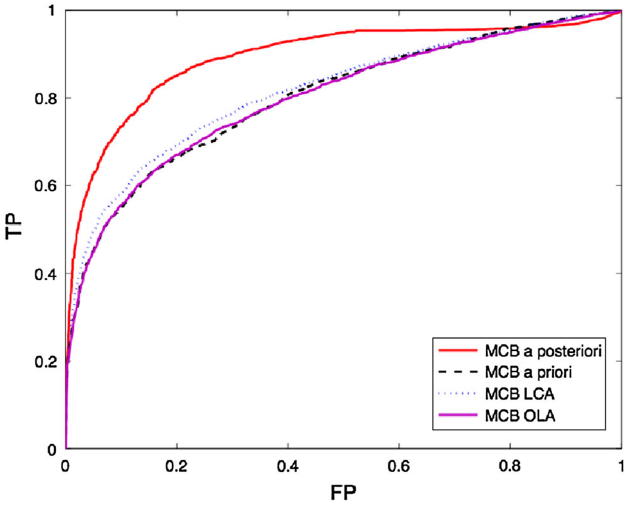


Fig. 8 ROC for the classifiers presented in Table 4

method is more robust to variations of the neighborhood size than the "a priori" method, which degrades as k gets bigger.

The third part of this experiment was devoted to assessing the Multiple Classifier Behavior (MCB) proposed in [10]. The original MCB, which uses OLA as an accuracy measure, performed worse than the original OLA method, achieving 84.36 % ($\sigma = 2.52$) for $k = 4$. Faced with this poor result we propose that the original MCB OLA be modified to use other competence measures, such as the LCA, "a posteriori" and "a priori". Compared to MCB OLA, MCB LCA, MCB "a priori" and MCB "a posteriori" increased the recognition rates in 4.04, 2.84, and 8.67 percentage points, respectively. Table 4 summarizes the performance of the modified MCB methods and Fig. 8 compares the ROC curves for these strategies.

Our results clearly show that the MBC "a posteriori" method can benefit from the probabilistic outputs produced by the classifiers. However, in spite of the compelling improvement brought by the dynamic selection, the Oracle (99.54 %) indicates that there is a lot of room for improvement.

In order to better understand the limits of the DSC methods, we analyzed how each method selects the classifiers from the pool. It is logical to expect the best classifiers to be selected more often. Although it does in fact happen, we can however observe from Table 5 that, in spite of the fact that we have 5 classifiers with similar performance (SURF,

Table 5 Classifiers selection rates according to DCS methods

Classifiers	DSC methods							
	A		*B*		*C*		*D*	
	%	σ	%	σ	%	σ	%	σ
LPQ	84.12	3.10	53.95	9.31	78.44	7.95	83.37	4.44
SIFT	12.65	2.15	25.79	4.04	12.55	7.02	12.33	4.37
SURF	2.27	0.52	14.56	7.01	3.74	1.12	2.10	0.26
MSER-SURF	0.67	0.35	2.93	0.77	1.11	0.51	1.07	0.44
LPQ-TOP	0.18	0.11	1.66	0.52	0.82	0.50	0.63	0.18
$\mathrm{LBP}_{8,2}^{u2}$	0.07	0.04	0.24	0.11	0.16	0.12	0.11	0.13
$\mathrm{LBP}_{8,2}^{ri}$	0.02	0.01	0.18	0.14	0.39	0.31	0.09	0.09
$\mathrm{LBP}_{8,2}^{riu2}$	0.00	0.00	0.19	0.19	1.92	1.04	0.07	0.08
GLCM	0.00	0.00	0.48	0.16	0.76	0.43	0.23	0.07
Gabor	0.00	0.00	0.02	0.01	0.11	0.11	0.00	0.00

A LCA, *B* a posteriori, *C* MCB LCA, *D* MCB a posteriori

Table 6 Summary of the results published in the literature using the microscopic images of forest species

References	Number of classes		Features/strategy	Recognition rates (%)
	Training	Testing		
Yadav et al. [44]	25	25	Gabor + GLCM	88.0–92.0
Yusof et al. [46]	52	52	Basic gray level aura matrix	89.0–93.0
Martins et al. [26]	112	112	LBP	80.7
Martins et al. [26]	112	112	LPQ + LBP	86.5
Cavalin et al. [6]	112	112	LPQ + GLCM	93.2
Kapp et al. [17]	112	112	LPQ−Blackman	95.0
Hafemann et al. [13]	112	112	Convolutional neural network	95.0
This work	68	44	DSC	93.0

MSER-SURF, SIFT, LPQ, and LPQ-TOP), the selection has strongly favors the LPQ-based classifier.

Reducing such a bias may be a solution in getting closer to the oracle. We have observed that on the few occasions where the other classifiers were selected, they mostly provided the correct answer. This should be subject of further investigation.

Table 6 summarizes the recent results published on the literature using microscopic images of forest species. Comparing our results with the literature is not straightforward since our method is the only one based on the dissimilarity approach that uses disjoint sets of classes for training and testing. In all other methods, the classes used for training are also used for testing. As stated previously, dissimilarity offers the advantage of not training the learning model each time a new class is presented to the system.

8 Conclusion

In this work, we have presented a framework based on dissimilarity feature vectors and DSC 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. The results reported in Sect. 7 indicate that these features surpass the traditional ones in performance.

Regarding the DSC, we have assessed several methods and observed that the methods proposed by Giacinto and Roli [9,10] seem to be most suitable for the pool of classifiers we have built. Our results also show that the modification we have made to the original MCB method brought an important improvement to the final recognition rate. Compared to the original MCB OLA (84.36 %), the proposed MCB "a Posteriori" was about 8 percentage points more accurate, achieving a recognition rate of 93.03 %.

In spite of the improvement produced by the DSC when compared to the single classifiers and also when compared to the combination of classifier, it is clear that there is a lot of room for improvement. Considering that the oracle points out to a recognition rate of more than 99 %, the challenge is how to select the good classifier for a given input pattern. The methods presented in this paper represent step closer to the oracle, but certainly, more research should be done to close this gap. For future works, we also plan to investigate automatic methods of generating pools of classifiers in the dissimilarity space.

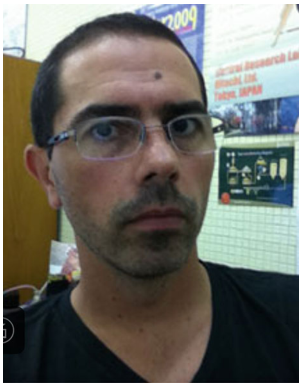
Acknowledgments This research has been supported by The National Council for Scientific and Technological Development (CNPq) grants 301653/2011-9 and 471050/2013-0 and the Coordination for the Improvement of Higher Education Personnel (CAPES).

References

1. Bay, H., Ess, A., Tuytelaars, T., Van Gool, L.: Speeded-up robust features (surf). Comput. Vis. Image Underst. **110**(3), 346–359 (2008)
2. Bertolini, D., Oliveira, L.S., Justino, E., Sabourin, R.: Reducing forgeries in writer-independent off-line signature verification through ensemble of classifiers. Pattern Recognit. **43**, 387–396 (2010)
3. Bertolini, D., Oliveira, L.S., Justino, E., Sabourin, R.: Texture-based descriptors for writer identification and verification. Expert Syst. Appl. **40**(6), 2069–2080 (2013)
4. Bianconi, F., Ceccarelli, L., Fernádez, A., Saetta, S.A.: A sequential machine vision procedure for assessing paper impurities. Comput. Ind. **65**, 325–332 (2014)
5. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)
6. Cavalin, P.R., Kapp, M.N., Martins, J., Oliveira, L.S.: A multiple feature vector framework for forest species recognition. In: 28th Annual ACM Symposium on Applied Computing, pp. 16–20, New York, NY, USA (2013)
7. Cavalin, P.R., Sabourin, R., Suen, C.Y.: Dynamic selection approaches for multiple classifier systems. Neural Comput. Appl. **22**(3–4), 673–688 (2013)
8. Chen, J., Shan, S., He, C., Zhao, G., Pietikäinen, M., Chen, X., Gao, W.: Wld: a robust local image descriptor. IEEE Trans. Pattern Anal. Mach. Intell. **32**(9), 1705–1720 (2010)
9. Giacinto, G., Roli, F.: Methods for dynamic classifier selection. In: International Conference on Image Analysis and Processing, pp. 659–665. IEEE Computer Society, Washington, DC (1999)
10. Giacinto, G., Roli, F.: Dynamic classifier selection based on multiple classifier behaviour. Pattern Recognit. **34**, 1879–1881 (2001)
11. Goldfarb, L.: What is distance and why do we need the metric model for pattern learning? Pattern Recognit. **25**(4), 431–438 (1992)
12. Gonzalez, R.C., Woods, R.E.: Digital Image Processing, 3rd edn. Prentice Hall, New Jersey (2008)
13. Haffemann, L.G., Oliveira, L.S., Cavalin, P.R.: Forest species recognition using deep convolutional neural networks. In: International Conference on Pattern Recognition, pp. 1103–1107 (2014)
14. Haralick, R.M.: Statistical and structural approaches to texture. IEEE **67**(5), 786–804 (1979)
15. Ho, T.K.: The random subspace method for constructing decision forests. IEEE Trans. Pattern Anal. Mach. Intell. **20**(8), 832–844 (1998)
16. Britto Jr, A.S., Sabourin, R., Oliveira, L.S.: Dynamic selection of classifiers—a comprehensive review. Pattern Recognit. **47**(11), 3665–3680 (2014)
17. Kapp, M., Bloot, R., Cavalin, P.R., Oliveira, L.S.: Automatic forest species recognition based on multiple feature sets. In: International Joint Conference ou Neural Networks, pp. 1296–1303 (2012)
18. Khalid, M., Lee, E.L.Y., Yusof, R., Nadaraj, M.: Design of an intelligent wood species recognition system. Int. J. Simul. Syst. Sci. Technol. **3**(9), 9–17 (2008). Special Issue on: Artificial Intelligence
19. Kittler, J., Hatef, M., Duin, R.P.W., Matas, J.: On combining classifiers. IEEE Trans. Pattern Anal. Mach. Intell. **20**(3), 226–239 (1998)
20. Ko, A.H.R., Sabourin, R., Britto Jr, A.S.: From dynamic classifier selection to dynamic ensemble selection. Pattern Recognit. **41**(5), 1718–1731 (2008)
21. Kumar, R., Sharma, J.D., Chanda, B.: Writer-independent off-line signature verification using surroundedness feature. Pattern Recognit. Lett. **33**(3), 301–308 (2012)
22. Kuncheva, L.I., Rodriguez, J.J.: Classifier ensemble with a random linear oracle. IEEE Trans. Knowl. Data Eng. **19**(4), 500–508 (2007)

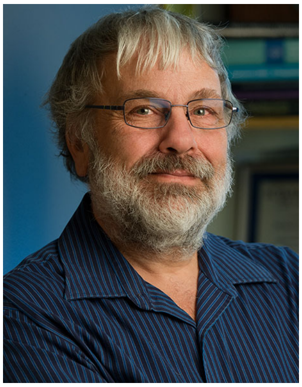
23. Lavine, B.K., Davidson, C.E., Moores, A.J., Griffiths, P.R.: Raman spectroscopy and genetic algorithms for the classification of wood types. Appl. Spectrosc. **55**(8), 960–966 (2001)
24. Lowe, D.G.: Object recognition from local scale-invariant features. Int. Conf. Comput. Vis. **2**, 1150–1157 (1999)
25. Martins, J.G., Oliveira, L.S., Nisgoski, S., Sabourin, R.: A database for automatic classification of forest species. Mach. Vis. Appl. **24**(3), 567–578 (2013)
26. Martins, J.G., Oliveira, L.S., Sabourin, R.: Combining textural descriptors for forest species recognition. In: 38th Annual Conference on IEEE Industrial Electronics Society, pp. 1483–1488 (2012)
27. Matas, J., Chum, O., Martin, U., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: Proceedings of the British Machine Vision Conference, vol. 1, pp. 384–393 (2002)
28. Nuopponen, M.H., Birch, G.M., Sykes, R.J., Lee, S.J., Stewart, D.: Estimation of wood density and chemical composition by means of diffuse reflectance mid-infrared fourier transform (drift-mir) spectroscopy. J. Agric. Food Chem. **54**(1), 34–40 (2006)
29. Ojala, T., Pietikäinen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. Pattern Recognit. **29**(1), 51–59 (1996)
30. Ojala, T., Pietikäinen, M., Mäenpää, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. **24**(7), 971–987 (2002)
31. Ojansivu, V., Heikkilä, J.: Blur insensitive texture classification using local phase quantization. In: 3rd International Conference on Image and Signal Processing, pp. 236–243 (2008)
32. Paivarinta, J., Rahtu, E., Heikkila, J.: Volume local phase quantization for blur-insensitive dynamic texture classification. In: 17th Scandinavian Conference on Image analysis, pp. 360–369 (2011)
33. Paula Filho, P.L., Oliveira, L.S., Britto Jr, A.S., Sabourin, R.: Forest species recognition using color-based features. In: 20th International Conference on Pattern Recognition (ICPR2010), pp. 4178–4181 (2010)
34. Paula Filho, P.L., Oliveira, L.S., Nisgoski, S., Britto Jr. A.S.: Forest species recognition using macroscopic image. Mach. Vis. Appl. **25**, 1019–1031 (2014)
35. Pekalska, E., Duin, R.P.W.: Dissimilarity representations allow for building good classifiers. Pattern Recognit. Lett. **23**(8), 943–956 (2002)
36. Piuri, V., Scotti, F.: Design of an automatic wood types classification system by using fluorescence spectra. IEEE Trans. Syst. Man Cybern. Part C **40**(3), 358–366 (2010)
37. Platt, J.: Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In: Smola, A., et al. (eds.) Advances in Large Margin Classifiers, pp. 61–74. MIT Press, Cambridge (1999)
38. Santini, S., Jain, R.: Similarity measures. IEEE Trans. Pattern Anal. Mach. Intell. **21**(9), 871–883 (1999)
39. Schapire, R.E., Freund, Y., Bartlett, P., Lee, W.S.: Boosting the margin: a new explanation for the effectiveness of voting methods. In: 14th International Conference on Machine Learning, pp. 322–330 (1997)
40. Tou, J.Y., Tay, Y.H., Lau, P.Y.: One-dimensional grey-level co-occurrence matrices for texture classification. In: International Symposium on Information Technology, pp. 1–6 (2008)
41. Tou, J.Y., Tay, Y.H., Lau, P.Y.: A comparative study for texture classification techniques on wood species recognition problem. In: International Conference on Natural Computation, pp. 8–12 (2009)
42. Tsuchikawa, S., Hirashima, Y., Sasaki, Y., Ando, K.: Near-infrared spectroscopic study of the physical and mechanical properties of wood with meso- and micro-scale anatomical observation. Appl. Spectrosc. **59**(1), 86–93 (2005)
43. Woods, K., Kegelmeyer Jr, W.P., Bowyer, K.: Combination of multiple classifiers using local accuracy estimates. IEEE Trans. Pattern Anal. Mach. Intell. **19**(4), 405–410 (1997)
44. Yadav, A.R., Dewal, M.L., Anand, R.S., Gupta, S.: Classification of hardwood species using ann classifier. In: National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics, pp. 1–5 (2013)
45. Yanikoglu, B., Aptoula, E., Tirkaz, C.: Automatic plant identification from photographs. Mach. Vis. Appl. **25**, 1369–1383 (2014)
46. Yusof, R., Khalid, M., Khairuddin, A.S.M.: Fuzzy logic-based preclassifier for tropical wood species recognition system. Mach. Vis. Appl. **24**(8), 1589–1604 (2013)
47. Zhao, G., Pietikäinen, M.: Dynamic texture recognition using local binary patterns with an application to facial expressions. IEEE Trans. Pattern Anal. Mach. Intell. **29**(6), 915–928 (2007)

J. G. Martins received his B.S. degree in Informatics from the Western Paraná State University (UNIOESTE), 2000, Cascavel, PR, Brazil, M.E. degree in Production Engineering from the Federal University of Santa Catarina (UFSC), 2002, Florianópolis, SC, Brazil and the Doctoral degree from the Federal University of Parana in 2014. In 2003 he joined the Technollogical Federal University of Parana (UTFPR) where he is currently an assistant professor. His research interests include Pattern Recognition and Image Analysis.



L. S. Oliveira received his B.S. degree in Computer Science from UnicenP, Curitiba, PR, Brazil, the M.Sc. degree in electrical engineering and industrial informatics from the Centro Federal de Educacão Tecnologica do Paraná (CEFET-PR), Curitiba, PR, Brazil, and Ph.D. degree in Computer Science from Ecole de Technologie Superieure, Universite du Quebec in 1995, 1998 and 2003, respectively. From 2004 to 2009 he was professor of the Computer Science Department at Pontifical Catholic University of Parana, Curitiba, PR, Brazil. In 2009, he joined the Federal University of Parana, Curitiba, PR, Brazil, where he is professor of the Department of Informatics and head of the Graduate Program in Computer Science. His current interests include Pattern Recognition, Machine Learning, Image Analysis, and Evolutionary Computation.

A. S. Britto Jr. received M.Sc. degree in Industrial Informatics from the Centro Federal de Educação Tecnológica do Paraná (CEFET-PR, Brazil) in 1996, and Ph.D. degree in Computer Science from the Pontifícia Universidade Católica do Paraná (PUCPR, Brazil) in 2001. In 1989, he joined the Informatics Department of the Universidade Estadual de Ponta Grossa (UEPG, Brazil). In 1995, he also joined the Computer Science Department of the Pontifícia Universidade Católica do Paraná (PUCPR) and, in 2001, the Postgraduate Program in Informatics (PPGIa). His research interests are in the areas of Pattern Recognition, Machine Learning and Computer Vision.



R. Sabourin received his B.Ing., M.Sc.A., and Ph.D. degrees in Electrical Engineering from the Ecole Polytechnique de Montreal in 1977, 1980 and 1991, respectively. In 1977, he joined the physics department of the Universite de Montreal where he was responsible for the design and development of scientific instrumentation for the Observatoire du Mont Megantic. In 1983, he joined the staff of the Ecole de Technologie Superieure, Universite du Quebec, Montreal, P.Q. Canada, where he is currently a professeur titulaire in the Departement de Genie de la Production Automatisee. In 1995, he joined also the Computer Science Department of the Pontificia Universidade Catolica do Parana (PUC-PR, Curitiba, Brazil) where he was corresponsible since 1998 for the implementation of a Ph.D. program in Applied Informatics. Since 1996, he is a senior member of the Centre for Pattern Recognition and Machine Intelligence (CENPARMI). His research interests are in the areas of handwriting recognition and signature verification for banking and postal applications.