A Multiple Feature Vector Framework for Forest Species Recognition

Paulo R. Cavalin
Universidade Federal do Tocantins - UFT
Quadra 109N Av. NS15 s/n
Bala II sala 20
Palmas (TO), Brazil
cavalin@uft.edu.br

Jefferson Martins
Universidade Tecnológica Federal do Paraná - UTFPR
Rua Cristo Rei, 10
Toledo (PR), Brazil
martins@utfpr.edu.br

Marcelo N. Kapp
Universidade Federal da Integração Latino-Americana - Unila
Av. Tancredo Neves, 6731 - Bloco 4
Foz do Iguaçú (PR), Brazil
marcelo.kapp@unila.edu.br

Luiz E. S. Oliveira
Universidade Federal do Paraná - UFPR
Centro Politécnico - Jardim das Américas
Curitiba (PR), Brazil
lesoliveira@inf.ufpr.br

ABSTRACT

In this work we focus on investigating the use of multiple feature vectors for forest species recognition. As consequence, we propose a framework to deal with the extraction of multiple feature vectors based on two approaches: image segmentation and multiple feature sets. Experiments conducted on a 112 species database containing microscopic images of wood demonstrate that with the proposed framework we can increase the recognition rates of the system from about 55.7% (with a single feature vector) to about 93.2%.

Categories and Subject Descriptors

I.5 [**Pattern Recognition**]: Miscellaneous; I.5.2 [**Pattern Recognition**]: Design Methodologies—*Classifier design and evaluation, Pattern analysis*; I.5.4 [**Pattern Recognition**]: Applications—*Computer vision, Signal processing*

General Terms

Forest species recognition

Keywords

Forest Species Recognition, Classifier Combination, Multiple Feature Vectors, Quadtree

1. INTRODUCTION

Automatic recognition of forest species has been drawing the attention of the machine learning community, given its both commercial and environment-preserving value. For example, by better monitoring wood timber trading, one may reduce commercialization of samples from species that are forbidden to be traded, e.g. species near extinction. But identifying forest species only from wood lumbers is a task that generally requires an expert. In the light of this, an automatic system is a low-cost alternative to deal with this issue, since we could reduce the costs for hiring and training human experts. For this reason, recently various systems have been proposed to cope with this problem [1, 3, 5, 6, 9].

An interesting remark in this literature is that the use of multiple feature vectors is promising to reduce recognition error. As reported in [9], by dividing the input image into N sub-images and extracting N distinct feature vectors, we can classify each vector individually then combine their classification outputs to compute the final result. This approach can outperform a baseline system considering only one feature vector extracted from the original image. In [5], the use of multiple feature vectors is considered as well. In this case, though, each feature vector is associated to a different feature set. Boost in performance can also be observed with this approach.

Given these standpoints, in this paper we propose the Multiple Feature Vector framework, which defines an algorithm to extract complementary feature vectors from an input image and combines their classification results to compute the final recognition decision. This framework considers that multiple feature vectors can be extracted by both segmenting the input image as in [9] and by considering different feature sets as in [5]. It is worth noting that the framework can be extended to include other multiple feature vector approaches, so that it can be used to evaluate and define the best multiple feature vector structure for other closely related pattern recognition problems. In this work, the framework is evaluated on forest species recognition by

SAC'13 March 18-22, 2013, Coimbra, Portugal.

using a quadtree-based approach to divide the original image, 3 feature sets (Gray Level Co-occurrence Matrix, Linear Binary Patterns, and Linear Phase Quantization), and four common combination rules (Sum, Product, Max, and Majority Voting rules). Experiments conducted on the database proposed in [6], which contains microscopic images from 112 different species of wood, show that the recognition rates can increase from 55.7%, extracting a single GLCM feature vector, to 93.2%, extracting both LPQ and GLCM features and dividing the image into 16 segments.

Greater detail of the proposed methodology and the experiments are described in Section 2 and Section 3, respectively. Then, conclusions and future work are discussed in Section 4.

2. METHODOLOGY

In this section we describe in greater detail the proposed method for forest species recognition using multiple feature vectors. We first describe a generic framework that, given a parameter for defining the number of segments, a collection of feature sets, and classifiers trained with these feature sets, conducts the recognition of an input sample using multiple feature vectors which are then combined with a given fusion function. Afterwards, we describe the methods considered in this work to implement and evaluate the framework, such as a quadtree-based approach to divide the image, three feature sets, and four well-known combination functions.

2.1 The Multiple Feature Vector Framework

The main idea of this framework consists of using information from multiple feature vectors to improve recognition performance, owing to the variability introduced by these multiple vectors. In this work, we propose to extract varied feature vectors by both dividing the input image into segments, and by combining different feature sets.

Algorithm 1 The main steps of the multiple feature vector framework.

1: **Input:** im, the input image; L, the parameter to compute the number of segments to divide im; $\Xi = \{\xi^1, \ldots, \xi^M\}$, the collection of feature sets, where ξ^j corresponds to a distinct feature set; $C_L = \{c_L^1, \ldots, c_L^M\}$, the set of classifiers trained for the given L, for each ξ^j; K, the number of classes of the problem; and λ, a fusion function to combine multiple classification results.
2: $N = f(L)$
3: Divide im into N non-overlapping sub-images with equal size, generating the set $I = \{im'_1, \ldots, im'_N\}$
4: **for** each image im'_i in I **do**
5: **for** each feature set ξ^j in Ξ **do**
6: Extract feature vector v_i^j from im_i by considering ξ^j as feature set, and save v_i^j in V
7: **end for**
8: **end for**
9: **for** each feature vector v_i^j in V **do**
10: Recognize v_i^j using classifier c_L^j, and save the scores $p_i^j(k)$ for each class k, where $1 \leq k \leq K$.
11: **end for**
12: Combine all scores $p_i^j(k)$ using λ, and compute the final recognition decision ϕ.

The main steps of the framework are listed in Algorithm 1. The main inputs for this algorithm are: the original image, denoted im; the parameter L, to define the number of segments in which im will be divided; the set of feature sets $\Xi = \{\xi^1, \ldots, \xi^M\}$, where ξ^j represents a distinct feature set with m^j features, and $M = |\Xi|$ corresponds to the total number of feature sets; the set of classifiers $C_L = \{c_L^1, \ldots, c_L^M\}$, corresponding to the set of classifiers trained for a given value of L for each ξ^j; the number of classes of the problem denoted K; and a fusion function λ. From these inputs, the first steps consist in dividing the input image into N segments. That is, in step 2, the number of segments N is computed as a function of L, e.g. $N = f(L)$. Then, in step 3, im is divided into N non-overlapping segments with identical sizes, generating the set $I = \{im'_1, \ldots, im'_N\}$. Next, the feature extraction is carried out. In steps 4 to 8, for each image im'_i in I and each feature set ξ^j in Ξ, the feature vector v_i^j is extracted and saved in V. Afterwards, each feature vector v_i^j in V is classified by the corresponding classifier c_L^j, resulting in the scores $p_i^j(k)$ for every class k, where $1 \leq k \leq K$. Finally, all the scores $p_i^j(k)$ are combined using the combination function λ and the final recognition decision ϕ is made, i.e. the forest species (a class k) from which im has been extracted is outputted.

2.2 Implementation Details

In this section, we describe the methods used to implement and evaluate the framework described in the previous section. First, we describe a quadtree-based approach to divide the input image. Next, we provide the feature sets and the combination functions considered in this work.

2.2.1 Quadtrees for Image Segmentation

A quadtree consists of a successive structure to divide an input image into four segments, according to the number of levels of the three [10]. That is, at each level L, where $L \geq 1$, the input image, denoted im, is divided into $(2^{L-1})^2$ non-overlapping sub-images of equal sizes. This means that at level 1 im remains as the original image, at level 2 it is divided in 4 images, at level 3 in 16 images, and so on. This approach is illustrated in Figure 1(a).

The relationship between an image at level L and the sub-images at level $L+1$ is depicted in Figure 1(b). For example, at level 2, images im'_1, im'_2, im'_3, and im'_4, correspond to image im from level 1 divided in four segments. At level 3, images im''_1, im''_2, im''_5, and im''_6, correspond to image im'_1 from level 2, divided by four. This approach can be recursively applied to the images from level 3 to level 4, and so on. As a result, a quadtree allows for evaluating a given image at different granularity levels, resulting in a hierarchical approach for image segmentation. In this work, nonetheless, such relationships are not important since we are just interested in the exponential function used by quadtrees to divide images. As a consequence, the parameter L of Algorithm 1 is used to compute the number of segments (step 2) according to the corresponding level L of the quadtree, e.g. $N = f(L) = (2^{L-1})^2$.

2.2.2 Feature Sets

We consider three distinct feature sets: Gray Level Co-occurrence Matrix, Local Binary Patterns and Linear Phase Quantization. While GLCM is a widely used statistical technique for texture recognition, LBP and LPQ are currently among the best feature sets for such a task. In addition, we

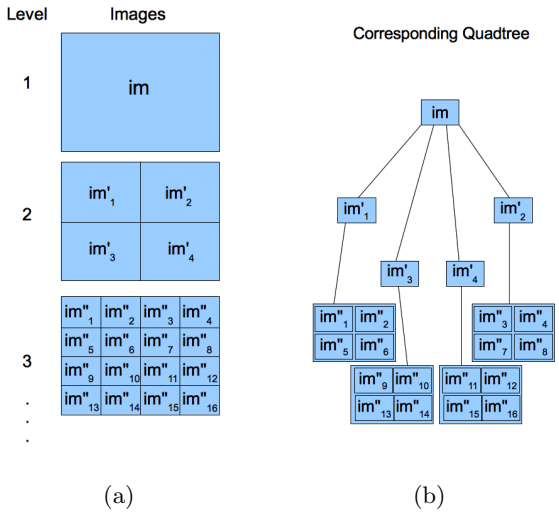


Figure 1: Illustration of the Quadtree approach. a) an image divided with three levels of a quadtree. b) the corresponding quadtree and the relationship between the images.

might consider LPQ as better than LBP for texture recognition given recent evaluations [8]. As a consequence, we believe that these three feature sets are diverse enough to be used in the framework.

Gray Level Co-occurrence Matrix - GLCM.

This technique consists of statistical experiments conducted on how a certain level of gray occurs on other levels of gray. It intuitively provides measures of properties such as smoothness, coarseness, and regularity. Haralick [2], who originated this technique, suggested a set of 14 characteristics, but most works in the literature consider a subset of these descriptors. In our case, we used the following six: Energy, Contrast, Entropy, Homogeneity, Maximum Likelihood, and 3rd Order Moment.

By definition, a GLCM is the probability of the joint occurrence of gray-levels 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 this GLCM, some information can be extracted. Assuming that G is the gray-level depth, and $co(i, j)$ is the probability of the co-occurrence of gray-level i and gray-level j observing consecutive pixels at distance d and angle θ, we can use a GLCM to describe wood texture.

In our experiments, we tried different values of d, as well as different angles. The best setup we found is $d = 5$ and $\theta = \{0, 45, 90, 135\}$. Considering the six descriptors mentioned above, we arrive at a feature vector with 24 components.

Local Binary Patterns - LBP.

We consider the original LBP proposed by Ojala et al. [7]. In this case, the pixels of an image are labelled 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, this histogram can be used as a texture descriptor. Figure 2 illustrates this process.

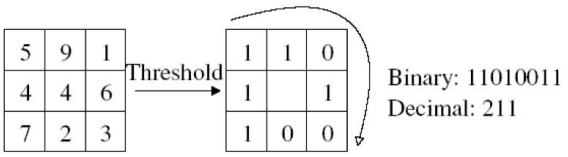


Figure 2: The original LBP operator.

The LBP operator produces $2^8 = 256$ different binary patterns that can be formed by the pixels in the neighborhood set. However, as stated in [7], uniform patterns, which can be represented by 58 binary patterns, might account for nearly 90% of all patterns in the neighborhood in texture images. Note that an binary pattern is called uniform if it contains at most two 0/1 transitions when the binary string is considered circular.

Given that uniform binary patterns can represent a reasonable quantity of information by a reduced set of binary patterns, here we implement an LBP feature extractor by considering a 59-position normalized histogram. While each of the first 58 positions take into account the occurrence of an uniform pattern, the 59-th position represents the occurrence of all non-uniform ones.

Linear Phase Quantization - LPQ.

Proposed by Ojansivu et Heikkilä [8], 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 $i(x)$ defined by Equation 1:

$$F(u, x) = \sum_{y \in N_x} i(x - y) e^{-2\pi j u^T y} = w_u^T f_x \qquad (1)$$

where w_u is the basis vector of the 2-D DFT at frequency u, and f_x is another vector containing all M^2 image samples from N_x.

The STFT can be implemented using a 2-D convolution $f(x)e^{-2\pi j u^T x}$ for all u. In LPQ only four complex coefficients are considered, corresponding to 2-D frequencies $u_1 = [a, 0]^T$, $u_2 = [0, a]^T$, $u_3 = [a, a]^T$, and $u_4 = [a, -a]^T$, where a is a scalar frequency below the first zero crossing of the DFT $H(u)$. $H(u)$ is DFT of the point spread function of the blur, and u is a vector of coordinates $[u, v]^T$. More details of the formal definition of LPQ can be found in [8], where those authors introduced all mathematical formalism.

At the end, we will have a 8-position resulting vector G_x for each pixel in the original image. These vectors G_x are quantized using a simple scalar quantizer (see Equation 2 and Equation 3), where g_j is the j-th components of G_x.

$$q_j = \begin{cases} 1, & \text{if } g_j \geq 0 \\ 0, & \text{otherwise.} \end{cases} \qquad (2)$$

$$b = \sum_{j=1}^{8} q_j 2^{j-1} \qquad (3)$$

The quantized coefficients are represented as integer values between 0-255 using binary coding (Equantion 3). These binary codes will be generated and accumulated in a 256-histogram, similar to the LBP method. The accumulated

values in the histogram will be used as the LPQ 256-dimensional feature vector.

2.2.3 Combination Functions

Let $p_i^j(k)$ be the score of class k outputted by classifier c_L^j given the feature vector v_i^j, we consider the following combination functions to compute the final decision ϕ.

1. **Sum rule**:

$$\phi = \arg\max_{k=1}^{K} \sum_{i=1}^{N} \sum_{j=1}^{M} p_i^j(k) \quad (4)$$

2. **Product rule**

$$\phi = \arg\max_{k=1}^{K} \prod_{i=1}^{N} \prod_{j=1}^{M} p_i^j(k) \quad (5)$$

3. **Max rule**

$$\phi = \arg\max_{k=1}^{K} \max_{i=1}^{N} \max_{j=1}^{M} p_i^j(k) \quad (6)$$

4. **Majority Voting**

$$\Delta_i^j(k) = \begin{cases} 1, & \text{if } p_i^j(k) = \max_{k=1}^{K} p_i^j(k) \\ 0, & \text{otherwise.} \end{cases} \quad (7)$$

$$\phi = \arg\max_{k=1}^{K} \sum_{i=1}^{N} \sum_{j=1}^{M} \Delta_i^j(k) \quad (8)$$

See Kittler et al. [4] for more details.

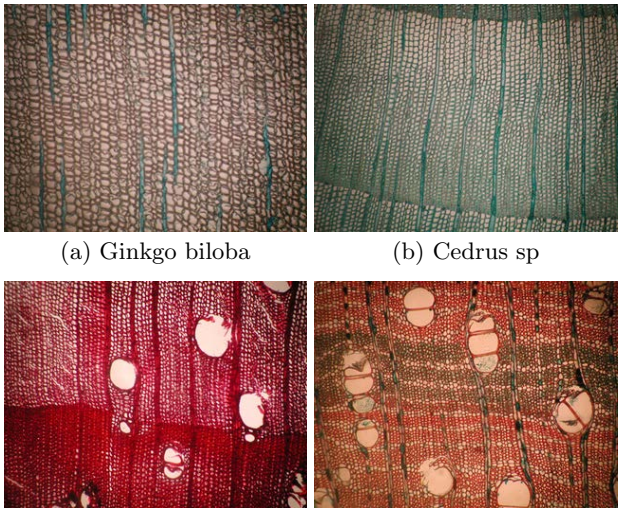
3. EXPERIMENTS

In order to evaluate the framework described in Section 2.1, we used the database presented in [6]. This database contains 2,240 microscopic images equally distributed in 112 distinct classes of forest species. From these, 37 species consist of Softwood and 75 consist of Hardwood species, which generally present different texture structures. See in Figure 3 some examples of the different texture pattern that can be found in this database.

For these experiments, the 20 samples of each class have been partitioned in: 10 samples for training; 4 samples for validation; and 6 samples for test. Each subset has been randomly selected, with no overlapping between the sets. For avoiding the results to be biased to a given partitioning, this scheme is repeated 10 times. As a consequence, the results presented further represent the average recognition rate over 10 replications (each replication is related to different a partitioning).

As the base classifier, we use Support Vector Machines (SVMs) with Gaussian kernel[1]. Parameters C and γ were optimized by means of a grid search with hold-out validation, using the training set to train SVM parameters and the validation set to evaluate the performance. After finding the best values for C and γ, an SVM was trained with both the training and validation sets together. Note that normalization was performed by linearly scaling each attribute to the range [-1,+1].

[1]in this work we used the LibSVM tool available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/.



(a) Ginkgo biloba (b) Cedrus sp

(c) Anadenanthera colubrina (d) Cariniana estrellensis

Figure 3: Some samples from the database. In a) and b) we present samples from Softwood species, and in c) and d) we present Hardwood samples.

In Figure 4 we present the results of the evaluation of the proposed Multiple Feature Vector framework, considering each feature set individually (note that, due to space constraints, the results presented hereafter consider only the Product combination rule, since this function achieved the best results in these experiments). The individual evaluation of these feature sets indicate that LPQ is the best feature set for this problem, followed by LBP and GLCM, reaching 91.5%, 88.5% and 80.7%, respectively. These results also demonstrate that segmenting the input image is effective in achieving better recognition rates. Considering the three values that we evaluated for L, i.e. 1, 2, and 3, we observe a gain of about 10.4 percentage points in the system using LPQ. The recognition rates increase from 81.1% with $L = 1$ (when the original image is used) to 91.5% with $L = 3$ (when the image is divided into 16 segments). With the other feature sets, such a gain is ever greater. With LBP there is an increase of about 13.7 percentage points, and with GLCM the gain is of about 25 percentage points. This indicates that, the weaker the feature set the more evident the impact of the segmentation-based multiple feature vector approach.

The results from the combination of two or more feature sets are presented in Figure 5. Compared with the best recognition rate of 91.5% achieved with a single feature set (LPQ features with L set to 3), the combination of multiple features sets and the segmentation of the input image can achieve 93.2%, with LPQ and GLCM features combined and $L = 3$. This demonstrates the effectiveness of the proposed framework, since better performance is achieved by both combining more than one feature set and segmenting the image (into 16 segments in this case). Nonetheless, we observe that such effectiveness depends on the choice of the feature sets. That is, higher recognition rates can be achieved by the combination of GLCM with LBP or LPQ than combining LBP with LPQ (these feature sets present the best results in Figure 4). This indicates that the framework takes

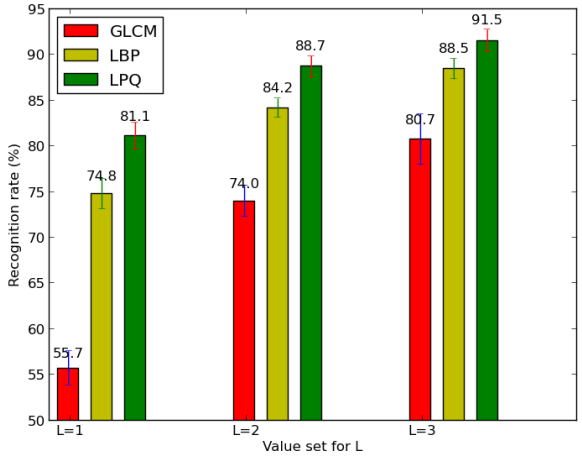


Figure 4: Evaluation of the proposed framework with only one feature set, using the Product rule for combination.

better advantage of diversity when a weaker feature set, e.g. GLCM, is combined with a stronger one, e.g. LBP and LPQ, since they are more likely to provide complementary results.

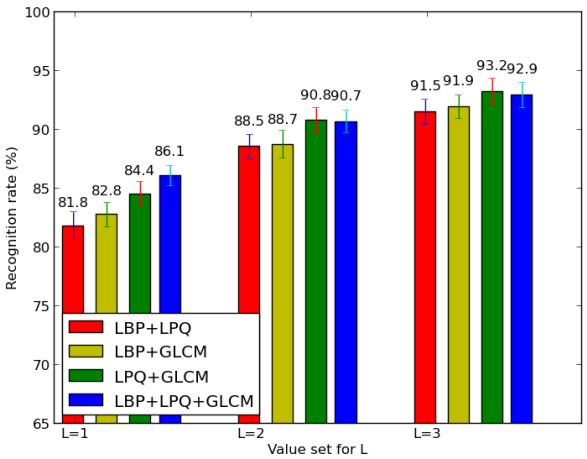


Figure 5: Evaluation of the proposed framework with more than one feature set, using the Product rule for combination.

Compared with the literature, it is worth mentioning that the recognition rates of 93.2%, achieved by the framework with LPQ and GLCM features combined and $L = 3$, are the best so far on this database. The best recognition rates previously reported were about 86.47% [5].

4. CONCLUSION AND FUTURE WORK

In this paper we formalized the Multiple Feature Vector framework for forest species recognition. Such a framework allows the combination of a set of feature vectors extracted by taking into account two different approaches: image segmentation and multiple feature sets.

Experiments conducted on a 112 species database demonstrated that it is possible to design a framework that outperforms a baseline, single feature-vector, system. In this case, the use of multiple feature vectors from both segmenting the input image and combining more than one feature set resulted in the best results so far in this database.

As future work, it might be interesting the investigation of other schemes to segment the images. Another direction might be the dynamic selection of feature vectors, so that only the most relevant ones are used for the final decision. In addition, the quadtree-based approach can be employed as a hierarchical approach to combine multiple feature vectors.

5. REFERENCES

[1] ArunPriya C. and A. S. Thanamani. A survey on species recognition system for plant classification. *International Journal of Computer Technology & Applications*, 3(3):1132–1136, 2012.

[2] R. M. Haralick. Statistical and structural approach to texture. *Proceedings of IEEE*, 65(5), 1979.

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

[4] J. Kittler, M. Hatef, R. P. Duin, and J. Matas. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998.

[5] J. Martins, L. S. Oliveira, and R. Sabourin. Combining textural descriptors for forest species recognition. In *38th Annual Conference of the IEEE Industrial Electronics Society (IECON 2012)*, 2012.

[6] J. Martins, L. S. Oliveira, and R. Sabourin. A database for automatic classification of forest species. *Machine Vision and Applications*, 2012.

[7] T. Ojala, M. Pietikãinen, and T. Mãenpää. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):971–987, 2002.

[8] V. Ojansivu and J. Heikkilä. Blunr insensitive texture classification using local phase quantization. In *Proceedings of the 3rd International Conference on Image and Signal Processing (ICISP '08)*, pages 236–243, 2008.

[9] P. L. Paula Filho, L. E. S. Oliveira, A. S. Britto, and R. Sabourin. Forest species recognition using color-based features. In *Proceedings of the 20th International Conference on Pattern Recognition*, pages 4178–4181, Instanbul, Turkey, 2010.

[10] G. Vamvakas, B. Gatos, and S. J. Perantonis. Handwritten character recognition through two-stage foreground sub-sampling. *Pattern Recognition*, 43(8):2807–2816, 2010.