X ENCONTRO ACADÊMICO
MODELAGEM COMPUTACIONAL

Ranking Texture Features Through AdaBoost.M2 Linear Ensembles for Granite Tiles Classification

Tiene Andre Filisbino

Gilson Antonio Giraldi

tiene@lncc.br

gilson@lncc.br

Laboratório Nacional de Computação Científica

Getúlio Vargas Av., 333, Quitandinha, CEP 25651-075, RJ, Petrópolis, Brasil

Carlos Eduardo Thomaz

cet@fei.edu.br

Department of Electrical Engineering, FEI

São Bernardo do Campo, 09850-901, SP, Brasil

Bruno Montandon Noronha Barros

bbarros@cetem.gov.br

CETEM/MCTIC

Cidade Universitária, 21941-908, RJ, Rio de Janeiro, Brasil

Millena Basílio da Silva

mbasilio@cetem.gov.br

CETEM-ES/MCTIC

Morro Grande, 29.300-970, Cachoeiro de Itapemirim, ES, Brasil

Abstract. *Texture analysis is a field of image processing that has been extensively applied for object recognition. In the specific case of automatic classification of granite tiles, the image acquisition can generate large resolution data. So, we need to construct efficient feature spaces in order to reduce the computational cost of further steps. The co-occurrence matrix and Haralick features are techniques to compute the feature vectors to represent the image samples. These methodologies are classical in pattern recognition and texture image analysis. In this paper, we apply the Multi-Class.M2 DPCA methodology for ranking texture features for granite tiles recognition. Hence, given a multi-class granite image database, we build multiple linear learners that are combined through an ensemble technique, the AdaBoost.M2, in order to determine the discriminant contribution of each feature. We implement the linear learners using the support vector machine (SVM). The strong learner built by the ensemble technique is processed following a strategy to get the global discriminant vector to sort texture features according to their relevance for classification tasks. In the computational experiments we analyse the obtained approach using a five-class granite image database. Our experimental results have shown that the features selected by the proposed technique allow competitive recognition rates when compared with related methods.*

Palavras-chave: *Texture, Feature Space, Ranking Texture features, Multi-class Recognition, AdaBoost.*

1 Introduction

Image analysis is one of the main tasks involved for pattern recognition in image databases. The main goal is to get enough information to separate different image regions or to distinguish sample groups in classification tasks [1]. The acquisition of information for different patterns can be obtained through the computation of texture features (TF) and subsequent analysis [2]. Texture analysis [2] is a field of image processing that has been extensively applied for object recognition. A particular case is automatic classification of granite tiles, where the image acquisition can generate large resolution data. So, we need efficient methods to construct feature spaces in order to reduce the computational cost of further steps. The co-occurrence matrix and Haralick's texture features, or simply Haralick's descriptors, have been applies to compute the feature vectors to represent the sample images in such application [3]. These methodologies are classical in the pattern recognition and texture image analysis [4].

A co-occurrence matrix for a given image is generated by computing the distribution of co-occurring pixel values (gray scale values, or colors) at a given offset. Hence, we can describe texture through a set of features considering multiple directions defined by a set of offsets. To summarize the information contained in the co-occurrence matrix we follow [5] and compute the Haralick's texture features [6], in order to generate the feature space. In this paper, we apply the Multi-Class.M2 DPCA methodology described in [7, 8], for discriminant analysis (ranking texture features) on a multiclass problems in granite tiles recognition.

Given an N-class granite tiles database, the Multi-Class DPCA.M2 algorithm (MDPCA.M2) [7] builds a linear support vector machine (SVM) ensemble, composed of N SVM machines, to get the discriminant weights that are combined through the AdaBoost technique in order to determine the discriminant contribution of each texture features. The Multi-Class DPCA.M2 methodology applies the Multi-Class.M2 algorithm described in [7, 8] and combines the separating SVM hyperplanes through a simple strategy to compute the global discriminant weights for ranking texture features given by the group-differences. Also, we follow [9] and apply the Multi-Class.M2 methodology to compute discriminant weights through the linear discriminant analysis (LDA) [1]. We have focused here on the SVM and LDA methods but any other separating hyperplane could be used. Besides, we verify the consequences of ranking techniques in textures feature through Fisher criterion and the Multi-Class DPCA technique [10, 11]. The computational experiments are performed using only 5 of RGB granite images with resolution 4488×4488, obtained through a scanner from multifunctional HP Deskjet 2050. Before computation, we convert each image to gray scale and perform a partition of the images into blocks in order to better explore the high resolution data for recognition taks. The results show that the features selected by the proposed technique allow competitive recognition rates.

The paper is organized as follows. In section 2 we presented the techniques to feature extraction. Next, section 3 describes the Multi-Class.M2 DPCA for texture analysis. The computational experiments are presented in section 4. Finally, in section 5, we conclude the paper, summarizing its main contributions and describing further developments.

2 Feature Extraction

In this paper, the process of image feature extraction consists in the computation of Haralick's texture features through the gray level co-occurrence matrix (GLCM), computed considering multiple offsets, to represent the visual information [5]. So, given an $M \times N$ image

I coded using L grey levels in the set $\{0, 1, 2, \ldots, L-1\}$ and an offset $(\Delta x, \Delta y)$, the GLCM matrix is a $L \times L$ two-dimensional array define by:

$$GLCM\left(i, j, \Delta x, \Delta y, I\right) = \sum_{x=1}^{M} \sum_{y=1}^{N} \delta\left(I\left(x, y\right) - i, I\left(x + \Delta x, y + \Delta y\right) - j\right), 0 \leq i, j \leq (L-1) \tag{1}$$

where the function δ is defined as: $\delta\left(m, n\right) = 1$, if $m = n = 0$, and, $\delta\left(m, n\right) = 0$, otherwise.

Therefore, the GLCM characterizes the texture of an image by calculating how often pairs of pixels $\left(\left(x, y\right), \left(x + \Delta x, y + \Delta y\right)\right)$ with the gray level intensities (i, j) occur in a specific offset $(\Delta x, \Delta y)$ [5]. We shall emphasize that we are working in the digital context and, consequently, expression (1) only makes sense if $\Delta x, \Delta y \in \mathbb{N}$.

Figure 1 illustrates the process to build the GLCM by only walking in horizontal direction, $(\Delta x, \Delta y) = (1, 0)$.

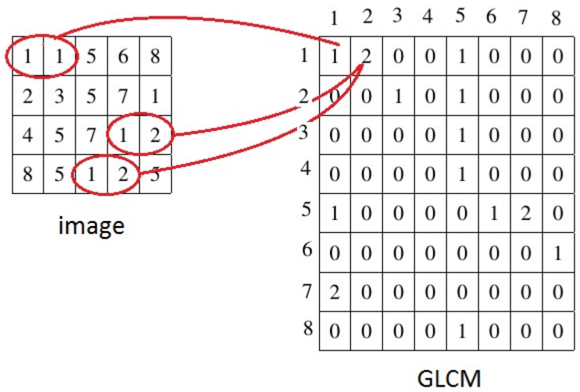


Figure 1: Process to create GLCM for image I, with $L = 8$ in expression (1).

In the Figure 1, $GLCM(1, 1, 1, 0, I) = 1$ because there is only one instance in the input image I where two horizontally adjacent pixels $I\left(x, y\right)$ and $I\left(x + 1, y\right)$ have values $I\left(x, y\right) = 1$ and $I\left(x + 1, y\right) = 1$. Analogously, the $GLCM(1, 2, 1, 0, I) = 2$ because there are two pairs of pixels such that $I\left(x, y\right) = 1$ and $I\left(x + 1, y\right) = 2$, and so on.

In this paper, the process of image feature extraction consists in calculating the Haralick's texture features through the GLCM matrix computed considering multiple directions to represent the visual information [12, 5].

Haralick extracted 14 descriptors from the co-occurrence matrix [6], but only five texture features are frequently used due to correlations between the descriptors [3]. So, in this work, we use only the five texture features presented in Table (1). According to [3], these descriptors are adequate to give good results in classification task.

In this work, for each Haralick's Descriptors presented in Table (1), we consider the offsets $(\Delta x, \Delta y) \in \{(1, 0), (1, 1), (0, 1), (-1, -1)\}$. Besides, $GLCM\left(i, j, \Delta x, \Delta y, I\right)$ that is defined by expression (1), is normalized such that the sum of its elements is equal to one [6]. In the above table, μ_x and μ_y are the horizontal and vertical mean while σ_x and σ_y denote the horizon-

Haralick's Descriptors	
Energy	$\sum_i \sum_j (GLCM(i,j,\Delta x,\Delta y,I))^2$
Contrast	$\sum_i \sum_j \lvert i-j\rvert^2 GLCM(i,j,\Delta x,\Delta y,I)$
Correlation	$\frac{\sum_i \sum_j (i-\mu_i)(j-\mu_j) GLCM(i,j,\Delta x,\Delta y,I)}{\sigma_i \sigma_j}$
Homogeneity	$\frac{\sum_i \sum_j GLCM(i,j,\Delta x,\Delta y,I)}{1+\lvert i-j\rvert}$
Entropy	$\sum_i \sum_j GLCM(i,j,\Delta x,\Delta y,I) log(GLCM(i,j,\Delta x,\Delta y,I))$

Table 1: Five texture features

tal and vertical standard deviations, defined by expressions (2)-(3):

$$\mu_x = \sum_i \sum_j iGLCM\left(i,j,\Delta x,\Delta y,I\right) \quad and \quad \mu_y = \sum_j \sum_i jGLCM\left(i,j,\Delta x,\Delta y,I\right) \quad (2)$$

$$\sigma_x = \sqrt{\sum_i \sum_j (i-\mu_x)^2 GLCM\left(i,j,\Delta x,\Delta y,I\right)} \quad and \quad \sigma_y = \sqrt{\sum_i \sum_j (j-\mu_y)^2 GLCM\left(i,j,\Delta x,\Delta y,I\right)} \quad (3)$$

3 Multi-Class Discriminant Analysis

The whole texture feature analysis is summarized in the pipeline of Figure 2 and the Multi-Class.M2 DPCA methodology is presented in the Algorithm 1. We follow [11, 7] and apply the technique presented in Section 2 for build texture features in the step (1) of the pipeline. Then, in step (2) of Figure 2, we compute a set of linear SVM hyperplanes, based on the "one-against-all" SVM multi-class approach. We also apply an ensemble technique, the AdaBoost.M2 algorithm, to combine the linear classifiers in order to compute the global discriminant vector. The key idea of this step is based on the fact that AdaBoost.M2 linearly combines weak classifiers to get the strong hypothesis. So, it is straightforward to obtain the global discriminant weights from the expression that defines the strong classifier by using a simple scheme, that corresponds to step (3) of Figure 2. This strategy can be also used to combine discriminant directions computed by other multi-class approaches, like linear discriminant analysis (LDA).

However, it is known that a strong learner like SVM does not work well as the base component for Adaboost [13]. Therefore, we follow [13] and implement a strategy to compute a weakened version of SVM that is useful as an Adaboost.M2 feature [9].

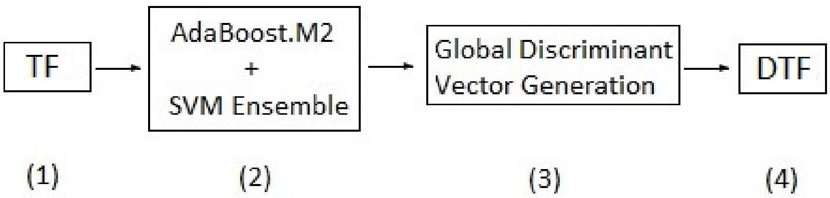


Figure 2: Flowchart with main steps of the proposed technique.

Finally, in the stage (4), we sort discriminant texture features (DTF) in the decreasing order of the global discriminant weights. The method is not restricted to any particular probability density function of the sample groups because it can be based on either a parametric or non-parametric separating hyperplane approaches.

In the Algorithm 1, at the input of the procedure, the training instances in the database $X \subset \Re^n$ are supposed independently and identically distributed from an uniform distribution D. In line 9 of MD-PCA.M2 algorithm, each weak learner generates an hypotheses, which has the form $h : X \times Y \to [0, 1]$, and can be interpreted as the probability that y is the correct label associated with instance $\mathbf{x}$. So, given a sample $\mathbf{x}_i$, the probability of choosing an incorrect label y is [9]: $Pr = \frac{1}{2}\left(1 - h(\mathbf{x}_i, y_i) + h(\mathbf{x}_i, y)\right)$.

However, we have $|Y| - 1$ possibilities to obtain the incorrect answer. So, we can define the loss of the hypothesis through a weighted average according to some $q_{i,y}$, called the label weighting function, that assigns to each example i in the training set a load, with $\sum_{y \neq y_i} q_{i,y} = 1$. The resulting formula is called the pseudo-loss of h on training instance i with respect to q [9]:

$$ploss_q\left(h, i\right) = \frac{1}{2}\left(1 - h(\mathbf{x}_i, y_i) + \sum_{y \neq y_i} q_{i,y} h(\mathbf{x}_i, y)\right). \tag{4}$$

So, following the AdaBoost.M2 strategy [9], in each iteration t of the Algorithm 1, the weak learner's goal is to minimize the expected pseudo-loss, computed in line 10 of the Algorithm 1, for a distribution D^t and weighting function q^t.

Next, the MDPCA.M2 computes a set of SVM hyperplanes, based on the one-against-all SVM multi-class approach presented in [14]. Hence, as we have N classes, the internal loop in the Algorithm 1 (line 6 to 9) constructs N weakened SVMs, in the texture features, using the Algorithm 2. In line 7 of Algorithm 1 we buid the Θ^y set by taking all k_y projected samples from class y and label them as 1. Then, using random sampling we choose $(2k_y)/(N-1)$ projected samples from classes other than y and label them as -1. The obtained set of feature vectors $\mathbf{x}_m^y \in \Re^{m'}$ and corresponding labels $y_m \in \{-1, 1\}$:

$$\Theta^y = \left\{\left(\mathbf{x}_1^y, l_1\right), \left(\mathbf{x}_2^y, l_2\right), ... \left(\mathbf{x}_{3k_y}^y, l_{3k_y}\right)\right\}, \tag{5}$$

are the input to call the Algorithm 2 which construct the weak SVM (WSVM) model y, represented by a hyperplane direction $\left(\phi_y^t\right)$ and a linear coefficient $\left(b_y^t\right)$.

The lines 16-18 of the Algorithm 1 are based on the AdaBoost.M2 idea of deriving a strong learner h_f by using the linear combination of weak (WSVM, in our case) learners $h^1, h^2, \cdots, h^T$:

$$h_f\left(\mathbf{x}\right) = \arg\max_{y \in Y} \sum_{t=1}^{T} \widetilde{\alpha}^t h^t(\mathbf{x}, y), \tag{7}$$

where $\widetilde{\alpha}^t$ is computed in line 16. This expression offers the possibility of extending the DPCA methodology to multi-class problems using the Adaboost.M2 result.

The output of the MDPCA.M2 procedure is the discriminant texture components $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{m'}$ where $\mathbf{q}_i$ is a texture feature component selected according to its discriminant weight $v\left(i\right)$.

On the other hand, if we compute the LDA in the texture feature space, we get $N-1$ hyperplane directions $\phi_{lda}^i \in \mathbb{R}^{m'}$, $i = 1, 2, \ldots (N-1)$. Consequently, we obtain in this case a LDA weight matrix $\phi_{lda}^{i,j}$, which can be processed according to lines 19-20 of Algorithm 1, by just replacing $\Phi_{i,y}$ by $\phi_{lda}^{i,j}$. The obtained global discriminant weights are named Multi-Class LDA-DPCA in the following sections.

Algorithm 1: Multi-Class.M2 DPCA Procedure

Input: Samples: $X = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \ldots (\mathbf{x}_M, y_M)\}$; where $y_i \in Y$ and $Y = \{1, 2, 3, .., N\}$; Distribution D over the M examples; Percentage μ;

Initialize the weight vector: $w^1_{i,y} = \frac{D(i)}{|Y|-1}, for \quad i = 1, \cdots, M; y \in Y - \{y_i\}$
Calculate texture features (TF) through of GLCM.
Build the labeled TF data set $\Theta = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2) \ldots (\mathbf{x}_M, y_M)\}$
for $t = 1, \ldots$ ***to*** T **do**
    for $y \neq y_i$: $q^t_{i,y} = \frac{w^t_{i,y}}{W^t_i}$; and set $D^t(i) = \frac{W^t_i}{\sum_{i=1}^N W^t_i}$
    for $y = 1, \ldots$ ***to*** N **do**
        Build the subset Θ^y, given by expression (5);
        $(\phi^t_y, b^t_y) = WSVM(\Theta^y, \mathcal{Y}, D^t, \mu)$ where $\mathcal{Y} = \{-1, 1\}$;
        Get hypothesis $h^t : X \times Y \longrightarrow [0, 1]$, given by $h^t(\mathbf{x}, y) = f\left(< \mathbf{x}, \phi^t_y > +b^t_y\right)$
    Compute:

$$e^t = \frac{1}{2} \sum_{i=1}^{N} D^t(i) \left(1 - h^t(\mathbf{x}_i, y_i) + \sum_{y \neq y_i} q^t_{i,y} h^t(\mathbf{x}_i, y)\right)$$

    if $e_t > 0.5$ **then**
        break;
    Calculate AdaBoost.M2 weights: $\alpha^t = \frac{1}{2} \ln\left(\frac{1-e^t}{e^t}\right)$;
    for $i = 1, \ldots N$ *and* $y \in Y - \{y_i\}$ **do**
        Update: $w^{t+1}_{i,y} = w^t_{i,y} exp(-\alpha^t(1 - h^t(x_i, y_i) + h^t(x_i, y)))$;
Normalize $\widetilde{\alpha}^t = \alpha^t / \sum_{j=1}^T \alpha^j$, $t = 1, 2, \ldots, T$
for $i = 1, \ldots$ ***to*** m' **do**

$$|\Phi_{i,y}| = \left| \sum_{t=1}^{T} \widetilde{\alpha}^t \frac{\phi^t_{i,y}}{z^t_{max,y} - z^t_{min,y}} \right|, y \in Y \tag{6}$$

Compute $v(i) = \max_{y \in Y} \{|\Phi_{i,y}|\}$, $i = 1, 2, \ldots, m'$
Sort discriminant weights: $v(1) \geq v(2) \geq \ldots v(m')$
Select the texture feature following $v(i)$

Output: discriminant texture components: $\mathbf{q}_1, \mathbf{q}_2, \cdots, \mathbf{q}_{m'}$

Algorithm 2: WSVM Procedure: Build a Weakened version of SVM.

Input: Labeled samples:$X = \{(\mathbf{x}_i, y_i), i = 1, 2, \cdots, n'\}$ where $y_i \in Y$ is the label of the sample $\mathbf{x}_i$;
Samples probability distribution $D(\mathbf{x}_i)$;
Percentage μ;
Select $\mathcal{J}$ so that, $\sum_{j \in \mathcal{J}} D(x_j) \leq (1 - \mu)$;
Select $(\mathbf{x}_i, y_i); i \in \mathcal{J}$, and define $D^* = D_{\mathcal{J}}$;
Compute the weighted data $X^* = \{(D^*_i \cdot \mathbf{x}_i, y_i), i \in \mathcal{J}\}$
Compute the (weak) SVM hyperplane ϕ_{svm} using X^*;
Output: WSVM hyperplane ϕ_{svm}, b.

4 Computational Experiments

In this section we perform granite images experiments maintained by CETEM [15]. We take five different granite tiles that are showed in Figure 3. In order to save memory allocation along the algorithms execution, we convert each granite to gray scale and resize it to 4480×4480 before computation. The high spatial resolution of each image allows to subdivide it into smaller blocks that keep the texture patterns. So, each image is partitioned into blocks using crop windows with size 280×280 and 320×320, generation new databases that we call $DB280$ and $DB320$, with 256 and 196 images, respectively. Next we take 128 and 98 images for training and 128 and 98 for test set obtain 640 and 490 images for train and 640 and 490 test respectively.

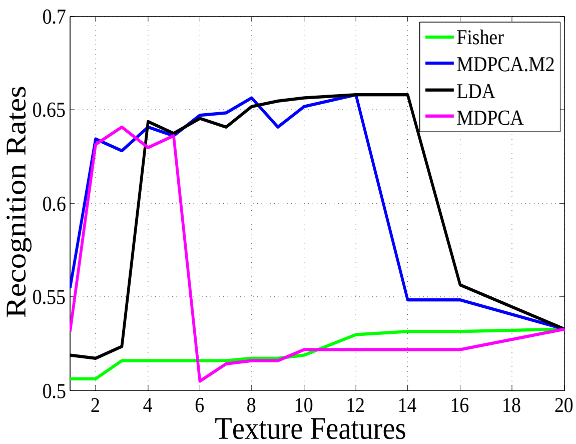
Figure 3: Granite Image database from CETEM

In the following, we consider the Multi-Class DPCA (MDPCA) and Fisher Criterion discriminant technique, presented in [11, 7] and the discriminant approaches explained in the section 3: Multi-Class.M2 DPCA (MDPCA.M2) and the Multi-Class LDA-DPCA (MLDA).

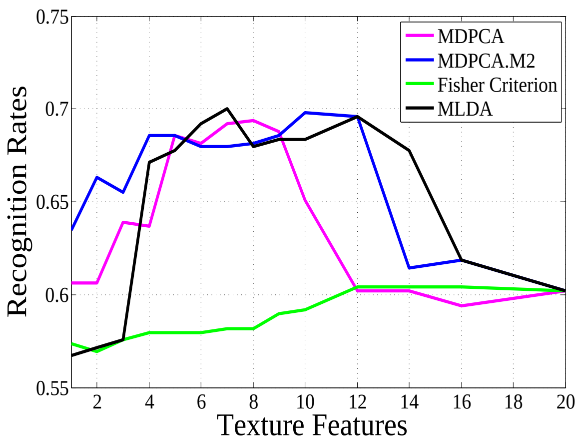
In the texture space, the mean of each class i has been calculated from the corresponding training images and the Euclidean distance from each class mean $\widehat{\mathbf{x}}_i$ has been used to assign a test observation $\mathbf{x}_r$ to either the different granite tiles. That is, we have assigned $\mathbf{x}_r$ to class i that minimizes:

$$d_i(\mathbf{x}_r) = \sqrt{\sum_{j=1}^{k} (x_{rj} - \widehat{x}_{ij})^2}, \tag{8}$$

where x_{rj} and $\widehat{x}_{ij}$ are the sample $\mathbf{x}_r$ and the mean $\widehat{\mathbf{x}}_i$, respectively, in the jth texture feature considered.



(a)



(b)

Figure 4: Recognition rates to texture features considering 32 gray level

The Figure 4 shows the average recognition rates of the 10-fold cross validation experiments of texture features using the discriminant techniques for the five-class classification problems above mentioned. The recognition rates presented in Figures 4.(a)-(b) were computed using the databases $DB280$ and $DB320$, respectively, both with 32 gray level to build texture features. We notice that in Figure 4 the MDPCA.M2 achieve highest recognition rates or perform closer to the best one. For instance, in the Figure 4.(b), the MDPCA.M2 is better than the other ones in the intervals for $1 \leq k \leq 5$ and $9 \leq k \leq 12$. Also, it is competitive with the other discriminant techniques for other intervals. Besides, the recognition rates of discriminant techniques showed in Figure 4.(b) are better than the recognition rates presented in Figure 4.(a). The Table 2 synthesizes some information presented in the Figure 4 related to maximum and minimum recognition rates.

Correlation of cropping in 32 gray level				
	cropping: 280		cropping: 320	
	Minimum	Maximum	Minimum	Maximum
MDPCA.M2	$k = 20$ (53%)	$k = 12$ (66%)	$k = 20$ (60%)	$k = 10$ (69%)
MDPCA	$k = 6$ (51%)	$k = 20$ (64%)	d $k = 16$ (58%)	$k = 8$ (68%)
MLDA	$k = 2$ (52%)	$k = 14$ (51%)	$k = 1$ (56%)	$k = 7$ (70%)
Fisher Criterion	$k = 1$ (51%)	$k = 20$ (53%)	$k = 2$ (56%)	$k = 16$ (61%)

Table 2: Table to show the effect of cropping in recognition rates

So, from Figure 4 and Table 2 we notice that the recognition rates for $DB320$ are higher than the ones obtained when using the $DB280$ database. Thus, the size of the crop window is an important parameter for the classification performance.

5 Conclusion and Future Works

This paper applies the Multi-Class.M2 DPCA algorithm for ranking texture features computed from multi-class granite image databases. The basic methodology has a computational complexity dominated by the AdaBoost.M2 algorithm plus texture feature computed through co-occurrence matrix and Haralick's descriptors. The granite image experiments show that, in general, the texture features selected by Multi-Class.M2 DPCA algorithm allow higher recognition rates using less features than Multi-Class LDA-DPCA, Multi-Class DPCA and Fisher Criterion. Besides, the recognition rates are very sensitive respect to the crop window size. This is an important parameter for our methodology and we must seek for nearly optimal values in further works as well as to explore deep learning methods based on convolutional neural networks (CNNs) in order to compute feature spaces components that will be processed by the discriminant techniques [16].

Bibliography

[1] T. Hastie, R. Tibshirani, and J. Friedman, "The elements of statistical learning," *Springer*, 2001.

[2] M. Petrou and P. García-Sevilla, *Image processing - dealing with texture*. Wiley, 2006.

[3] V. Arvis, C. Debain, M. Berducat, and A. Benassi, "Generalization of the cooccurrence matrix for colour images: application to colour texture classification," *Image Analysis & Stereology*, vol. 23, no. 1, pp. 63–72, 2011.

[4] F. Bianconi, E. González, A. Fernández, and S. A. Saetta, "Automatic classification of granite tiles through colour and texture features," *Expert Syst. Appl.*, vol. 39, no. 12, pp. 11 212–11 218, Sep. 2012. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2012.03.052

[5] N. Zayed and H. A. Elnemr, "Statistical analysis of haralick texture features to discriminate lung abnormalities," *Journal of Biomedical Imaging*, vol. 2015, pp. 12:12–12:12, Jan. 2015.

[6] R. M. Haralick, "Statistical and structural approaches to texture," *Proceedings of the IEEE*, vol. 67, no. 5, pp. 786–804, 1979.

[7] T. Filisbino, G. Giraldi, and C. Thomaz, "Ranking principal components in face spaces through adaboost.m2 linear ensemble," in *Graphics, Patterns and Images (SIBGRAPI), 2016 26th SIBGRAPI - Conference on*, São José dos Campos, SP, Brazil, Octuber 2016.

[8] T. A. Filisbino, G. A. Giraldi, and C. E. Thomaz, "Multi-class discriminant analysis based on linear classifiers and adaboost.m2 for ranking principal components in face spaces," Tech. Rep., 2016. [Online]. Available: http://www.lncc.br/departamentos/producaocientificageral.php?vMenu=2&vTipo=13&vCabecalho=pesq&vTitulo=lncc&vDepto=&idt_responsavel=&vAno=2016&ano=2016&anof=2016&idt_linha_pesquisa=

[9] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[10] H. Lu, K. Plataniotis, and A. Venetsanopoulos, "Mpca: Multilinear principal component analysis of tensor objects," *Neural Networks, IEEE Transactions on*, vol. 19, no. 1, pp. 18–39, Jan 2008.

[11] T. Filisbino, D. Leite, G. Giraldi, and C. Thomaz, "Multi-class discriminant analysis based on svm ensembles for ranking principal components," in *36th Ibero-Latin Am. Cong. on Comp. Meth. in Eng. (CILAMCE)*, Nov 2015.

[12] A. Felix, M. Oliveira, A. Machado, and J. Raniery, "Using 3d texture and margin sharpness features on classification of small pulmonary nodules," in *Graphics, Patterns and Images (SIBGRAPI), 2016 26th SIBGRAPI - Conference on*, São José dos Campos, SP, Brazil, Octuber 2016.

[13] E. Garcia and F. Lozano, “Boosting Support Vector Machines,” in *Proceedings of International Conference of Machine Learning and Data Mining (MLDM’2007)*. Leipzig, Germany: IBal publishing, Jul. 2007, pp. 153–167.

[14] E. Yildizer, A. M. Balci, M. Hassan, and R. Alhajj, “Efficient content-based image retrieval using multiple support vector machines ensemble,” *Expert Syst. Appl.*, vol. 39, pp. 2385–2396, Feb. 2012.

[15] M. B. da Silva, “CETEM/NRES Centro de Tecnologia Mineral, Núcleo Regional do Espírito Santo.”

[16] D. Mo, “A survey on deep learning: one small step toward ai,” *Dept. Computer Science, Univ. of New Mexico, USA*, 2012.