A NOVEL FEATURE-EXTRACTION ALGORITHM FOR EFFICIENT CLASSIFICATION OF TEXTURE IMAGES

Ionuț MIRONICĂ[1], Radu DOGARU[2]

In this paper, a non-linear model is investigated for texture characterization and retrieval. The power of our descriptors was validated both in the context of a classification system and as part of an information retrieval approach. For this purpose, we have used four different texture databases and we have compared our descriptor with state of the art algorithms. In most of experiments, our approach has archieved best results on most of the recognition and retrieval problems.

Keywords: content based image retrieval systems, texture recognition

1. Introduction

During the last 20 years Content Based Image Retrieval (CBIR) established itself as a domain with an important role in application areas such as multimedia database systems. A major part of the work focused on low level feature study like texture. Textures can usually be described informally as the output of some physical process wherein local structure is repeated seemingly at random [1].

The main purpose of this paper is to show improvements of CBIR systems using classification algorithms. We aim to select the best-suited classifiers by making a comparison of various classification methods for certain image databases. In this paper a novel feature classification is introduced, inspired by nonlinear diffusive operators previously used to quantify the degree of randomness in an image pattern generated by a 2-dimensional cellular automaton [15]. The relevance and advantages of this new feature extraction method for texture images classification as required by CBIR (Content Based Image Retrieval) systems is extensively investigated through comparisons with other methods previously cited in the literature. A wide set of benchmark image databases was selected in order to select the best suited classification method (both feature classification and classifier) including the novel feature extraction method called next a Nonlinear Diffusive Transform (NDT).

[1] PhD student, Image Processing and Analysis Laboratory (LAPI), e-mail: imironica@ alpha.imag.pub.ro

[2] Prof., Natural Computing Laboratory, Dept. of Applied Electronics and Information Engineering, University POLITEHNICA of Bucharest, Romania, e-mail:radu_d@ieee.com

In addition to performance aspects (such as the percentage of correctly classified patterns), efficiency and implementation complexity issues are considered. In this respect it appears that our novel transform allows to obtain good classification performance while having a reduced implementation complexity.

The paper is organized as follows: we describe in Section II the previous work for texture detection content-based image retrieval systems, including a brief discussion about classical descriptors. Then we describe a novel approach for texture detection (Section III). Experiments are discussed in Section IV and conclusions are presented in Section V.

2. Previous Work

One of the earliest and most successful texture descriptors is the run-length matrix. From the original run-length matrix *p(i; j)*, many numerical texture measures can be computed. Galloway [2], Chu [3] and Dasarathy [4] have proposed different sets of original features.

A co-occurrence matrix [5] or co-occurrence distribution is a matrix or distribution that is defined over an image to be the distribution of co-occurring values at a given offset. Mathematically, a co-occurrence matrix C is defined over an image *I* (with *m* and *n* dimensions), parameterized by an offset *(dx,dy)*, as:

$$C_{\Delta X,\Delta Y} = \sum_{p=1}^{n}\sum_{q=1}^{m}\begin{cases}1, I(p,q) = i; I(p+\Delta X, y+\Delta Y) = j\\ 0, otherwise\end{cases} \tag{1}$$

The 'value' of the image originally referred to the grayscale value of the specified pixel. The value could be anything, from a binary on/off value to 32-bit color and beyond.

Image moments [6] provide a measurement for color similarity between images. There are three central moments of an image's color distribution: mean, standard deviation and skewness.

The simplest non-parametric approach for density estimation is histogram calculation [7]. Color Histogram is a representation of the distribution of colors in an image.

Color Coherence Vectors [8] represent the degree to which pixels of that color are members of large similarly-colored regions. Computing color coherence vectors involves two main steps:

- use a mean filter to eliminate small variations between adjacent pixels (using an eight per eight mask)
- quantify the color space (HSV) into *24* channels (*16* for Hue, *4* for Saturation and *4* to Value)
- classify pixel as coherent or incoherent and create two histograms (2 x 24 features). A coherent pixel is part of a large group of pixels of the same color

(all adjacent pixel have the same color), while an incoherent point represent an edge pixel.

The corelogram [9] of a gray-scale image I is defined for $i, j \in [m], d \in [n]$ as:

$$\gamma_{ci,cj}^{(k)}(I) = \Pr_{p_1 \in I_{ci}, p_2 \in I_{c2}} (p_2 \in I_{cj} \mid p_1 - p_2 \mid= d) \qquad (2)$$

where *Pr* is the probability of appearance for two pixels *(* p_1, p_2 *)* having similar colors I_{cj} , and distance between these two pixels is equal to *d*.

The edge histogram descriptor (EHD) represents the local edge distribution by dividing image space into *4×4* sub images and representing the local distribution of each sub image by a histogram. In the sense of generating histograms, edges in all sub images are categorized into five types: vertical, horizontal, diagonal and no directional edges (namely edges with no particular directionality), resulting in a total of *5 × 16 = 80* histogram bins [10].

The Homogenous Texture Descriptor [11] describes directionality, coarseness, and regularity of patterns in images and is most suitable for a quantitative characterization of texture that has homogenous properties. It provides a quantitative representation using *62* numbers (quantified to *8* bits each) that is useful for similarity retrieval. The extraction is done as follows: the image is first filtered with a bank of orientation and scale tuned filters using Gabor filters. The first and the second moments of the energy in the frequency domain from the corresponding sub-bands are then used as the components of the texture descriptor.

3. Proposed Non-Linear Method

Our approach is inspired by the Cellular Automata Theory [12]. A cellular automaton [13] is a discrete model studied in computability theory, mathematics, physics, complexity science, theoretical biology and microstructure modeling. It consists of a regular grid of cells, each in one of a finite number of states, such as "On" and "Off" (in contrast to a coupled map lattice). The grid can be in any finite number of dimensions. For each cell, a set of cells called its neighborhood (usually including the cell itself) is defined relative to the specified cell. For example, the neighborhood of a cell might be defined as the set of cells a distance of two or more from the cell.

The first task is to transform the image in a binary lattice. To create binary images, we use a thresholding process with a various number of limits. During the thresholding process, individual pixels in an image are marked as "object" pixels if their value is greater than some threshold value (assuming an object to be brighter than the background) and as "background" pixels otherwise. We have used in our experiments a fixed number of equally spaced thresholds (from one to 64 thresholds)

Using these thresholds, we have extracted a number of binary images. For every binary image, we have extracted two features using the following formula:

$$C = \frac{1}{M \cdot N} \sum_{i=1}^{N} \left[\sum_{j=1}^{M} | F(i,j) | \right] \tag{3}$$

where M and N are the image width and height and *F(i,j)* is a kernel function, computed on current pixel neighborhood (Fig. 1). The kernel function is defines as:

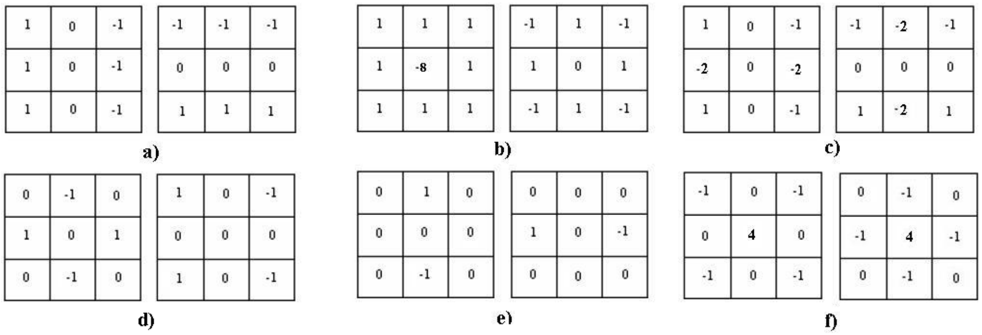
$$F(i,j) = \sum_{k \in N_{i,j}} \left| I_{i,j}(k) A(k) \right| \tag{4}$$

where $N_{i,j}$ is the *3x3* neighborhood centered around the *(i,j)* location, $I_{i,j}(k)$ is a pixel value at location *k* (*k=1..9)* in the neighborhood centered on *(i,,j)* and $A(k)$ is one of the 3x3 template matrices presented in Fig. 1.

$I_{i-1,j+1}$	$I_{i,j+1}$	$I_{i+1,j+1}$
$I_{i-1,j}$	$I_{i,j}$	$I_{i+1,j}$
$I_{i-1,j-1}$	$I_{i,j-1}$	$I_{i+1,j-1}$

Fig. 1. 3x3 neighborhood for kernel function

There are a high number of possible functions that can be used. For example, using the von Neumann neighborhood it is possible to have as many as $2^{2^5} = 2^{32} \cong 4295 \cdot 10^{29}$ different cell genes, which makes searching emergent phenomena a time-consuming process. We have been tested a reduced number of possibilities, namely six variants of functions (a – to f in Fig.1), using 3x3 neighborhood. These functions are similar to kernels used for edge detection (Prewitt, Sobel, the Laplacian operator and Robert's cross operator) [26]. Fig. 2 presents the templates used in our experiment.



$$\begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

a)

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \begin{bmatrix} -1 & 1 & -1 \\ 1 & 0 & 1 \\ -1 & 1 & -1 \end{bmatrix}$$

b)

$$\begin{bmatrix} 1 & 0 & -1 \\ -2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & -2 & 1 \end{bmatrix}$$

c)

$$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 0 & -1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{bmatrix}$$

d)

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & -1 & 0 \end{bmatrix} \quad \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & -1 \\ 0 & 0 & 0 \end{bmatrix}$$

e)

$$\begin{bmatrix} -1 & 0 & -1 \\ 0 & 4 & 0 \\ -1 & 0 & -1 \end{bmatrix} \quad \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

f)

Fig. 2. Six templates for computing nonlinear parameters

It was demonstrated in [28], that a value of C close to 1 indicates a homogeneous state while a value of *C=0.5* is a measure of a perfect (high frequency) chaotic pattern. At the other extreme *C=0* indicates the presence of perfectly regular chess-board pattern. Consequently, such synthetic indicators as C are strongly correlated with the human perception. Using various A templates ensures that various directions of interests in the image are better characterized.

To improve the feature performance, we applied the calculation of C (using equation 3) for to different image scales (*s*=1, 0.5, 0.25, 0.125 and 0.0625). The distance between two neighbor pixels (where template matrices A apply) is respectively *d=1/s*. The reason of using different image scales is that computing different texture resolution, we compute different coarseness: one macro texture of high coarseness and one micro texture of low coarseness. Using 7 thresholds, 3 scales with two template matrices each (left and right in Fig.1) per each scale results in a 42-dimensional feature vector associated with an image.

4. Experimental Results

Four image databases are used in our experiments (Fig. 3):

- The Vistex database with 900 images (9 images per class [23].

- The UIUC[22] database with 25 texture classes, 40 samples each. All images are in grayscale JPG format, 640x480 pixels.

- The Brodatz's photo album (Brodatz 1966) [24] is a well known benchmark database used to evaluate texture recognition algorithms. It contains 111 different texture classes. For each class, it is represented by only one sample, which is then divided into 9 sub-images non-overlappingly to form the database. Thus, there are 999 images altogether with resolution of 215x215.

- The KTH-Tips [25] database which contains 10 textures under different illumination, pose and scale (81 images per class)

As far as texture descriptors are concerned, we test several state of the art approaches from the existing literature which are known to be successfully employed to the CBIR task, namely: GrayScale Histogram, Color Coherence Vectors, Image Moments, Cooccurrence Matrix texture, Auto-Correlogram, Run-Lenght Matrix, Edge Histogram Descriptor and Homogeneous Texture Descriptor.

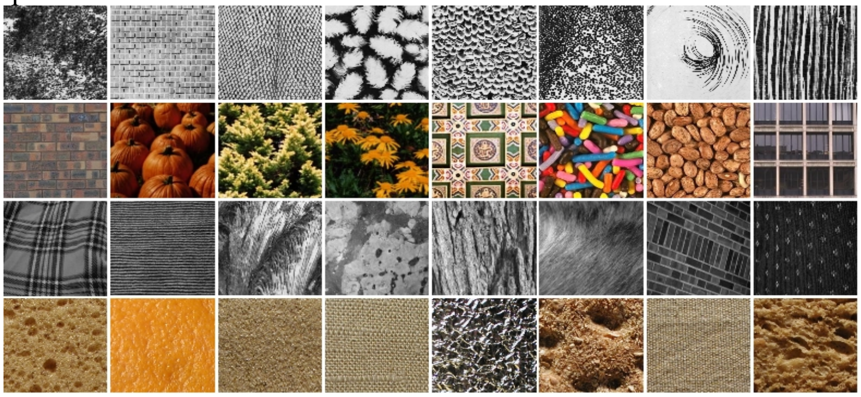


Fig. 3. Example of images used in experiment : first line – The Brodatz's database, Vistex database in second line, UIUC database in third line and KTH database in fourth line

To assess the retrieval performance, we have used several measures. First, we have computed the classical precision and recall chart. Precision is the fraction of retrieved documents that are relevant to the search (measure of false positives) and recall is the fraction of the documents that are relevant to the query that are successfully retrieved (measure of false negatives). The system retrieval response is assessed with the precision-recall curves which plots the precision for all the recall rates that can be obtained according to the current image class population.

Second, to provide a global measure of performance we determine the overall Mean Average Precision - MAP as the area under the uninterpolated precision-recall curve (http://trec.nist.gov/trec_eval/). The evaluation consists of systematically considering each image from the database as query image and retrieving the remainder of the database accordingly. Precision, recall and MAP are averaged over all retrieval experiments. Experiments were conducted for various browsing windows, ranging from 9 to 40 images, depending the number of textures per class.

4.1. Choosing the algorithm's parameters

In the first experiment we have analyzed the influence of the parameters on the system performance. We have compared various strategies for each test database. We have varied the number of the threshold (using only one image scale) in Fig. 4, the number of the image scales using only one threshold in Fig. 5 and we have tested various kernel functions performance in Fig. 6.

By plotting the chart of MAP performance against the threshold values, the first thresholds will add much performance, but at some point the marginal gain will drop, giving an angle in the graph. We have the same scenario, by varying the number of images scales. Using more than three image scales, it will not add more information. It can be observed in Fig. 5, that the performance of various kernel functions is similar in most cases, with a little increase of performance using function kernel a. The lowest performance is obtained using the second kernel functions.

In our experiments, we have chosen a number of seven to fifteen thresholds with three image scales and the kernel function a). We have chosed this combinations of kernels, number of thresholds and image scales because it provide a reasonable balance between MAP performance and descriptors computational complexity. Adding a large number of thresholds and image scales will not improve the algoritm performance with more than one percent. If we use a smaller descriptor length, it will decrease the algorithm performance with more then 5 percents.

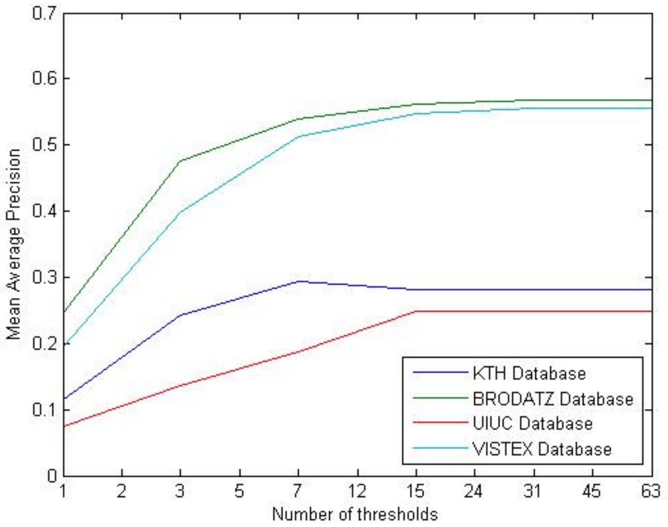


Fig. 4. Mean Average Precision using different number of thresholds (using only one image scale)

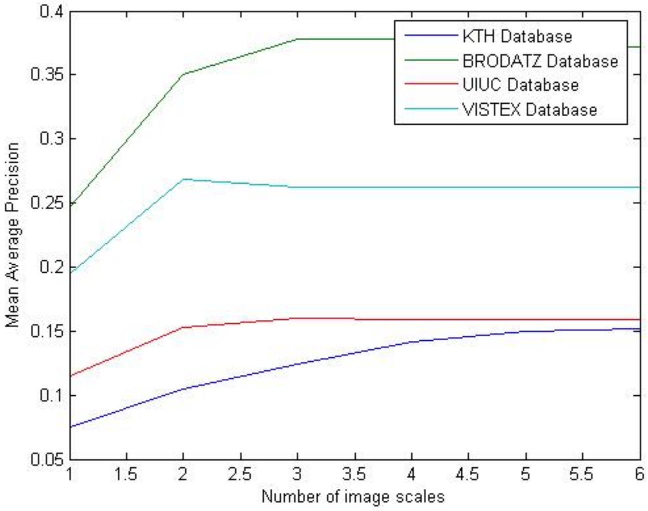


Fig. 5. Mean Average Precision using different number of image scales (using only one threshold)

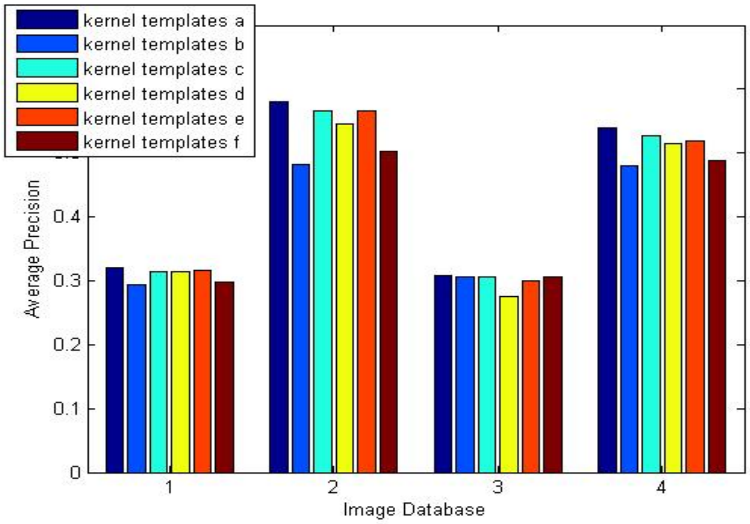


Fig. 6. Mean Average Precision using different nonlinear kernel templates (1 - UIUC database, 2 - Brodatz's database, 3 - KTH database and 4 - Vistex)

4.2. Retrieval Experiment

In Fig. 7, we present the precision-recall curves computed for all four image test databases.

We compare our method (red dotted line) with the following state of the art algorithms: GrayScale Histogram (black dotted line), Color Coherence Vectors (yellow dashed line), Image Moments (magenta dashed line), Cooccurrence Matrix texture (green dashed line), Auto-Correlogram (green dotted line), Run-Length Matrix (cyan dashed line), Edge Histogram Descriptor (blue dashed line), and Homogeneous Texture Descriptor (black dashed line). The best MAP results are presented in Table 1.

Table 1

Improvement achieved by the proposed algorithm (MAP values)

Database	1st MAP	2nd MAP	3rd MAP
KTH	31.95% - Our Approach	30.34% - HTD	23.43% - CCV
Brodatz	60.58% - HTD	59.94% - Our Approach	39.45% - CCV
UIUC	33.43% - Our Approach	32.56% - CCV	24.90% - Coocurence Matrix
Vistex	66.33% - HTD	56.73% - Our Approach	48.05% - Coocurence Matrix

Discussion on the results. We have obtained the best results with our approach in two cases: for the KTH and UIUC database, and the second position on Brodatz and Vistex database. We have also obtained good results using the Homogenous Texture Descriptor, Color Coherence vectors and Coocurence Matrix. The worst performance was obtained using the Image Moments, Edge Histogram Descriptor and Run-Length Matrix. In most of cases the performance of our algorithm is double than of these algorithms. Better results are obtained with GrayScale Histogram and Auto-Corelogram, but the differences are greater too (from 10% to 25 %).

The computational complexity and descriptors sizes are presented in Table. 2. Our approach has the lowest computational complexity, equal to Histogram, CCV, EHD and image moments. Homogenous Texture Descriptor has the biggest computational complexity ($O(n^2 \log(n))$), and Run Length Matrix, Auto Corelogram and Coocurence matrix have a bigger complexity of the algorithm than our approach. Another criterion for feature comparison is the descriptor length. Our descriptor has 42 features (three scales and seven thresholds). Four descriptors have smaller sizes (Histogram, Image Moments, Run Length Matrix and Coocurence Matrix), while Color Coherence Vectors, Auto-Corelogram, Homogenous Texture Descriptor and Edge Histogram Descriptor do not.

Table 2

Comparation of Computational complexity and descriptor size

Algorithm	**Computational complexity**	**Descriptor size**
Histogram	O(n)	24
CCV	O(n)	48

Image moments	O(n)	9
Run Length Matrix	O(n) + O(k·m) – where k is the number of colors and m the maxim size of run-length	23
Auto Corelogram	O(n) + O(k·m) – where k is the number of colors and m the number of neighborhood	96
Coocurence matrix	O(n) + O(k^2) where k is the number of image colors	16
EHD	O(n)	80
HTD	O(n^2 log(n))	64
Our Approach	O(n)	42

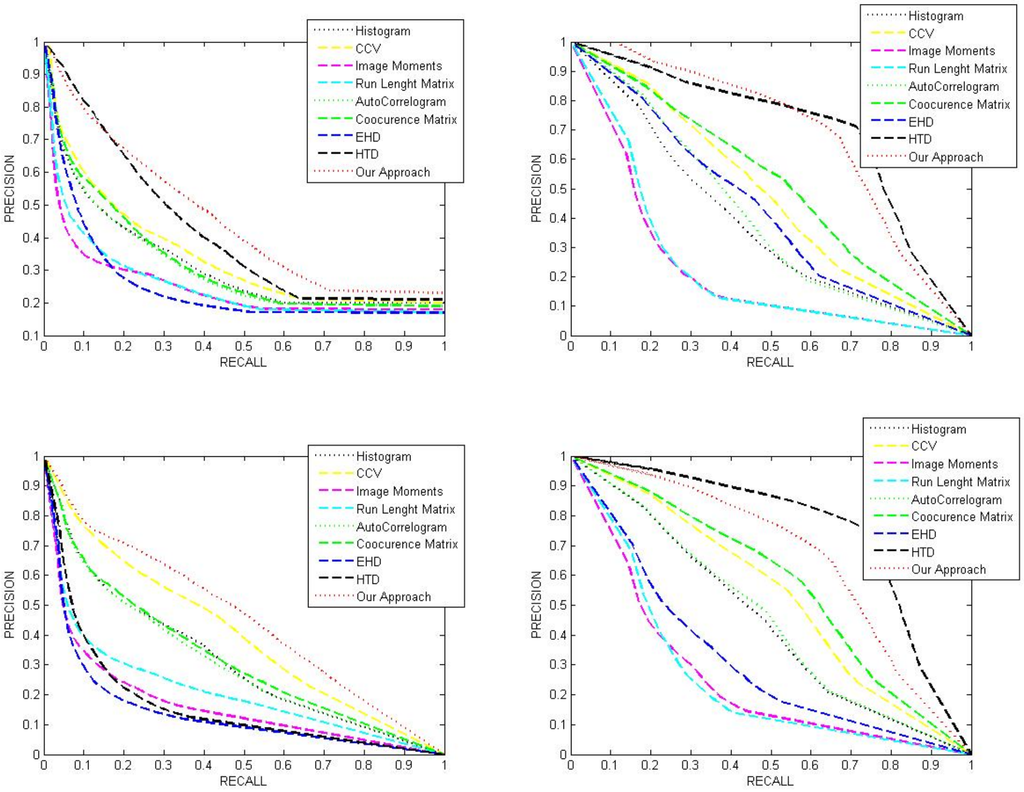


Fig. 7. Precision-recall curves for different content descriptors and test databases.

4.3. Recognition Experiment

In the second experiment, we address texture categorization from the perspective of machine learning techniques. We attempt to regroup the data according to related clusters. For classification we use the OpenCV [19] environment which provides many implementations of the classification algorithms.

We have tested the following methods: Naive Bayes [16], Nearest Neighbor [18] to SVM [17] (linear and RBF kernel), Random Trees [19], Gradient Boosted Trees[20], Extremely Random Forest [21]. Method parameters were tuned based on preliminary experimentations.

As the choice of training data may distort the accuracy of the results, we use a cross validation approach. The data set is split into train and test sets. We use different values for the percentage split 25% to 75%. To assess performance we compute the average precision.

Discussion on the results. In Figs. 7, 8, 9 and 10 we present the overall average correct classification for a selection of seven machine learning techniques (the ones providing the most significant results) on the several image databases: Brodatz, UIUC, KTH, and Vistex. The global results are very promising. The most accurate classification is obtained when using our approach in combination with Extremely Random Forests, Random Trees, Naive Bayes and SVM with RBF kernel. The highest average precision is up to 97% while the average of maxim correct classification is up to 92%. In terms of classification technique, the most accurate proves to be a the Extremely Random Forests, followed very closely by SVM with RBF kernel and further Random Trees, Naive Bayes, Nearest Neighbor, Gradient Boosted Trees and finally SVM with linear kernel.

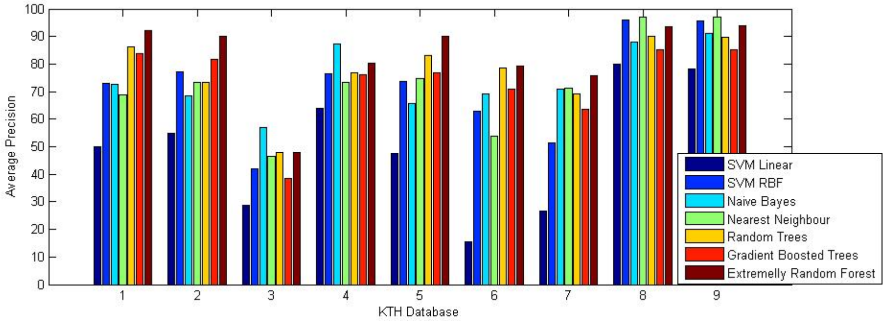


Fig. 8. Classification results using different classification methods (Naive Bayes, Nearest Neighbor, SVM linear, SVM with RBF kernel, Random Trees, Gradient Boosted Trees, Extremely Random Forest) and content descriptors(1.Histogram, 2. CCV, 3. Image Moments, 4. Run Lenght Matrix, 5. AutoCorrelogram, 6. Coocurence Matrix, 7. EHD, 8. HTD, 9.Our Approach) on Brodatz database

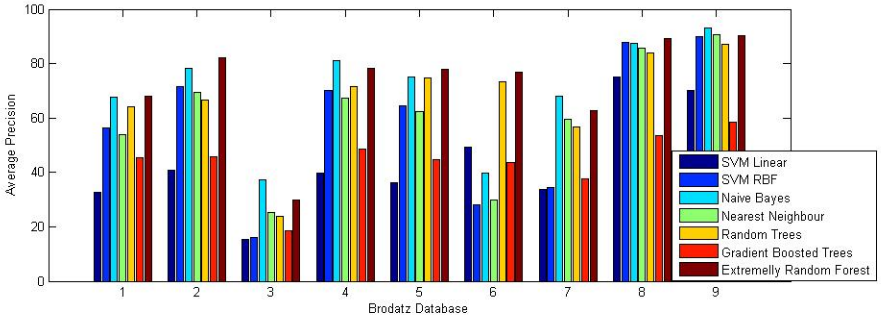


Fig. 9. Classification results using different classification methods (Naive Bayes, KNN, SVM linear, SVM with RBF, Random Trees, Gradient Boosted Trees, Extremely Random Forest) and content descriptors (1.Histogram, 2. CCV, 3. Image Moments, 4. Run Lenght Matrix, 5. AutoCorrelogram, 6. Coocurence Matrix, 7. EHD, 8. HTD, 9.Our Approach) on UIUC database

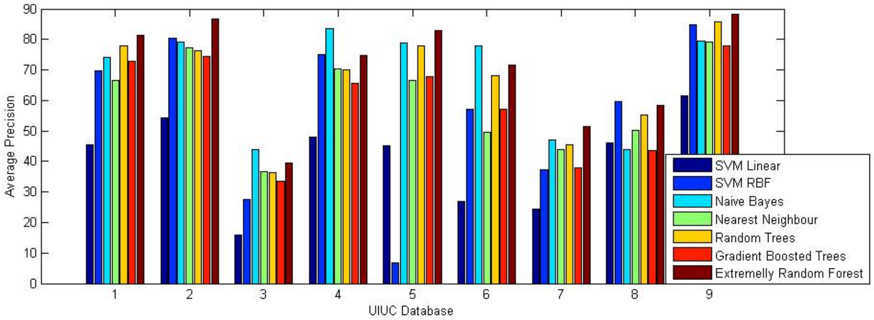


Fig. 10. Classification results using different classification methods(: Naive Bayes, Nearest Neighbor, SVM linear, SVM with RBF kernel, Random Trees, Gradient Boosted Trees, Extremely Random Forest) and content descriptors(1.Histogram, 2. CCV, 3. Image Moments, 4. Run Lenght Matrix, 5. AutoCorrelogram, 6. Coocurence Matrix, 7. EHD, 8. HTD, 9.Our Approach) on KTH database

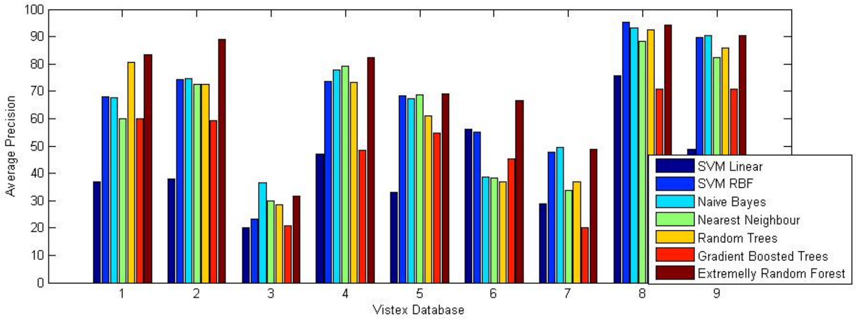


Fig. 11. Classification results using different classification methods (Naive Bayes, Nearest Neighbor, SVM linear, SVM with RBF kernel, Random Trees, Gradient Boosted Trees, Extremely Random Forest) and content descriptors(1.Histogram, 2. CCV, 3. Image Moments, 4. Run Lenght Matrix, 5. AutoCorrelogram, 6. Coocurence Matrix, 7. EHD, 8. HTD, 9.Our Approach) on Vistex database

5. Conclusions

Experimental results on a wide spectrum of benchmark problems suggest that given its simplicity, our approach may be a good alternative for texture detection and recognition. In most of experiments our approach achieves the best results on recognition and retrieval problems.

Future improvements will mainly consist of fine tuning and adapt the method to address a higher diversity of image categories. We will try to implement other kernel functions and other threshold strategies.

REFERENCES

[1] *S. Santini,* Exploratory Image Databases Content-based Retrieval, Academic Press, 2001

[2] *M. M. Galloway,* Texture analysis using gray level run lengths, Comput. Graphics Image Process., vol. 4, pp. 172–179, June 1975.

[3] *A. Chu, C. M. Sehgal, J. F. Greenleaf,* Use of gray value distribution of run lengths for texture analysis, Pattern Recognit. Lett.,vol. 11, pp. 415–420. June 1990.

[4] *B. R. Dasarathy, E. B. Holder,* Image characterizations based on joint gray-level run-length distributions, Pattern Recognit. Lett., vol. 12, pp. 497–502, 1991.

[5] *R. M. Haralick, K Shanmugam, I. Dinstein,* Textural Features for Image Classification. IEEE Transactions on Systems, Man, and Cybernetics SMC-3 (6): 610–621, 1973.

[6] *M. Stricker, M. Orengo*. Similarity of color images. In In SPIE Conference on Storage and Retrieval for Image and Video Databases III, volume 2420, pages 381392, Feb. 1995.

[7] *M. J. Swain D.H. Ballard,* Color indexing, International Journal of Computer Vision, 7(1), pp.11-32, 1991.

[8] *G. Pass, R. Zabih*, Histogram renement for content based image retrieval, IEEE Workshop on Applications of Computer Vision, pp. 96-102, 1996.

[9] *J. Huang, S. Kumar, M. Mitra, W. Zhu, R. Zabih*, Spatial color indexing and applications, Intl. J. of Computer Vision, vol. 35, no. 3, pp. 245–268, 1999

[10] *B. S. Manjunath, P. Salembier, T. Sikora,* Introduction to MPEG-7: Multimedia Content Description Interface. John Wiley and Sons, pp. 214-229 (2002)

[11] *B. S. Manjunath, J.-R. Ohm, V. V. Vasudevan*,“ Color and Texture Descriptors”, in IEEE Trans. on Circuits and Systems for Video Technology, vol. 11, no. 6, 2001, pp. 703-715.

[12] S. *Wolfram,* Statistical mechanics of cellular automata. Reviews of Modern Physics 55 (3): 601–644, 1983.

[13] *D. Dennett,* Darwin's Dangerous Idea, Penguin Books, London, ISBN 978-0-14-016734- 4, ISBN 0-14-016734-X, 1995

[14] *R. Dogaru,* Systematic design for emergence in cellular nonlinear networks – with applications in natural computing and signal processing, Springer-Verlag, Berlin Heidelberg, 2008, ISBN: 978-3-540-76800-5, 165 pages

[15] *R. Dogaru,* “Fast and efficient speech signal classification with a novel nonlinear transform”, in Proceedings of ISITC2007, International Symposium on Information Technology Convergence, 23-24 november, Jeonju, Republic of Korea.

[16] *H. Zhang,* The Optimality of Naive Bayes, AAAI Press, 2004

[17] *V. N. Vapnik*, Statistical Learning Theory. New York: John Wiley & Sons, 1998.

[18] *K. Beyer, J. Goldstein, R. Ramakrishnan, Uri Shaft:* When Is Nearest Neighbor Meaningful? Database Theory ICDT Lecture Notes in Computer Science, Volume 1540/1999

[19] *L. Breiman*. Random forests. Machine Learning Journal, 45:5–32, 2001.

[20] *J. H. Friedman*, Greedy function approximation: A gradient boosting machine. Technical report, Department of Statistics, Stanford University, 1999

[21] *P. Geurts, D. Ernst, L. Wehenkel.* Extremely randomized trees. Machine Learning, 63(1):3–42, 2006.

[22] *S. Lazebnik, C. Schmid, J. Ponce,* A Sparse Texture Representation Using Local Affine Regions. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 27, no. 8, pp. 1265-1278, August 2005.

[23] *http://vismod.media.mit.edu/vismod/imagery/VisionTexture*

[24] *P. Brodatz*, Textures: A Photographic Album for Artists and Designers, Dover, New York, NY, 1966.

[25] *B. Caputo, M. Frits, E. Hayman, J.O. Eklundh,* The kth-tips database. Available at http://www.nada.kth.se/cvap/databases/kth-tips, 2004.

[26] *R. Maini, H. Aggarwal*, "Study and Comparison of Various Image Edge Detection Techniques", International Journal of Image Processing (IJIP), 2009, 3 (1), pp 1-60.

[27] *R. Dogaru, M. Glesner*, Novel tools and methods for fast identification of emergent behaviors in CNNs with relevance to biological modeling , in Proceedings CNNA2004 (IEEE Int l Workshop on Cellular Neural Networks and their Applications, Budapest 22-24 July), pp. 339-345, 2004

[28] *R. Dogaru, R. Tetzlaff, M. Glesner,* Semi-Totalistic CNN Genes for Compact Image Compression , Cellular Neural Networks and Their Applications, 2006. CNNA '06. 10th International Workshop on 28-30 Aug. 2006, Istanbul, pp. 1 - 6