Texture Image Segmentation Approach Based on Neural Networks

https://doi.org/10.3991/ijes.v6i1.8166

Khalid Salhi(✉), El Miloud Jaara, Mohammed Talibi Alaoui
Mohammed First University, Oujda, Morocco
salhi.0.khalid@gmail.com

Abstract—One of the major problems in texture analysis is segmenting images into different regions based on textures. In this paper, we present a new approach of texture segmentation, which is based on both Kohonen maps and mathematical morphology, using three different texture features, namely, Haralick features based on gray-level co-occurrence matrix (GLCM), fractal features based on fractal dimension using the differential box counting method, and wavelet features based on wavelet transform. These features are used to train the Kohonen Network, which will be represented by the underlying probability density function (PDF). The segmentation of this map's representation is made by morphological watershed transformation. In the final part of our algorithm, this will help on the segmentation of the textural image, by assigning each pixel to a modal region extracted from the map. Our work covers the results obtained by the three extraction methods taking into consideration the execution time and the error rate.

Keywords—Segmentation, texture features, neural networks, watershed.

1 Introduction

In order to interpret an image, the texture is a rich information component, thus considered an important feature for the detection and segmentation of different objects of the same color.

The purpose of image segmentation is to seek the number and types of textures in this image, and which are the textures of each particular region. The application of the segmentation generally involves two steps, the first step is to extract the texture features for each pixel in the image, and the second step is to use these features to determine the uniform regions.

In this paper, we present three texture feature extraction methods. The first calculates the Haralick features extracted from Grey Level Co-occurrence Matrix (GLCM), the second computes local fractal features using the differential box counting method, and the third uses wavelet transform to calculate wavelet features.

After calculating the features of one of the three methods, we use our approach for unsupervised classification based on a morphological and neural concept, which, unlike other methods such k-means, doesn't oblige us to know the number of regions in our texture image.

First, for each vector to classify, we make a projection on a two-dimensional self-organizing map, named Kohonen self-organizing feature map. To help extract the homogeneous regions in this map, we present in the first stage the information in each cell of this map with the probability density function value (PDF) estimated by a nonparametric procedure, in the second stage we extract automatically the modal regions using watershed transformation. The segmentation phase consists in comparing the weight vector of each pixel of the image to the vectors corresponding to the detected modal regions, in order to assign each pixel to one of the extracted classes.

In the last section of our paper, we start by presenting the result of our comparative study of the effectiveness of three different features, by calculating the error rate and the execution time, then the results of comparison between our segmentation approach and the k-means clustering.

2 Features Extraction

As any statistical approach for segmenting textured images, our approach starts by selecting a set of features characterizing the local texture. In our study, we use three observations' samples, the first is constituted by Haralick features extracted from GLCM matrices, the second consists of Fractal features calculated by the differential box-counting method, and the third is composed of Wavelet features extracted using the wavelet transform.

These three methods of extraction are based on the same principle, as shown in Figure 1. An analysis window is assigned to each pixel of the image to extract a vector of this window which will characterize the local textural information around each pixel.

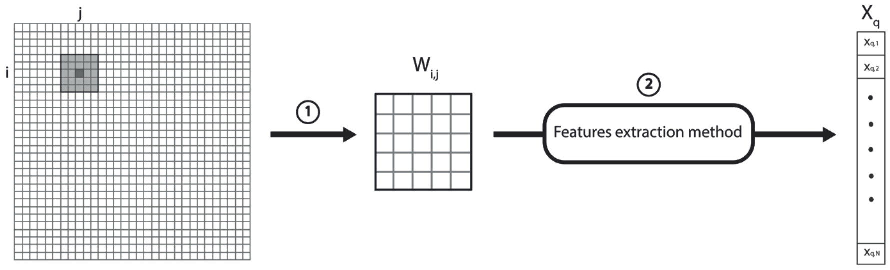

Fig. 1. Features extraction principle: (1) Analysis window, (2) Textural feature extraction.

The choice of window's size is very important for the quality of the information extracted. In such a manner that choosing a size too small is likely to make us lose a lot of textural information, and the choice of a too large size will reduce border location accuracy between the different regions in the image.

2.1 Haralick Features

The gray co-occurrence matrix is a statistical method of examining texture proposed by Haralick [1-2], unlike the first order methods, GLCM is a second order method that considers the relationship between the values of neighboring pixels. This method is one of the most used feature extraction methods in many types of texture analysis applications. It has great implementation simplicity and it provides good results for most types of images.

A Gray level co-occurrence matrix $P_{(dx,dy)}(i,j)$ is a square matrix, which represents the relative frequencies with which two pixels separated by a distance $d = d(dx,dy)$ on the image I, the first with gray-tone i and the second pixel with gray-tone j. Mathematically, we can define The GLCM as:

$$P_{(dx,dy)}(i,j) = \sum_{p=1}^{n} \sum_{q=1}^{m} \begin{cases} 1, if I(p,q) = i \; and \; I(p+dx, q+dy) = j \\ 0, otherwise \end{cases} \tag{1}$$

Choosing a short distance between two pixels generally gives good results, so in our study case, we chose the value of one pixel for the distance d.

The large size of information obtained by GLCM matrix makes treatment difficult. Thus, instead of directly using GLCM, we calculate some of the fourteen features defined by Haralick. According to the experimental results, in our work, we use the five least correlated Haralick coefficients for texture analysis: Homogeneity (f_1), Energy (f_2), Entropy (f_3), Contrast (f_4) and Correlation (f_5):

- The Homogeneity coefficient (f_1) returns a value that measures the closeness of the distribution of elements in the GLCM to the GLCM diagonal, the value of the homogeneity is high if the texture has more homogeneous regions:

$$f_1 = \sum_{i,j} \frac{P(i,j)}{1+|i-j|} \tag{2}$$

- The Energy coefficient (f_2) returns the sum of squared elements in the GLCM, it reaches high values when the distribution of the gray levels is either constant or in a periodic form:

$$f_2 = \sum_{i,j} P(i,j)^2 \tag{3}$$

- The Entropy coefficient (f_3) measures the complexity of the image, it achieves high values when the values of the co-occurrence matrix are all almost equal, and produces a low value when the texture is completely random:

$$f_3 = \sum_{i,j} P(i,j). \ln\,[P(i,j)] \tag{4}$$

- The Contrast coefficient (f_4) measures the local variations, if these variations are important, the contrast will reach high values:

$$f_4 = \sum_{i,j} (i-j)^2 . P(i,j) \tag{5}$$

- The Correlation coefficient (f_5) measures the linear dependence of gray level values in the GLCM, the value of this coefficient is high when the values are more uniformly distributed in the co-occurrence matrix:

$$f_5 = \sum_{i,j} \frac{(i-\mu x)(j-\mu y).P(i,j)}{\sigma x.\sigma y} \tag{6}$$

The process of extracting Haralick features from an image I is defined by the following algorithm:

1. Browsing the image for each pixel by a sliding window MW of size $k\times k$, the current pixel $I_{(i,j)}$ is the center pixel of $MW_{(i,j)}$.
2. Computing the four directions $\theta \in (0°, 45°, 90°, 135°)$ GLCM matrices for the current window $MW_{(i,j)}$.
3. Extraction the five Haralick features for each Direction GLCM:

$$X_{q\theta} = \{f_{1\theta}, f_{2\theta}, f_{3\theta}, f_{4\theta}, f_{5\theta}\} \tag{7}$$

4. Calculation the mean of these features, so each pixel $I(i,j)$ have a vector X_q:

$$X_q = (X_{q0°} + X_{q45°} + X_{q90°} + X_{q135°})/4 \tag{8}$$

2.2 Fractal Features

Fractal geometry appeared in the 70s, it has proposed new concepts in order to understand some complex phenomena. The concept of Fractal has many fields of application including image analysis.

During image analysis application, the fractal geometry is in most cases used through the concept of fractal dimension (FD). Many methods exist to estimate this dimension, in our study, we have chosen to work with the differential box counting method, as it can be computed automatically and can be applied to patterns either with or without self-similarity.

The differential box counting method [3-4] consist in partitioning the image space into boxes of different sizes, then the probability $N(r)$ is calculated as the difference between the maximum and minimum gray levels for each box. Then, the fractal dimension is estimated using the equation:

$$FD = \lim_{r\to 0} \frac{\ln{[N(r)]}}{\ln{(^1/_r)}} \tag{9}$$

To compute the fractal dimension of a pixel $I(i,j)$ for image I, we use the local $m\times m$ pixel window $W(i,j)$:

1. For various $r \in [0,1]$.
    a. Divide $W_{(i,j)}$ into $(1/r)^2$ boxes.
    b. Divide the range of intensities $[0,255]$ into $1/r$ levels numbered $[1,1/r]$.

```
        c. For each box $b(p,q) \in W_{(i,j)}$ do:
              i. $l = minimum(b(p,q))$
             ii. $k = maximum(b(p,q))$
            iii. $n_{p,q}(r) = l - k + 1$
        d. $N(r) = \sum_{p,q} n_{p,q}(r)$
2.      Do line-fit of $N(r)$ and ln $(1/r)$.
3.      The fractal dimension is obtained by linear regression of this line-fit.
```

In this paper, we use the differential box counting method as a manner to extract different features from the textural image. We have used not only the original image (I_1) but also derived images:

- high gray valued image (I_2):

$$I_2(i,j) = \begin{cases} I_1(i,j) - L_1, if\ I_1(i,j) > L_1 \\ 0, otherwise \end{cases} \tag{10}$$

- low gray valued image (I_3):

$$I_3(i,j) = \begin{cases} 255 - L_2, if\ I_1(i,j) > (255 - L_2) \\ I_1(i,j), otherwise \end{cases} \tag{11}$$

- horizontally smoothed image (I_4):

$$I_4(i,j) = \frac{1}{2\omega+1} \sum_{k=-\omega}^{\omega} I_1(i, j+k) \tag{12}$$

- vertically smoothed image (I_5):

$$I_5(i,j) = \frac{1}{2\omega+1} \sum_{k=-\omega}^{\omega} I_1(i+k, j) \tag{13}$$

Finally, we have for each pixel $I(i,j)$ a vector $X_q = \{f_1, f_2, f_3, f_4, f_5\}$ where f_k is the fractal dimension of the sliding window $W_{k(i,j)}$ from I_k.

2.3 Wavelet Features

The Wavelet Transform is a powerful tool widely used in various image processing fields, the main advantage of this transform is that unlike the Fourier transform, the wavelets are localized in both time and frequency.

The input signal can be decomposed using a series of elementary wavelet functions $\Psi_{a,b}(t)$, created from a translation and dilatation of a mother wavelet Ψ.

$$C_{a,b} = \int_{-\infty}^{+\infty} x(t).\Psi_{a,b} dt \tag{14}$$

According to Grossman and Morlet previous works [5] the wavelet coefficients $C_{a,b}$ resulting from this transform contain information about the signal $x(t)$ studied at different scales.

$$\Psi_{a,b}(t) = \frac{1}{\sqrt{a}} \Psi\left(\frac{t-b}{a}\right) with\ a \neq 0 \tag{15}$$

To facilitate the implementation of the transformed discrete wavelet (DWT), Mallat produced a fast wavelet decomposition and reconstruction algorithm based on the use of a filter bank, including a low pass filter (LP) and a high pass filter (HP) [6]. The input signal undergoes a high pass filter and a low-pass filter, the filtering resumes on each sub-band after a sub-sampling operation, as shown in Figure 2.

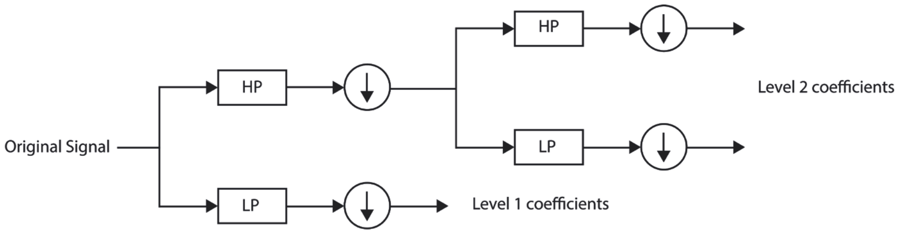

Fig. 2. Wavelet decomposition algorithm.

Wavelet decomposition can be extended to superior dimensions, including the two-dimensional signal, in this case, the DWT is applied first line by line, then column by column. Four images are then generated at each level. The Figure 3 shows an example of decomposition of the image on three levels.

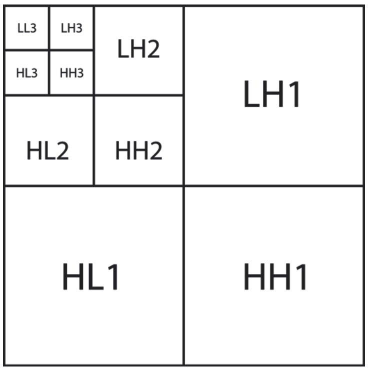

Fig. 3. Channels resulting from the decomposition of 3 levels.

In what concerns the extraction of wavelet attributes of each pixel of our image, the DWT algorithm is on an analysis window representing the local texture [7]. For a decomposition of J levels, we end up with $3J + 1$ features channel that will result in a features vector Y for each pixel:

$$Y = \{a_j, (d_k^1, d_k^2, d_k^3), k = 1 \ldots J\} \tag{16}$$

To minimize the size of the vector Y, we will replace each features channel of each scale by its energy using the following equation:

$$Energy(channel, scale) = \frac{1}{n} \sum_{x,y \in R} d_{x,y}^2(channel) \tag{17}$$

Ultimately, we have for each pixel of the image a vector X_q which characterizes the local texture of the vicinity of the latter:

$$X_q = (e_1, e_2, \ldots, e_{3J+1}) \tag{18}$$

3 Segmentation

After extraction of the texture attributes of our image, we use our unsupervised classification approach to classify our cloud observation made of the extracted features. Our segmentation approach is composed of four main modules: in the first module we use the textural feature vectors in the learning phase of Kohonen Self-Organizing feature map, then in the second module we represent this map using the underlying probability density function (PDF), in the third module we apply the watershed transformation to identify homogeneous regions in the PDF representation, finally the segmentation of the image is the last module which is done by assigning each pixel to an extracted homogeneous region, as shown in Figure 4.

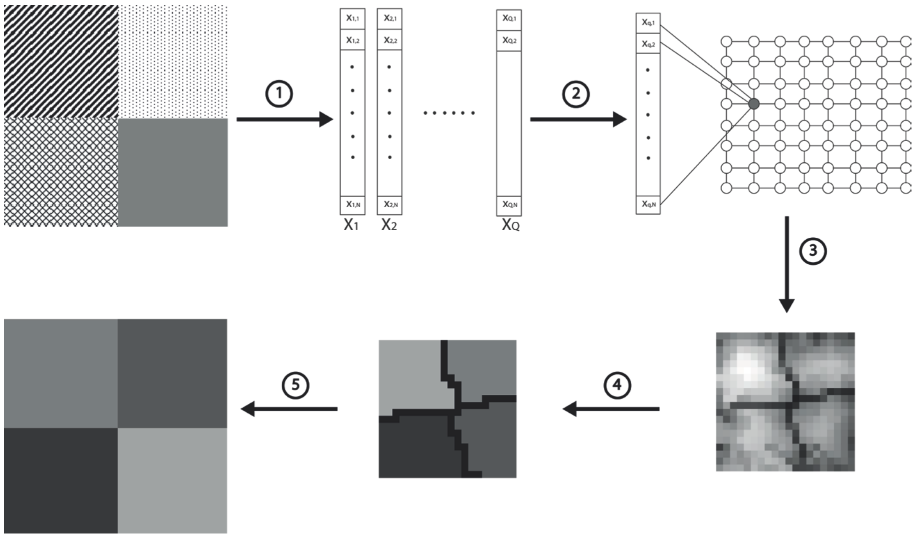

Fig. 4. Segmentation modules: (1) Features extraction, (2) Training the Kohonen map, (3) Representing the Kohonen map by the PDF, (4) Application of the watershed transformation to extract the homogeneous regions, (5) Segmentation of the image.

3.1 Kohonen map

In 1984 Kohonen designed a neural network known as the self-organizing Kohonen map [8]. Using a learning rule similar to the cortex interaction laws, he was able to show the same phenomenon of self-organization. The Kohonen network has been successfully applied in various fields, such as speech recognition, robotics or image processing.

The architecture of the Kohonen network differs from that of competitive learning network in the organization of neurons in the output layer.

Kohonen takes into account in its modeling, the physical position of the neurons within the network. Furthermore, it defines different types of topological organization of neurons within a layer. Thus, the neurons are ordered according to a grid on a map. Each neuron is associated with a couple of indexes position on the grid, as shown in Figure 5.

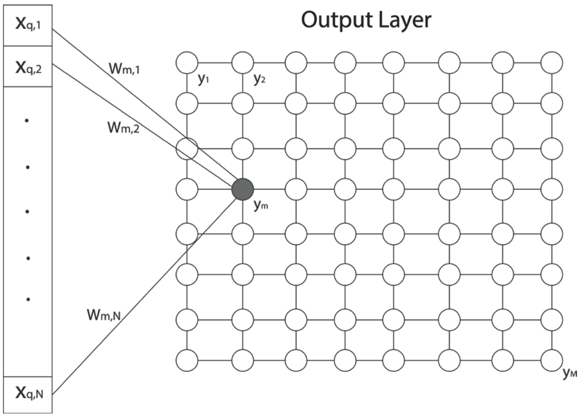

Fig. 5. Kohonen map layers: Input Layer (Left), Output Layer (Right).

Let $\Gamma = \{X_1, X_2, \dots\dots, X_Q\}$ be a sample of Q observations in a N-dimensional space such as $X_q = [x_{q,1}, x_{q,2}, \dots\dots, x_{q,N}]^T, q = 1,2, \dots, Q$. The Kohonen network is made of two layers, the first one is the input layer which is composed of N attributes of the observation X_q. The output layer is composed of M neural units regularly distributed on the map which elaborates prototypes of the data.

The neural units of the first layer are connected to the units of the second layer. Each interconnection from an input unit j to an output unit m has a weight $W_{m,j}$. That means that each output unit m has a corresponding weight vector $W_m = [W_{m,1}, W_{m,2}, \dots\dots, W_{m,N}]^T$.

The followed steps of the learning algorithm are:

1. Initializing the weights of the neurons in the Kohonen map layer by giving them small random values.
2. Presenting an input vector X_q.
3. Finding the winning node m^* using the Euclidean distance between the vector X_q and the nodes of the output layer.
4. Updating the weights W_i winner node, as well as those around him, using the equation 19.

$$\begin{cases} W_m(t) = W_m(t-1) + \alpha(t).\left[X_q - W_m(t-1)\right]\ if\ m = m^* \\ W_m(t) = W_m(t-1) + \alpha(t).h_m(t).\left[X_q - W_m(t-1)\right]\ if\ m \in V(m^*, r(t)) \end{cases} \tag{19}$$

5. Decreasing the size of the neighborhood area winners' nodes.

6. Decreasing the learning coefficient $\alpha(t)$.
7. Going back to Step 2., or else complete learning.

3.2 Probability density function

Once the learning phase is processed, the determined weight vectors in the multidimensional data space are used to estimate the underlying probability density function (PDF) [9]. This technique of representation is based on the calculation of distances between the weight vectors. It helps to distinguish classes relatively distant from one to another, when two classes are very close, the distances calculated at the borders of these classes are relatively small and become undetectable.

To view the PDF, we use the non-parametric Parzen estimator, defined by:

$$p(W_m) = \frac{1}{Q} . \sum_{q=1}^{Q} \frac{1}{V[D(W_m)]} \Omega(\frac{W_m - X_q}{h_Q}) \tag{20}$$

This visualization permits to display the Kohonen map as a digital image where each unit of the map is represented by a gray value pixel, however, the parameter h_0 has a great effect on the quality of the estimation. If it is too big, then small maximum density will be undetectable. Conversely, if too small, we obtain an estimator with a lot of noise maxima. In Figure 6 we can observe that the map is constituted by four regions where the PDF presents high values, separated by valleys where the PDF presents low values. This figure shows the importance of h_0 value.

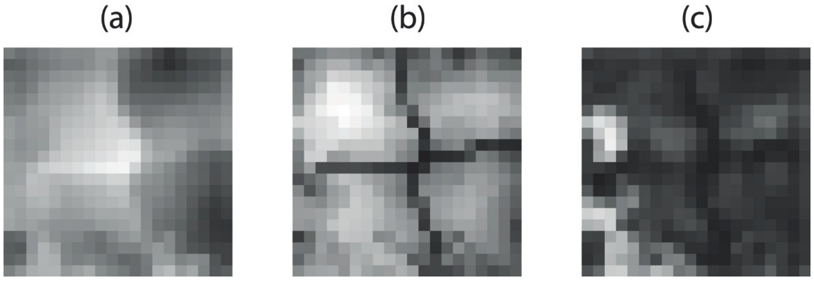

Fig. 6. PDF estimation with different h_0 = {0.02 (a), 0.005 (b), 0.001 (c)}.

3.3 Watershed transformation

Although the PDF presentation allows us to reduce the size of the set of multidimensional observations, this method cannot be considered automatic because of the intervention of the analyst to decide the number of classes presented in the sample and do so in an interactive classification. So, to automate this step, we use digital mathematical morphology and specifically the watershed technique [10-12].

To detect modal regions of the PDF, as a first step, we apply a numerical morphological opening on this estimation, this transformation tends to remove the tips of the peaks of the PDF, while preserving the valleys that separate the modal regions. This opening operation is therefore interesting to highlight the significant regional maxima of the PDF.

Let $p(x)$ be the filtered PDF function. The proposed procedure allows determining the so-called catchment basins corresponding to the regional minima of the inverse function $-p(x)$, through consecutive homotopic thinning operations followed by sequential pruning operations until idempotence [12].

Figure 7 shows a watershed technique on a PDF presentation containing four different modal regions.

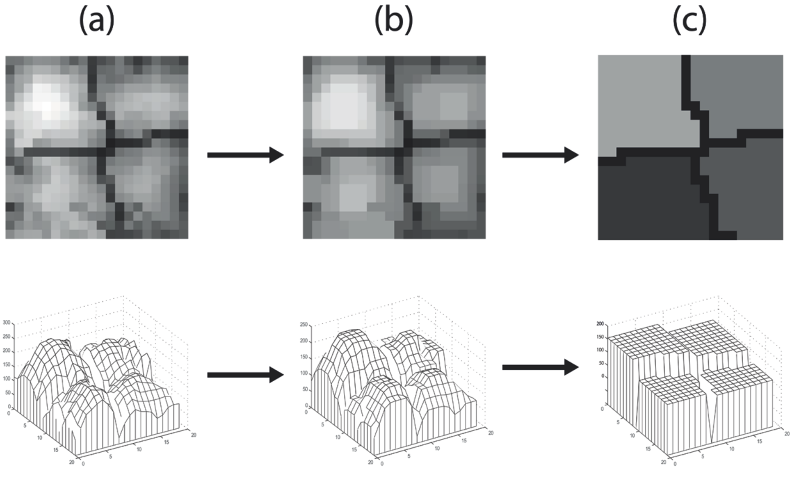

Fig. 7. Watershed transformation: (a) PDF, (b) opening PDF, (c) PDF modal regions.

3.4 Segmentation process

The characteristics of the resulting function $p'(x)$ from our watershed technique lie in the fact that the modal regions of this function take constant values, which equals to the regional maximum of each region. Besides that, the thickness of the line separating two neighboring modal Regions does not exceed one pixel. These two properties will allow us to extract the modal regions, by means of a simple numerical dilation of the function $p'(x)$.

After extracting the modal regions of our image, we compare each pixel of the image, which is represented by its vector of textural features, with all the cells in the Kohonen map through the Euclidean distance. Finally, we assign to our pixel the value of the closest modal region.

4 Experimental Results and Evaluations

Our segmentation approach is tested on three images generated by combining several different textures, as shown in Figure 8. The three images have a size of 256×256 pixels and 256 gray levels. The first image is composed of four different synthetic textures created by an image manipulation software, the second image is a combination of two natural textures from the Brodatz album [13], combined by using an irregular shape, the last image is composed of five Brodatz textures of different types. All these textures had their histograms equalized and are therefore indistinguishable on the basis of only their gray level or their first-order statistics.

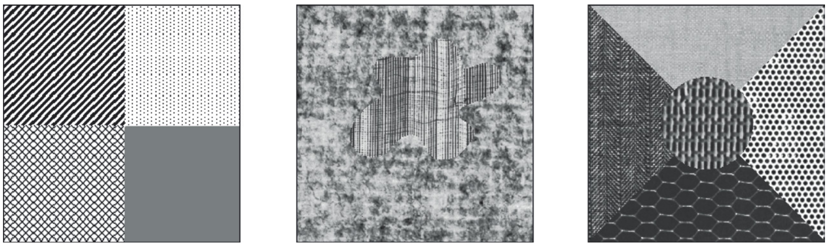

Fig. 8. Test images: Image 1 (Left image), Image 2 (Middle Image), Image 3 (Right Image).

To extract Haralick five features of these images, we looked for an optimal size of analysis window, finally, we opted for a size of 7×7 pixels, it was the same for the other two extraction methods. We find that the 11×11 pixel size is optimal to extract five fractal features, while the size of 7×7 has given the best outcome for wavelet features. To the latter method, Daubechies wavelets [14] have given a better result for segmentation and optimal results in terms of speed, which push us to use this wavelet family.

After extracting the vector features of each image, we classified them using our approach of segmentation. Figure 9 shows us the visual results of the segmentation of our three textured images, using the three features extraction methods.

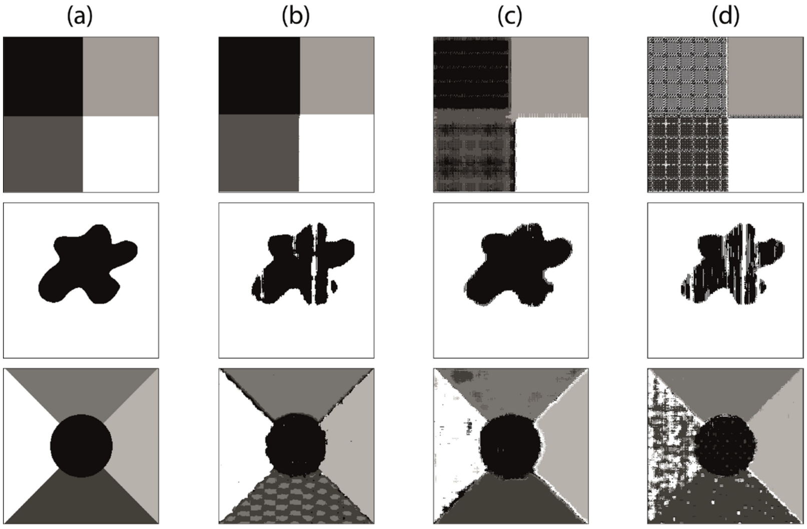

Fig. 9. Visual results: (a) Ground truth images, (b) Haralick results, (c) Fractal results, (d) Wavelet results.

To correctly evaluate the segmentation results, our original images were manually segmented to get the labeled ground truth images, then they were compared with our results to have a segmentation error rate. In addition, we made a comparison between our result and segmentation's result obtained by k-means clustering method [15].

All the modules that compose our segmentation approach were programmed using the C++ library OPENCV2, all the results are obtained using one i5-2540M 2.60GHz computer with a 4.00GB of RAM.

Table 1 presents all the segmentation results obtained for our three images, and the computing time of each extraction approach.

Table 1. Segmentation Results

Image	Extraction method	Extraction time	K-means Approach %Error Rate	Our Approach %Error Rate
Image 1	Haralick	25m 55s	02.75	02.20
	Fractal	01m 15s	13.27	13.04
	Wavelet	01m 48s	23.47	17.07
Image 2	Haralick	25m 30s	02.77	02.80
	Fractal	01m 11s	01.92	01.54
	Wavelet	01m 50s	05.78	03.45
Image 3	Haralick	26m 07s	11.54	10.14
	Fractal	01m 17s	07.64	07.12
	Wavelet	01m 14s	12.05	14.20

The results of the first image show us that the method of Haralick is more effective for synthetic images than the other methods, however, in terms of classification rate, the second and the third image results prove that fractals attributes are more suitable for the case of images with natural textures.

One of the major flaws of the method based on the Haralick features is the large extraction time, being based on the calculation of the co-occurrence matrix for each pixel in the image, it requires more computational resources than the other methods; furthermore, according to the results of segmentation of the third image, it can be seen that unlike the fractal method, Haralick attributes are not well suited for macro textures.

As opposed to k-means approach who needs to fix a number of classes k, our segmentation approach allows us to segment the image without knowing in advance the number of regions of the image, thanks to its combination of networks Kohonen neuron and mathematical morphology. Besides this advantage, the results prove that our approach it has a more important result for classification rate than the k-means approach.

5 Conclusion

In this paper, we present a comparative study of three different texture feature extraction methods, using our textural images segmentation approach, based on the application of morphological watershed transformation in Kohonen map.

The objective of this study is to characterize the textures presented in the textural images using these three extraction methods, the first based on GLCM matrix, the second is to use the local fractal dimension, while the last method is based on wavelet features. The application of these three methods shows us that the fractal features are most effective for the natural textures, in addition to that, it does not consume a great amount of computing time.

In comparison with the k-means approach, the classification rate results obtained by our segmentation approach is very encouraging. The thing which allows us to have a perspective to apply to both color textural image and 3D image.

6 References

[1] Haralick, R. M. (1979). Statistical and structural approaches to texture. Proceedings of the IEEE, 67(5):786-804. https://doi.org/10.1109/PROC.1979.11328

[2] Haralick, R. M., & Shanmugam, K. (1973). Textural features for image classification. IEEE Transactions on systems, man, and cybernetics, (6):610-621. https://doi.org/10.1109/TSMC.1973.4309314

[3] Chaudhuri, B. B., & Sarkar, N. (1995). Texture segmentation using fractal dimension. IEEE transactions on pattern analysis and machine intelligence, 17(1):72-77. https://doi.org/10.1109/34.368149

[4] Sarkar, N., & Chaudhuri, B. B. (1994). An efficient differential box-counting approach to compute fractal dimension of image. IEEE Transactions on systems, man, and cybernetics, 24(1):115-120. https://doi.org/10.1109/21.259692

[5] Grossmann, A., & Morlet, J. (1984). Decomposition of Hardy functions into square integrable wavelets of constant shape. SIAM journal on mathematical analysis, 15(4):723-736. https://doi.org/10.1137/0515056

[6] Mallat, S. G. (1989). A theory for multiresolution signal decomposition: the wavelet representation. IEEE transactions on pattern analysis and machine intelligence, 11(7):674-693. https://doi.org/10.1109/34.192463

[7] Hela, S. M., & Kamel, H. (2005, March). Segmentation d'images texturées par transformée en ondelettes et classification C-moyenne floue. In International conference: Sciences of Electronic, Technologies of information and Telecommunications SETIT.

[8] Kohonen, T. (2012). Self-organization and associative memory (Vol. 8). Springer Science & Business Media.

[9] Parzen, E. (1962). On estimation of a probability density function and mode. The annals of mathematical statistics, 33(3):1065-1076. https://doi.org/10.1214/aoms/1177704472

[10] Alaoui, M. T., & Sbihi, A. (2013, September). Texture classification based on co-occurrence matrix and neuro-morphological approach. In International Conference on Image Analysis and Processing (pp. 510-521). Springer, Berlin, Heidelberg.

[11] Beucher, S. (1993). Segmentation tools in mathematical morphology. In Handbook of Pattern Recognition and Computer Vision (pp. 443-456). https://doi.org/10.1142/9789814343138_0015

[12] Talibi-Alaoui, M., & Sbihi, A. (2012). Application of a mathematical morphological process and neural network for unsupervised texture image classification with fractal features. IAENG International Journal of Computer Science, 39(3):286-294.
[13] Brodatz, P. (1966). Textures: a photographic album for artists and designers. Dover Pubns.
[14] Daubechies, I. (1992). Ten lectures on wavelets. Society for industrial and applied mathematics. https://doi.org/10.1137/1.9781611970104
[15] Lloyd, S. (1982). Least squares quantization in PCM. IEEE transactions on information theory, 28(2):129-137. https://doi.org/10.1109/TIT.1982.1056489

7 Authors

Khalid Salhi received a Master's degree in computer science in 2012 from faculty of science, Oujda, Morocco. He is a Ph.D. student who work on classification and image processing in LaRI laboratory of the faculty of science, Oujda, Morocco. His fields of research interest are in image clustering based on neural networks.

El Miloud Jaara is a Professor at the Department of Mathematics and Computer Science, Faculty of Science, Mohammed First University, Oujda, Morocco. He earned his PhD in parallelization of neural networks in 2000 from University Mohammed Premier, Oujda. His research interests include the parallelization, the neural networks, image processing.

Mohammed Talibi Alaoui has obtained his PH Degree in automatic and information processing from Ibn Tofail University, Kenitra, Morocco. He obtained the Habilitation degree in 2014 from the Mohammed First University, Oujda (Morocco). He is currently professor at the Sidi Mohammed Ben Abdellah University of Fes (Morocco). He teaches the computer science. His field of research interest is in image processing, computer vision and their applications to the medical imaging, quality control and pattern recognition.

Article submitted 24 December 2017. Resubmitted 25 January 2018. Final acceptance 23 February 2018. Final version as submitted by the authors.