Pattern Recognition 103 (2020) 107189

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/patcog

Fusion of complex networks and randomized neural networks for texture analysis

Lucas C. Ribas[a,c], Jarbas Joaci de Mesquita Sá Junior[b], Leonardo F.S. Scabini[c], Odemir M. Bruno[a,c,*]

[a] Institute of Mathematics and Computer Science, University of São Paulo, Avenida Trabalhador São-Carlense, 400, Centro, São Carlos, SP 13566-590, Brazil
[b] Curso de Engenharia da Computação, Programa de Pós-Graduação em Engenharia Elétrica e de Computação, Campus de Sobral, Universidade Federal do Ceará, Rua Coronel Estanislau Frota, 563, Centro, Sobral, CE, CEP 62010-560, Brasil
[c] São Carlos Institute of Physics, University of São Paulo, PO Box 369, São Carlos, SP 13560-970, Brazil

ARTICLE INFO

Article history:
Received 31 January 2019
Revised 13 November 2019
Accepted 24 December 2019
Available online 2 January 2020

Keywords:
Randomized neural networks
Complex networks
Texture analysis
Feature extraction

ABSTRACT

This paper presents a high discriminative texture analysis method based on the fusion of complex networks and randomized neural networks. In this approach, the input image is modeled as a complex network and its topological properties as well as the image pixels are used to train randomized neural networks to create a signature that represents the deep characteristics of the texture. The results obtained surpassed the accuracy of many methods available in the literature. This performance demonstrates that our proposed approach opens a promising source of research, which consists of exploring the synergy of neural networks and complex networks in the texture analysis field.

1. Introduction

Most of computer vision applications consider texture as a key factor to image discrimination, thus texture analysis has been a constant research field since the 1960s. The texture is a visual pattern related to the object surface, which is represented in an image by the pixel spatial organization. However, the interpretation of texture is ambiguous, thus there is no formal definition for the term that is widely accepted by the scientific community. This resulted in an extensive and heterogeneous literature of texture analysis methods proposed throughout the years [9,31,32,48], which are employed in several areas that benefit from image recognition systems.

Classical texture analysis techniques can be grouped into four different approaches: statistical, spectral, structural, model-based methods [54]. The earlier and most diffused methods are statistical-based, such as variants of gray-level co-occurrence matrices (GLCM) [24], local binary patterns (LBP) [26,38] and local ternary patterns [16]. Spectral methods explore texture in the frequency domain, and some examples are Gabor filters [34] and wavelet transforms [15]. On the other hand, structural methods consider texture as a combination of smaller elements, called textons, that compose the overall texture as a spatially organized pattern. A common approach to this kind of analysis is the Morphological decomposition [29]. Finally, model-based methods represent textures through sophisticated mathematical models and the estimation of their parameters. Common methods of this category include Fractal models [2,46,47] and stochastic models [40].

Besides classical methods, more recent and innovative techniques are addressing texture differently, achieving promising results. An example is the set of techniques that use learning [55], such as descriptors based on a vocabulary of scale invariant feature transform (SIFT) [11], often called bag-of-visual-words (BOVW). Methods based on image complexity analysis are also gaining attention such as cellular automata [13] and complex networks (CN) [3,8,50]. In particular, methods based on the CN theory have achieved promising results due to their capacity to represent the relationships among structural elements of texture. However, the problem of how to achieve more satisfactory modeling (i.e., a lesser number of parameters) and new ways of characterizing the network remains a challenge to overcome.

In this paper, we propose a novel approach that combines complex networks and randomized neural networks (RNN) to obtain a texture signature. Complex networks are attracting increasing attention due to their flexibility and generality for representing many real-world systems, including texture images. On the other hand, a

* Corresponding author.
E-mail addresses: lucasribas@usp.br (L.C. Ribas), jarbas_joaci@yahoo.com.br (J.J.d.M. Sá Junior), scabini@ifsc.usp.br (L.F.S. Scabini), bruno@ifsc.usp.br (O.M. Bruno).

https://doi.org/10.1016/j.patcog.2019.107189
0031-3203/

randomized neural network is a type of neural network that has a very fast learning algorithm, which has been used in many pattern recognition tasks. Here we first model the texture image as a directed network, representing the information about the pixels and their neighbors as vertices and edges. To characterize the texture, the topological properties from the modeled network and the image pixels are used to train a randomized neural network, and the set of output weights is used as a feature vector that represents discriminative characteristics of the texture. Experimental results on four databases demonstrated a higher performance of the proposed method when compared to other methods of the literature.

The remainder of this paper is organized as follows. Section 2 describes the fundamentals of complex networks and randomized neural networks. A novel method for texture classification based on the fusion of complex networks and randomized neural networks is presented in Section 3. Section 4 describes the databases and experiments performed to evaluate the proposed method. The discussion about the results achieved and comparisons are presented in Section 5. Finally, in Section 6, we conclude the work with some remarks.

2. Background

2.1. Complex networks

Almost any natural phenomena can be modeled as networks by defining a set of entities and establishing a criterion of the relation between them. Some classical examples are the internet, composed of various connected computers and routers, and a network of a cell, describing chemicals connected by chemical reactions. From this perspective emerges the study of complex networks, or network science [4]. Network science is strongly based on graph theory. In the last decades, works have shown patterns present in many networks or graphs, which were then understood as a structural characteristic of some models such as the scale-free and the small-world. These findings have caused increasing interest from the scientific community on the study of complex networks, creating a new multidisciplinary research field.

The theoretical foundations of this area arise from the intersection of graph theory, physics, mathematics, statistics, and computer science. Therefore, CN has been employed as a powerful tool for pattern recognition [35], where natural systems of many areas are modeled as networks and then quantified through their topological structure. CN applications are found in various areas of science, such as physics, social sciences, biology, mathematics, ecology, medicine, computer science, linguistics, neuroscience, among others [12].

Formally, a network or graph G is described by a tuple of vertices and edges (V, E). Let v_i be a vertex of the set $V = \{v_1, \ldots, v_n\}$. An edge e_{v_i,v_j} represents a connection between two vertices v_i and v_j, so the set $E = \{e_{v_i,v_j}, \ldots\}$ is composed of all edges connecting vertices of V. The network can also be directed (in this case, the edges e_{v_i,v_j} have a direction from v_i to v_j). In most of the CN applications, the first step is to define how to model the target problem as a network, thus defining what are the vertices and what are the edges. Once G is properly built, many measures can be computed to quantify its structure, varying from centrality, path-based measures, community structure, etc. Moreover, the structure of a real network is the result of the continuous evolution of the forces that formed it, and certainly affects the function of the system [5]. Therefore, the network dynamics can be analyzed by the characterization of its structural evolution in function of time or some modeling parameter.

2.2. Randomized neural networks

Randomized neural networks (RNN) [25,41,42,51] are neural networks that, in their simplest version, are composed of two neuron layers (hidden and output layer), each one with a different role in the regression/classification task. The hidden layer has its neural weights determined randomly according to a probability distribution (for instance, a uniform or normal distribution). Its purpose is to project non-linearly the input data in another dimensional space where it is more likely that the feature vectors are linearly separable, as stated in Cover's theorem [10]. In turn, the output layer aims to linearly separate these projected feature vectors using the least-squares method.

To mathematically explain the details of the RNN used in this work, let $X = [\vec{x_1}, \vec{x_2}, \ldots, \vec{x_N}]$ be a matrix of input feature vectors (including -1 for bias weight) and $D = [\vec{d_1}, \vec{d_2}, \ldots, \vec{d_N}]$ be the corresponding labels. Thus, once the training set is established, the first step to create the RNN is to build the matrix of hidden neuron weights W of dimensions $Q \times (p+1)$, where Q and p are the number of hidden neurons and the number of attributes in each input feature vector, respectively.

Next, the output of the hidden layer for all the feature vectors $\vec{x_i}$ $(i \in 1, \ldots, N)$ can be obtained by $Z = \phi(WX)$, where $\phi(.)$ is generally a sigmoid or hyperbolic tangent function. This matrix of projected vectors $Z = [\vec{z_1}, \vec{z_2}, \ldots, \vec{z_N}]$ (including -1 for bias weight), in turn, can be used to compute the output neuron weights, according to the following equation

$$M = DZ^T(ZZ^T)^{-1}. \tag{1}$$

where $Z^T(ZZ^T)^{-1}$ is the Moore–Penrose pseudo-inverse [36,44].

Also, the matrix ZZ^T sometimes becomes close to singular, which may result in an inaccurate inverse in Eq. 1. To avoid this drawback, it is possible to apply the Tikhonov regularization [7,52], according to

$$M = DZ^T(ZZ^T + \lambda I)^{-1} \tag{2}$$

where λ and I are the regularization parameter and identity matrix, respectively.

3. Proposed method

In this section, we describe the proposed method that combines a new texture modeling in complex networks and randomized neural networks for texture characterization.

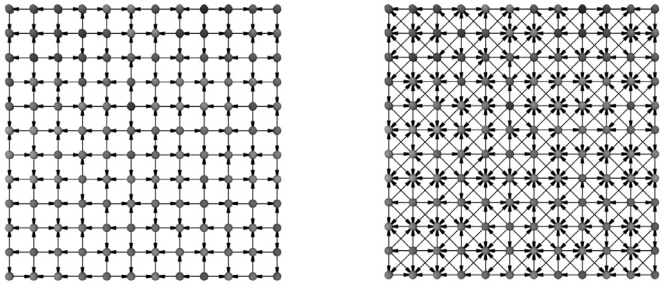
3.1. Modeling texture as directed CN

Let I be an image composed of pixels i, which have as Cartesian coordinates x_i and y_i. In gray-scale images, each pixel has an intensity represented by an integer value $I(i) \in [0, L]$, where L is the highest gray-level value. To model a texture image as a directed network, each pixel i is mapped as a vertex $v_i \in V$ of a network R. The set of edges E is built connecting two vertices v_i and v_j, which represent two pixels i and j, by a directed edge from v_i to v_j, $e_{v_i,v_j} \in E$, if the Euclidean distance between them is less than or equal to a radius r and $I(i) < I(j)$, according to

$$E = \left\{ e_{v_i,v_j} \in E \mid dist(v_i, v_j) \leqslant r \wedge I(i) < I(j) \right\}, \tag{3}$$

where $dist(v_i, v_j) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ is the Euclidean distance between two pixels. Each edge has a weight $w(e_{v_i,v_j})$ defined as

$$w(e_{v_i,v_j}) = \begin{cases} \dfrac{|\, I(i) - I(j)\,|}{L}, & \text{If } r = 1 \\ \dfrac{\left(\frac{dist(v_i,v_j)-1}{r-1}\right) + \left(\frac{|I(i)-I(j)|}{L}\right)}{2}, & \text{Otherwise.} \end{cases} \tag{4}$$



(a) $r = 1$ (b) $r = 2$

Fig. 1. Directed complex network for two different radii. Each pixel represents a vertex and two vertices are connected by a directed edged pointing towards the pixel (vertex) with the higher gray level.

It is worth mentioning that the direction of the edges is determined by the pixel gray-levels. In other words, an edge points to the vertex that represents a pixel with greater intensity. If both intensities are equal, the edge is bidirectional. It is also important to stress that the value r is the unique modeling parameter and determines the size of the neighborhood of each vertex. Thus, as r increases, the reach of connection and the degree of vertices increase as well. The analysis of this evolution is an interesting way of characterizing these networks. Fig. 1 shows the modeling of a directed network for different values of r.

3.2. Proposed signature based on RNN

Our method aims to use as texture signature the weights of the output layer of the RNN trained with information from the modeled complex networks. To this end, three sources of information are considered for each vertex: out-degree, weighted out-degree, and weighted in-degree. As the out-degree is directly related to the in-degree in the modeled networks (i.e. the sum of these two degrees is equal in all the vertices) and, therefore, provide the same information, we considered only the out-degree.

The out-degree k_{v_i} of a vertex v_i represents the number of out-edges connected to other vertices,

$$k_{v_i} = \sum_{v_j \in V} \begin{cases} 1, & e_{v_i,v_j} \in E \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

On the other hand, the weighted out-degree ks_{v_i} is given by the sum of the weights of the out-degree edges of a vertex v_i,

$$ks_{v_i} = \sum_{v_j \in V} \begin{cases} w(e_{v_i,v_j}), & e_{v_i,v_j} \in E \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

Finally, the weighted in-degree ke_{v_i} is defined as the sum of the weights of the in-degree edges in v_i,

$$ke_{v_i} = \sum_{v_j \in V} \begin{cases} w(e_{v_j,v_i}), & e_{v_j,v_i} \in E \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

To build a matrix of input vector for the RNN, we adopted a strategy of analysis of the evolution of the complex network for different values of the modeling parameter r. In this way, the input feature vector and the corresponding label of a vertex v_i are built according to the following procedure: the gray-scale intensity of the pixel is considered as an output label $d_{v_i} = I(i)$ and the values of out-degree of the vertex for different values of the modeling parameter r are attributes of the input feature vector $\vec{x}_{v_i} = [k^1_{v_i}, k^2_{v_i}, \ldots, k^R_{v_i}]$, where R is the maximum value of the modeling parameter. A matrix of input feature vectors $X_{(k)}$ and a matrix of output labels D are then built for all the vertices of the complex network. Thus, it is possible to analyze the evolution of the topology of vertices that represent pixels that have a determined gray-scale intensity. Fig. 2(a) shows an example of how to build these matrices X and D. In addition to building the matrix of input feature vectors $X_{(k)}$ for the out-degree, we also built matrices of input vectors for the weighted out-degree $X_{(ks)}$ and for the weighted in-degree $X_{(ke)}$.

The next step is to define the weights of the matrix W of the hidden layer of the RNN. In general, these weights are determined randomly in each training stage. Nevertheless, because we want our method to provide the same signature for the same texture image, it is necessary to use the same values in the matrix W. Thus, we adopted the strategy proposed in [49] and used the Linear Congruent Generator (LCG) [30,43] to obtain pseudo-random uniform values for the matrix W, according to the following equation

$$V(n+1) = (a * V(n) + b) \bmod c, \tag{8}$$

where V is a random numeric sequence and a, b and c are tuning parameters. The sequence V has length $E = Q * (p+1)$, its first value is $V(1) = E + 1$, and the values of the parameters are $a = E + 2$, $b = E + 3$ and $c = E^2$ (values adopted in [49]). Hence, the matrix W is composed of the vector V divided into Q segments of length $p + 1$. Finally, all values of matrix W and each line of the matrix X are normalized using z-score (zero mean and unit variance).

The proposed texture signature is built based on the matrix M, which becomes a vector $\vec{f} = DZ^T(ZZ^T + \lambda I)^{-1}$, where $\lambda = 10^{-3}$ (Fig. 2(b)). Notice that $\vec{f}$ has length $Q + 1$ due to the bias weight. Thus, the first step is to concatenate the vectors $\vec{f}$ obtained from RNNs trained with the three matrices of input data $X_{(k)}$, $X_{(ks)}$, $X_{(ke)}$, according to

$$\vec{\Upsilon}(Q)_R = \left[\vec{f}_k, \vec{f}_{ks}, \vec{f}_{ke}\right], \tag{9}$$

where Q is the number of neurons of the hidden layer and R is the maximum radius for building the complex network.

The vector $\vec{\Upsilon}(Q)_R$ is built using a single value of Q and R. These two parameters influence the weights of the neural network and, therefore, provide different characteristics for different values. Thus, initially, we propose a vector $\vec{\Theta}(R)_{(Q_1,Q_2,Q_m)}$ that concatenates the vectors $\vec{\Upsilon}(Q)_R$ for different values of Q,

$$\vec{\Theta}(R)_{Q_1,Q_2,\ldots,Q_m} = \left[\vec{\Upsilon}(Q_1)_R, \vec{\Upsilon}(Q_2)_R, \ldots, \vec{\Upsilon}(Q_m)_R\right]. \tag{10}$$

Finally, we propose a feature vector $\vec{\Psi}(R_1, R_2)_{Q_1,Q_2,\ldots,Q_m}$ that concatenates the vector $\vec{\Theta}(R)_{Q_1,Q_2,\ldots,Q_m}$ for two values of R,

$$\vec{\Psi}(R_1, R_2)_{Q_1,Q_2,\ldots,Q_m} = \left[\vec{\Theta}(R_1)_{Q_1,Q_2,\ldots,Q_m}, \vec{\Theta}(R_2)_{Q_1,Q_2,\ldots,Q_m}\right]. \tag{11}$$

4. Experiments

In the experiments, the signature proposed in this work is extracted for each image of the databases. These signatures are extracted using the directed CN modeling and randomized neural networks, as described in Section 3. For comparison, we also extract the signatures using other texture analysis methods. After that, to validate our proposed method and compare it to other texture analysis methods of the literature, the signatures were classified using Linear Discriminant Analysis (LDA) [17,19]. This classifier was adopted due to its simplicity, which emphasizes the characteristics obtained by the methods. The leave-one-out cross-validation scheme was used. In this validation strategy, one signature extracted from a sample (image) of the database is used for testing and the signatures of the remaining images are used for training the LDA classifier model. This process runs N times (N is the number of samples), each time using a distinct sample for testing. The performance measure is the average accuracy of the N training/test trials. The gray-scale texture databases used as benchmark to evaluate our proposed method were:

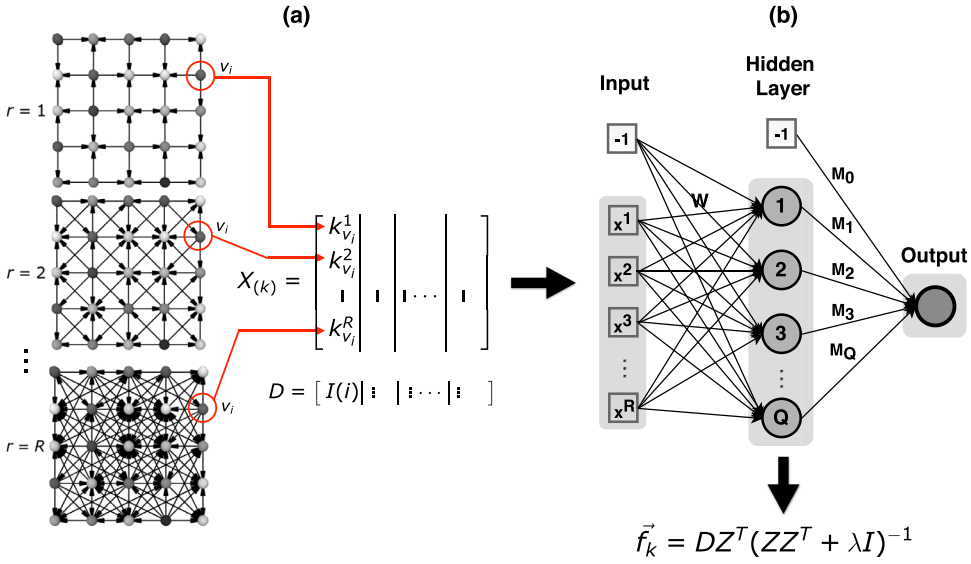


Fig. 2. Building of an input feature vector and corresponding output label for the out-degree k_{v_i} using different values of r to model the complex networks. (a) the out-degree k_{v_i} of the vertex v_i for different radii r is used to build the input vector and the gray scale of the pixel i is used as output label. (b) the randomized neural network is trained using $X_{(k)}$ and D, and the weights of the output layer are used as signature.

- Brodatz [6]: Just as in [3], 1776 texture images of 128 × 128 pixel size from this database divided into 111 classes were used in this work.
- Outex [37]: Just as in [3], the original 68 images 746 × 538 from TC_Outex_00013 were divided into 20 sub-images 128 × 128 pixel size without overlapping. Thus, the database used in this work has 1360 textures.
- USPTex [2]: This database has 2292 samples divided into 191 classes, 12 images per class, and each image has 128 × 128 pixel size.
- Vistex: The database *Vision Texture* is provided by the Vision and Modeling Group - MIT Media Lab [45]. Just as in [3], the original 54 images 512 × 512 were split into 16 sub-images 128 × 128 pixel size without overlapping. Thus, the database used in this work has 864 images.

The proposed method is applied to the aforementioned databases and the accuracy is compared to other methods of the literature. They are: Grey-Level Co-occurrence Matrix (GLCM) [23], Gray Level Difference Matrix (GLDM) [53], Fourier [53], Gabor Filters [14,34], Fractal [1], Fractal Fourier [18], Local Binary Patterns (LBP) [38], Local Binary Patterns Variance (LBPV) [22], Complete Local Binary Pattern (CLBP) [21], Local Phase Quantization (LPQ) [39], Local Configuration Pattern (LCP) [20], Local Frequency Descriptor (LFD) [33], Binarized Statistical Image Features (BSIF) [27], Adaptive Hybrid Pattern (AHP) [56], Complex Network Texture Descriptors (CNTD) [3] and ELM signature [49].

5. Results and discussion

5.1. Parameter evaluation

Fig. 3 shows the accuracies achieved on the four databases with the feature vector $\vec{\Upsilon}(Q)_R$ using $R = 4$. In this experiment, we used different values of $Q \in \{04, 09, 14, 19, 29, 39\}$, which were selected because they produce feature size multiple of five for each feature vector considered. As can be seen in the figure, the success rates increase as we increase the value of Q. This increase is followed by an increase in the number of features used. The best accuracies are obtained using $Q = 14$ on the Vistex database and $Q = 19$ on the other databases. These values of Q produce feature vectors

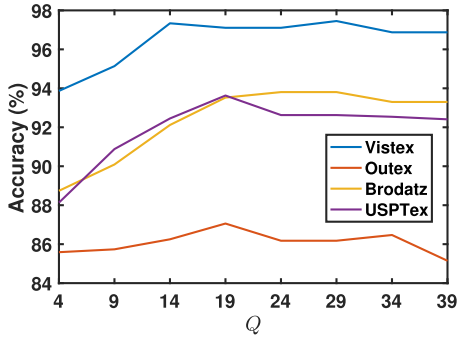
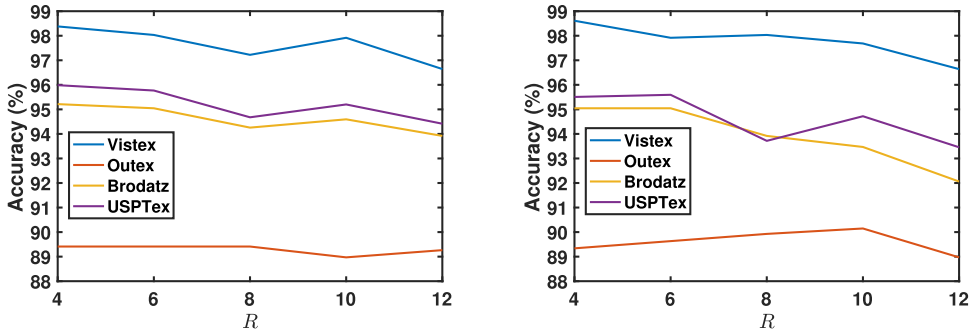


Fig. 3. Accuracies for the feature vector $\vec{\Upsilon}(Q)_R$ using different values of Q on the four databases.

of size 45 ($Q = 14$) and 60 ($Q = 19$). Furthermore, the success rate stabilizes when we use values of Q larger than $Q = 14$ on the Vistex database and values larger than $Q = 19$ on the other databases.

Table 1 shows the accuracies obtained on the four databases using the feature vector $\vec{\Theta}(R)_{Q_1,Q_2,...,Q_m}$ with $R = 4$. The results show that as the values of Q and their combinations increase (i.e. the number of features increases), the success rates increase as well. However, very large feature vectors do not assure the highest performance, once the success rates tend to stabilize at a determined value. For instance, if we compare the vector $\vec{\Theta}(04)_{19,29,39}$, which has 270 features, with the vector $\vec{\Theta}(04)_{04,09,14}$, which has 90 attributes, the former has a lower performance in all the databases, except for Brodatz. This suggests that the proposed signature reaches its limit in terms of discrimination. Thus, we considered the vectors $\vec{\Theta}(04)_{04,09,14}$ and $\vec{\Theta}(04)_{04,14,19}$, since they presented a good trade-off between high accuracy and a small number of features.

We also evaluated the feature vector $\vec{\Theta}(R)_{Q_1,Q_2,...,Q_m}$ for different values of R. Fig. 4 shows the accuracies yielded considering the combinations $Q = \{04, 09, 14\}$ and $Q = \{04, 14, 19\}$. The maximum value of radius R is associated with the zone of connection between the pixels (i.e., vertices). Thus, lower values of R represent the closest pixels and, as R increases, the reach of connection increases as well. The results show that the lowest values of R provide better accuracies when compared to the highest values. This



(a) $Q = \{04, 09, 14\}$ (b) $Q = \{04, 14, 19\}$

Fig. 4. Accuracies using the feature vector $\vec{\Theta}(R)_{Q_1,Q_2,...,Q_m}$ for the two better set of Q with different values of maximum radius R.

Table 1
Accuracies of the feature vector $\vec{\Theta}(R)_{Q_1,Q_2,...,Q_m}$ using different values of Q and their combinations for the maximum radius $R = 4$.

$\{Q_1, Q_2, \ldots, Q_m\}$	No of features	Outex	USPTex	Brodatz	Vistex
{04, 09}	45	88.60	94.07	93.02	97.22
{04, 14}	60	88.82	94.98	93.86	98.50
{04, 19}	75	88.97	95.46	94.88	97.92
{04, 29}	105	88.38	94.76	94.48	97.80
{04, 39}	135	87.57	95.42	95.27	98.15
{09, 14}	75	88.09	94.72	93.52	97.92
{09, 19}	90	88.60	94.42	94.26	98.15
{09, 29}	120	87.50	94.24	94.14	97.80
{09, 39}	150	86.76	94.81	94.88	97.80
{14, 19}	105	89.04	94.94	94.54	98.26
{14, 29}	135	88.09	94.90	94.48	97.92
{14, 39}	165	87.65	95.29	95.33	98.15
{19, 29}	150	87.50	94.59	94.76	97.45
{19, 39}	180	87.28	95.03	95.27	98.03
{29, 39}	210	85.88	94.37	95.05	97.92
{04, 09, 14}	**90**	**89.34**	**95.46**	**95.05**	**98.61**
{04, 09, 19}	105	89.71	95.55	95.16	98.26
{04, 09, 29}	135	88.68	95.51	94.88	97.80
{04, 09, 39}	165	87.94	95.90	95.72	97.80
{04, 14, 19}	**120**	**89.41**	**95.94**	**95.21**	**98.38**
{04, 14, 29}	150	88.68	95.59	95.05	98.26
{04, 14, 39}	180	88.53	95.90	95.89	98.84
{04, 19, 29}	165	89.12	95.94	95.27	98.03
{04, 19, 39}	195	88.68	95.90	95.61	98.38
{04, 29, 39}	225	87.94	95.03	95.72	98.38
{09, 14, 19}	135	89.56	95.24	94.99	98.50
{09, 14, 29}	165	88.75	95.42	95.10	98.15
{09, 14, 39}	195	88.01	95.68	95.61	98.73
{09, 19, 29}	180	88.75	95.24	94.93	98.26
{09, 19, 39}	210	87.94	95.38	95.05	97.92
{09, 29, 39}	240	88.01	94.59	94.99	97.92
{14, 19, 29}	195	88.75	95.33	95.05	98.50
{14, 19, 39}	225	88.38	95.72	95.44	98.61
{14, 29, 39}	255	88.09	95.03	95.50	98.50
{19, 29, 39}	270	88.01	94.63	95.50	97.92

suggests that local patterns are more important than global patterns to discriminate the textures used in this work.

Furthermore, we analyzed the combination of vectors $\vec{\Theta}(R)_{Q_1,Q_2,...,Q_m}$ (showed in Table 1) with different values of maximum radius R, resulting in the vector $\vec{\Psi}(R_1, R_2)_{Q_1,Q_2,...,Q_m}$. To compute this vector, we used the combinations of Q that provided the best results in Table 1: $Q = \{4, 9, 14\}$ and $Q = \{4, 14, 19\}$. In this experiment, we computed the vector $\vec{\Psi}(R_1, R_2)_{Q_1,Q_2,...,Q_m}$ for two values of R (i.e, up to two combinations of $\vec{\Theta}(R)_{Q_1,Q_2,...,Q_m}$) due to the large number of features generated.

Table 2 shows the results of the vectors $\vec{\Psi}(R_1, R_2)_{Q_1,Q_2,...,Q_m}$ using the combination $Q = \{04, 09, 14\}$. The highest accuracy was provided by the vector $\vec{\Psi}(04, 06)_{04,09,14}$. The results of the vectors

Table 2
Accuracies using different sets of radii R and $Q = \{04, 09, 14\}$.

$\{R_1, R_2\}$	No of features	Outex	USPTex	Brodatz	Vistex
{04, 06}	180	91.54	96.64	96.11	98.73
{04, 08}	180	91.47	96.25	95.88	98.26
{04, 10}	180	91.47	96.55	95.83	98.84
{04, 12}	180	91.69	96.25	95.72	98.26
{06, 08}	180	91.54	96.47	95.77	98.61
{06, 10}	180	90.74	96.25	95.83	98.50
{06, 12}	180	90.44	96.38	95.83	98.38
{08, 10}	180	90.58	96.03	95.15	98.49
{08, 12}	180	90.29	95.72	95.21	98.14
{10, 12}	180	90.59	95.46	94.93	98.38

Table 3
Accuracies using different sets of radii R and $Q = \{4, 14, 19\}$.

$\{R_1, R_2\}$	No of features	Outex	USPTex	Brodatz	Vistex
{04, 06}	240	90.07	96.73	95.83	99.19
{04, 08}	240	91.17	96.68	96.05	98.95
{04, 10}	**240**	**91.32**	**96.95**	**96.06**	**99.19**
{04, 12}	240	91.76	96.64	96.11	98.61
{06, 08}	240	90.14	96.29	96.39	98.61
{06, 10}	240	89.63	96.60	95.95	98.50
{06, 12}	240	90.29	96.55	96.06	98.26
{08, 10}	240	90.00	95.94	95.72	98.37
{08, 12}	240	91.32	95.94	95.77	98.03
{10, 12}	240	90.51	96.03	95.05	98.03

$\vec{\Psi}(R_1, R_2)_{Q_1,Q_2,...,Q_m}$ built with the combination $Q = \{04, 14, 19\}$ are shown in Table 3. In this experiment, the best results were obtained by using the vector $\vec{\Psi}(04, 10)_{04,14,19}$. Tables 2 and 3 also show that by combining the vector $\vec{\Theta}(R)_{Q_1,Q_2,...,Q_m}$ with different values of R, the accuracy increases approximately 1% on the databases. However, in the two cases, the combinations of high values of R provide inferior results. Also, even though the best results of the two Tables ($\vec{\Psi}(04, 06)_{04,09,14}$ and $\vec{\Psi}(04, 10)_{04,14,19}$) are similar, notice that the vector $\vec{\Psi}(04, 10)_{04,14,19}$ has a number of features larger than the vector $\vec{\Psi}(04, 06)_{04,09,14}$.

5.2. Comparison with other methods

To evaluate the results obtained by the proposed method, we performed comparisons with methods present in the literature. The experimental setup used was the same for all the methods (LDA with leave-one-out), except for CLBP, which used the classifier 1-Nearest Neighborhood (1-NN) with distance chi-square, according to the original paper. For our method, we adopted the two texture signatures that obtained the best results in the previous analysis: $\vec{\Psi}(04, 06)_{04,09,14}$ and $\vec{\Psi}(04, 10)_{04,14,19}$.

Table 4
Comparison of accuracies of different texture analysis methods in four texture databases. A subset of the compared results is present in [49].

Methods	No of features	Outex	USPTex	Brodatz	Vistex
GLCM [23]	24	80.73	83.64	90.43	92.24
GLDM [53]	60	86.76	92.06	94.43	97.11
Gabor Filters [34]	48	81.91	89.22	89.86	93.29
Fourier [53]	63	81.91	67.50	75.90	79.51
Fractal [1]	69	80.51	78.27	87.16	91.67
Fractal Fourier [18]	68	68.38	59.47	71.96	79.75
LBP [38]	256	81.10	85.43	93.64	97.92
LBPV [22]	555	75.66	54.97	86.26	88.65
CLBP [21]	648	85.80	91.14	95.32	98.03
AHP [56]	120	88.31	94.85	94.88	98.38
BSIF [27]	256	77.43	77.66	91.44	88.66
LCP [20]	81	86.25	91.14	93.47	94.44
LFD [33]	276	82.57	83.55	90.99	94.68
LPQ [39]	256	79.41	85.12	92.51	92.48
ELM Signature [49]	180	89.71	95.11	95.27	98.15
CNTD [3]	108	86.76	91.71	95.27	98.03
$\vec{\Theta}(04)_{04,09,14}$	90	89.34	95.46	95.05	98.61
$\vec{\Psi}(04,06)_{04,09,14}$	180	**91.54**	96.64	**96.11**	98.73
$\vec{\Psi}(04,10)_{04,14,19}$	240	91.32	**96.95**	96.06	**99.19**

Table 5
Comparison of accuracies of different texture analysis methods in rotated texture databases. A subset of the compared results is present in [49].

Methods	No of features	Outex	Brodatz	Vistex
GLCM [23]	24	75.53	91.48	94.28
GLDM [53]	60	84.76	95.17	99.16
Gabor Filters [34]	48	78.07	90.17	96.97
Fourier [53]	63	88.90	94.59	97.64
Fractal [1]	69	86.76	95.99	97.47
Fractal Fourier [18]	68	81.02	89.68	97.31
LBP [38]	256	71.39	91.97	94.44
LBPV [22]	555	89.83	98.11	98.48
CLBP [21]	648	97.59	99.18	99.49
AHP [56]	120	92.25	99.43	99.83
BSIF [27]	256	65.78	82.88	85.19
LCP [20]	81	93.58	98.53	99.83
LFD [33]	276	92.51	98.53	99.33
LPQ [39]	256	90.64	99.67	98.99
ELM Signature [49]	180	97.99	99.43	99.83
CNTD [3]	108	85.56	94.66	98.48
$\vec{\Theta}(04)_{04,09,14}$	90	95.86	99.10	99.83
$\vec{\Psi}(04,06)_{04,09,14}$	180	96.52	99.43	99.83
$\vec{\Psi}(04,10)_{04,14,19}$	240	97.33	99.43	99.83

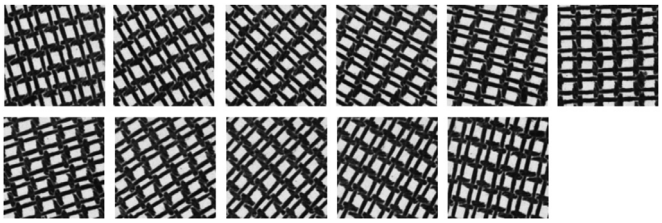


Fig. 5. Examples of a texture from the Brodatz database rotated in 11 different angles. From left to right, top to bottom: 15°, 30°, 45°, 60°, 75°, 90°, 105°, 120°, 135°, 150°, 165°.

Table 4 presents the results obtained by all the methods in the four image databases evaluated. The results show that our proposed method obtained the best results when compared to the other methods using both signatures. Also, it is important to stress that our method reached higher accuracies than the ELM signature and CNDT method, which are based on neural networks and complex networks, respectively. This suggests that our method obtained superior performance because it has simultaneously the main characteristics of both compared methods. In other words, the ELM signature uses only pixel intensities to train the neural network, without any valuable information from complex network modeling, and the CNTD method models images as complex networks and computes only traditional measures, without using a neural network to extract the deep characteristics from these complex networks.

5.3. Rotation robustness

In this section, we evaluated the ability of the method to deal with rotated textures. Rotation robustness is an important characteristic desirable in problems and applications that use texture analysis methods. In this experiment, we used three rotated texture databases: Brodatz, Outex, and Vistex. In these databases, each original texture image is rotated in a specific angle, totaling 11 different angles, according to the procedure in [3]. Thus, each texture class has as samples the 11 different angles. For example, the Brodatz database has 111 original texture images (each image is a class), totaling $11 \times 111 = 1221$ samples. The Vistex and Outex have 594 (54 texture classes) and 748 (68 texture classes) samples, respectively. Fig. 5 shows examples of a texture image rotated in different angles. For validation, as previously explained in Section 4, a leave-one-out scheme was used in this experiment. In this scheme, a texture rotated in a determined angle is used for testing while the remaining samples representing the other 10 angles as well as the samples of the other classes are used for training the LDA classifier. In other words, as we are using a single image in different rotations for each class, this experiment then measures the rate of rotation tolerance of the methods.

Table 6
Comparison of processing time of the methods.

Methods	Time (s)
GLCM [23]	0.0167
GLDM [53]	0.0051
Gabor Filters [34]	0.0392
Fourier [53]	0.0148
Fractal [1]	0.5092
Fractal Fourier [18]	0.0038
LBP [38]	0.0032
LBPV [22]	0.2005
CLBP [21]	0.4740
AHP [56]	0.0649
BSIF [27]	0.0051
LCP [20]	0.0398
LFD [33]	0.1067
LPQ [39]	0.1376
ELM Signature [49]	1.0067
CNTD [3]	0.0851
$\vec{\Theta}(04)_{04,09,14}$	0.1132
$\vec{\Psi}(04,06)_{04,09,14}$	0.2538
$\vec{\Psi}(04,10)_{04,14,19}$	0.4619

The accuracies obtained by each method in the three databases are shown in Table 5. These results demonstrate that our proposed method is very robust to rotation, once it presents the highest accuracy on the Vistex database (drawing with AHP, LCP and ELM signature), the second-best accuracy on the Brodatz database (drawing with ELM signature and AHP), and the third-best success rate on the Outex dataset. It is also important to stress the very slight difference in the performance of our method when compared to the best accuracies obtained from these two databases, which is 0.66% for Outex and 0.24% for the Brodatz dataset.

On the other hand, the method with the lowest rotation robustness was BSIF. Moreover, note that some methods that had a good performance in the non-rotated databases (Table 4) do not work very well with rotated images. For example, in Table 4, the

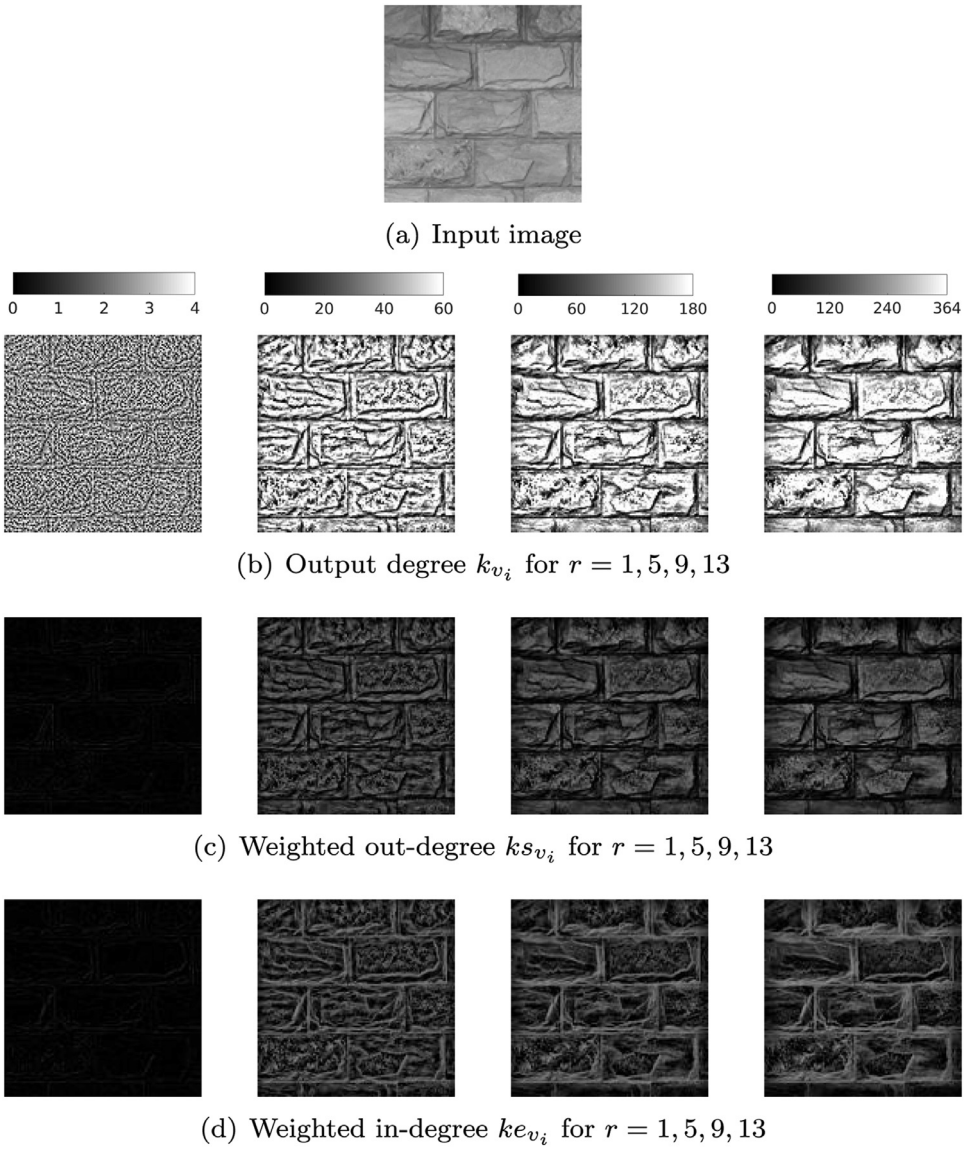


Fig. 6. A texture image (a) from the USPTex database is modeled as directed complex network for different values of r and visual representations of the network measures of each vertex are shown (d-f). The pixel values are obtained by normalizing each measure so that the max vertex degree for each radius (4,60,180 and 364) correspond to intensity 255 (white).

LBP descriptor obtained the highest accuracies when compared to LPQ and LBPV methods, but its results are lower than the accuracies of these two methods in the rotation experiment. This confirms even more how discriminative our proposed method is, once it obtained high performance in both the rotated and non-rotated databases.

5.4. Processing time analysis

In order to improve the evaluation of the methods, the running time of each approach per image processed (200 × 200 pixels) is shown in Table 6. In the experiments, we used a 3.60 GHz Intel(R) Core i7-7820X, 64GB RAM and 64-bit Operating System. The LBP method has the lowest computational time, while the ELM Signature approach obtained the highest processing time. On the other hand, the results of the proposed method indicate a competitive running time for real-time applications compared to other methods. The proposed approach took, on average, 0.1132 s, 0.2538 s and 0.4619 s using the feature vectors $\vec{\Theta}(04)_{04,09,14}$, $\vec{\Psi}(04,06)_{04,09,14}$ and $\vec{\Psi}(04,10)_{04,14,19}$, respectively. Thus, we believe that our method has a good tradeoff between the running time and the classification performance.

5.5. Discussion

To understand the features obtained by the proposed method and their relation with the image, we first analyze the topological properties of the modeled directed complex network. Fig. 6 shows visual representations of the three vertex measures obtained by converting its value into a gray-level pixel. In general, it is possible to notice that each one highlights different intensity variation patterns of the image in a multiscale fashion by varying r. For example, for low values of r (column 1), finer details of the texture are evidenced by the network, and as we increase r (column 4), it tends to highlight more global characteristics. This demonstrates that network modeling captures the main physical properties of the texture.

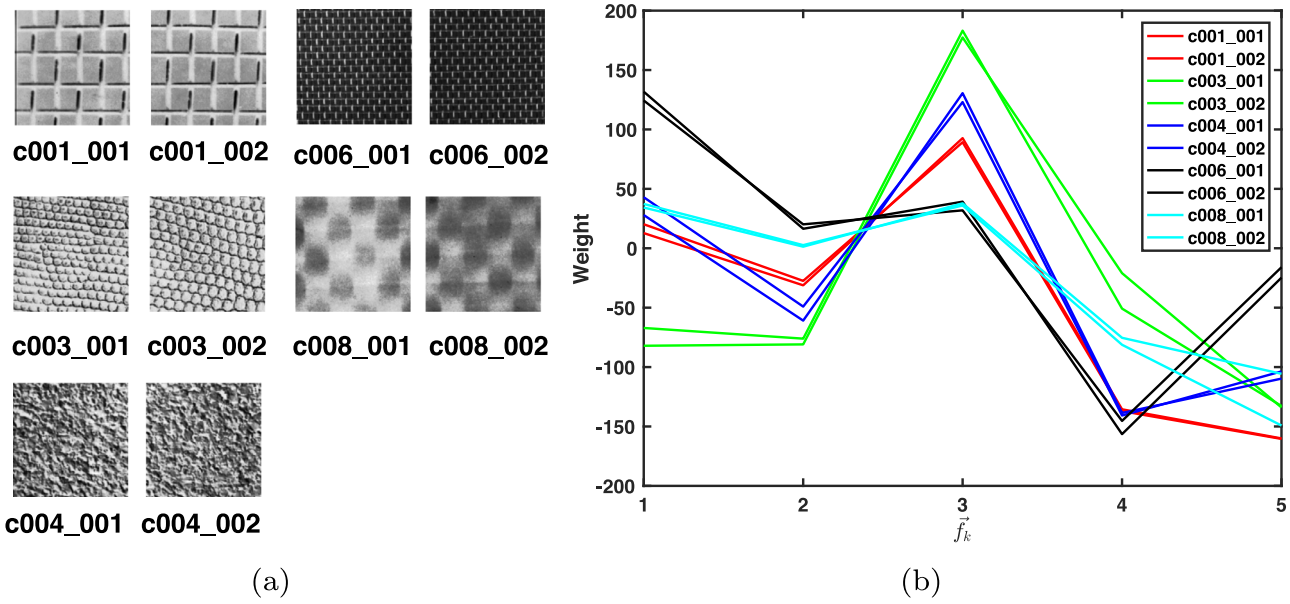


Fig. 7. Texture samples (a) from the Brodatz database and their respective feature vectors $\vec{f}_k$ in (b). This feature vector was calculated using the parameters $R = 04$ and $Q = 04$.

Table 7
Results for Sequential Forward Selection (SFS) and Sequential Backward Selection (SBS) applied on the three proposed feature vectors. # represents the number of features selected.

FS Approach	Feature Vector	Outex	USPTex	Brodatz	Vistex
SFS	$\vec{\Theta}(04)_{04,09,14}$	85.00 (#11)	93.46 (#35)	92.06 (#23)	92.82 (#11)
	$\vec{\Psi}(04,06)_{04,09,14}$	86.91 (#20)	92.36 (#22)	94.20 (#33)	95.72 (#21)
	$\vec{\Psi}(04,10)_{04,14,19}$	88.16 (#19)	93.54 (#30)	94.03 (#39)	96.18 (#18)
SBS	$\vec{\Theta}(04)_{04,09,14}$	89.19 (#78)	95.64 (#82)	94.59 (#76)	98.50 (#74)
	$\vec{\Psi}(04,06)_{04,09,14}$	91.91 (#168)	96.86 (#159)	96.06 (#170)	99.07 (#80)
	$\vec{\Psi}(04,10)_{04,14,19}$	91.40 (#230)	96.95 (#223)	95.78 (#228)	98.96 (#105)

In the second step of the method, a randomized neural network is used to compute the feature vector from the modeled networks. The feature vector $\vec{f}_k$ calculated for different samples of textures is shown in Fig. 7. This feature vector is composed of the trained weights of the output layer and bias of the RNN. As can be seen, each class of texture has a different signature. This corroborates that the neural networks can effectively learn to synthesize the complex network topological measures into texture information. Therefore, their learned weights are capable of representing and separating different texture classes.

In terms of feature vector size of the method, even though our proposed method has signatures with a larger number of descriptors when compared to some methods of the literature, it is important to emphasize that, if we consider only the vector $\vec{\Theta}(R)_{Q_1,Q_2,...,Q_m}$, the results are still competitive (see Table 4). For instance, the vector $\vec{\Theta}(04)_{4,9,14}$, which has only 90 features, provides superior performance on the Vistex and USPTex databases. In the remainder databases, the results are very close to the highest accuracies (only 0.37% smaller than the result of ELM signature on the Outex database and 0.27% smaller than the accuracy of the CLBP on the Brodatz database).

To evaluate the discriminative power of the features and reduce the feature vector size, we also test strategies for feature selection (FS). The idea is to reduce a feature vector to a subset of features that is most relevant to classification. For this, we used the strategy of sequential feature selection [28]. This approach has two variants: the sequential forward selection (SFS), in which, from an empty set, the features are sequentially added, and the sequential backward selection (SBS), in which, starting from a full set, the features are sequentially removed until reaching a certain criterion [28]. After applying the feature selection approach over the feature vectors, we classified the selected feature set using a leave-one-out cross-validation scheme and the LDA classifier. Table 7 summarizes the results using the SFS and SBS strategies to reduce the size of the proposed features vectors. For the SFS, the size of the feature vectors is reduced considerably and the accuracy decreased between 1.91% and 5.79%. On the other hand, in the SBS approach, the size of the selected feature set is larger and the accuracies are higher, sometimes surpassing the results of the original feature vectors. These results demonstrate that the features extracted by the proposed method are discriminative and, even with a reduced feature vector, it overcomes many methods of the literature.

6. Conclusion

This paper presents an innovative approach of texture feature extraction based on the fusion of complex network and randomized neural network. In the proposed method, a new approach to model the image as a CN that uses only one parameter is presented. We also proposed a new way of characterizing the CN based on the idea of using the output weights of a randomized neural network trained with topological properties of the CN. The obtained classification results on four databases outperformed other methods of the literature. Also, the proposed approach has an excellent trade-off between performance and size of the feature vectors. This demonstrates that the proposed approach is highly discriminative using the three feature vectors considered. In this way, this paper shows that the fusion of complex network and ran-

domized neural network is a research field with great potential as a feasible texture analysis methodology.

Declaration of Competing Interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Acknowledgments

Lucas Correia Ribas gratefully acknowledges the financial support grant #s 2016/23763-8 and 16/18809-9, São Paulo Research Foundation (FAPESP). Jarbas Joaci de Mesquita Sá Junior thanks the *National Council for Scientific and Technological Development* - CNPq, Brazil (grant: 302183/2017-5) for the financial support of this work. Leonardo Felipe dos Santos Scabini acknowledges support from CNPq (grant 134558/2016-2) and FAPESP (grant 2019/07811-0). Odemir M. Bruno thanks the financial support of CNPq (grant # 307897/2018-4) and FAPESP (grant #s 14/08026-1 and 16/18809-9).

References

[1] A.R. Backes, D. Casanova, O.M. Bruno, Plant leaf identification based on volumetric fractal dimension, Int. J. Pattern Recognit. Artif. Intell. 23 (6) (2009) 1145–1160.
[2] A.R. Backes, D. Casanova, O.M. Bruno, Color texture analysis based on fractal descriptors, Pattern Recognit. 45 (5) (2012) 1984–1992.
[3] A.R. Backes, D. Casanova, O.M. Bruno, Texture analysis and classification: a complex network-based approach, Inf. Sci. 219 (2013) 168–180.
[4] A.L. Barabási, Network Science, Cambridge university press, 2016.
[5] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, D.-U. Hwang, Complex networks: structure and dynamics, Phys. Rep. 424 (4) (2006) 175–308.
[6] P. Brodatz, Textures: A Photographic Album for Artists and Designers, Dover Publications, New York, 1966.
[7] D. Calvetti, S. Morigi, L. Reichel, F. Sgallari, Tikhonov regularization and the l-curve for large discrete ill-posed problems, J. Comput. Appl. Math. 123 (1) (2000) 423–446.
[8] S.V.A.B. Cantero, D.N. Gonalves, L.F. dos Santos Scabini, W.N. Gonalves, Importance of vertices in complex networks applied to texture analysis, IEEE Trans. Cybern. (2018) 1–10.
[9] M. Cimpoi, S. Maji, I. Kokkinos, A. Vedaldi, Deep filter banks for texture recognition, description, and segmentation, Int. J. Comput. Vis. 118 (1) (2016) 65–94.
[10] T.M. Cover, Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition, IEEE Trans. Electron. Comput. EC-14 (3) (1965) 326–334.
[11] G. Csurka, C. Dance, L. Fan, J. Willamowski, C. Bray, Visual categorization with bags of keypoints, in: ECCV International Workshop on Statistical Learning in Computer Vision, 2004, pp. 1–22.
[12] L. da Fontoura Costa, O.N.O. Jr., G. Travieso, F.A. Rodrigues, P.R.V. Boas, L. Antiqueira, M.P. Viana, L.E.C. Rocha, Analyzing and modeling real-world phenomena with complex networks: a survey of applications, Adv. Phys. 60 (3) (2011) 329–412.
[13] N.R. da Silva, P. Van der Weeën, B. De Baets, O.M. Bruno, Improved texture image classification through the use of a corrosion-inspired cellular automaton, Neurocomputing 149 (2015) 1560–1572.
[14] J. Daugman, C. Downing, Gabor wavelets for statistical pattern recognition, in: The handbook of brain theory and neural networks, MIT Press, 1995, pp. 414–419.
[15] E. de Ves, D. Acevedo, A. Ruedin, X. Benavent, A statistical model for magnitudes and angles of wavelet frame coefficients and its application to texture retrieval, Pattern Recognit. 47 (9) (2014) 2925–2939.
[16] I. El khadiri, A. Chahi, Y. El merabet, Y. Ruichek, R. Touahni, Local directional ternary pattern: a new texture descriptor for texture classification, Comput. Vision Image Understanding 169 (2018) 14–27.
[17] R.A. Fisher, The use of multiple measurements in taxonomic problems, Ann. Eugen. 7 (7) (1936) 179–188.
[18] J.B. Florindo, O.M. Bruno, Fractal descriptors based on Fourier spectrum applied to texture analysis, Physica A 391 (20) (2012) 4909–4922.
[19] K. Fukunaga, Introduction to Statistical Pattern Recognition, 2nd, Academic Press, 1990.
[20] Y. Guo, G. Zhao, M. Pietikäinen, Texture classification using a linear configuration model based descriptor., in: BMVC, Citeseer, 2011, pp. 1–10.
[21] Z. Guo, L. Zhang, D. Zhang, A completed modeling of local binary pattern operator for texture classification, IEEE Trans. Image Process. 19 (6) (2010) 1657–1663.
[22] Z. Guo, L. Zhang, D. Zhang, Rotation invariant texture classification using lbp variance (lbpv) with global matching, Pattern Recognit. 43 (3) (2010) 706–719.
[23] R.M. Haralick, Statistical and structural approaches to texture, Proc. IEEE 67 (5) (1979) 786–804.
[24] R.M. Haralick, K. Shanmugam, I.H. Dinstein, Textural features for image classification, IEEE Trans. Syst. Man Cybern. (6) (1973) 610–621.
[25] G.-B. Huang, Q.-Y. Zhu, C.-K. Siew, Extreme learning machine: theory and applications, Neurocomputing 70 (1) (2006) 489–501.
[26] B. Kaddar, H. Fizazi, A.-O. Boudraa, Texture features based on an efficient local binary pattern descriptor, Comput. Electr. Eng. 70 (2018) 496–508.
[27] J. Kannala, E. Rahtu, Bsif: Binarized statistical image features, in: Pattern Recognition (ICPR), 2012 21st International Conference on, IEEE, 2012, pp. 1363–1366.
[28] R. Kohavi, G.H. John, Wrappers for feature subset selection, Artif. Intell. 97 (1–2) (1997) 273–324.
[29] W.-K. Lam, C.-K. Li, Rotated texture classification by improved iterative morphological decomposition, IEE Proceedings-Vision Image Signal Process. 144 (3) (1997) 171–179.
[30] D.H. Lehmer, Mathematical methods in large scale computing units, Annals Comp. Laboratory Harvard University 26 (1951) 141–146.
[31] L. Liu, J. Chen, P. Fieguth, G. Zhao, R. Chellappa, M. Pietikainen, A survey of recent advances in texture representation, arXiv:1801.10324 (2018).
[32] L. Liu, J. Chen, P. Fieguth, G. Zhao, R. Chellappa, M. Pietikäinen, From bow to cnn: two decades of texture representation for texture classification, Int. J. Comput. Vis. 127 (1) (2019) 74–109.
[33] R. Maani, S. Kalra, Y.-H. Yang, Noise robust rotation invariant features for texture classification, Pattern Recognit. 46 (8) (2013) 2103–2116.
[34] B.S. Manjunath, W.-Y. Ma, Texture features for browsing and retrieval of image data, IEEE Trans. Pattern Anal. Mach. Intell. 18 (8) (1996) 837–842.
[35] G.H.B. Miranda, J. Machicao, O.M. Bruno, Exploring spatio-temporal dynamics of cellular automata for pattern recognition in networks, Sci Rep 6 (2016) 37329.
[36] E.H. Moore, On the reciprocal of the general algebraic matrix, Bull. Am. Math. Soc. 26 (1920) 394–395.
[37] T. Ojala, T. Mäenpää, M. Pietikäinen, J. Viertola, J. Kyllönen, S. Huovinen, Outex: new framework for empirical evaluation of texture analysis algorithms, 2002a.
[38] T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Trans. Pattern Anal. Mach. Intell. 24 (7) (2002) 971–987.
[39] V. Ojansivu, J. Heikkilä, Blur insensitive texture classification using local phase quantization, in: International conference on image and signal processing, Springer, 2008, pp. 236–243.
[40] D.K. Panjwani, G. Healey, Markov random field models for unsupervised segmentation of textured color images, IEEE Trans. Pattern Anal. Mach. Intell. 17 (10) (1995) 939–954.
[41] Y.-H. Pao, G.-H. Park, D.J. Sobajic, Learning and generalization characteristics of the random vector functional-link net, Neurocomputing 6 (2) (1994) 163–180.
[42] Y.-H. Pao, Y. Takefuji, Functional-link net computing: theory, system architecture, and functionalities, Computer 25 (5) (1992) 76–79.
[43] S.K. Park, K.W. Miller, Random number generators: good ones are hard to find, Commun. ACM 31 (10) (1988) 1192–1201.
[44] R. Penrose, A generalized inverse for matrices, Math. Proc. Cambridge Philos. Soc. 51 (3) (1955) 406–413.
[45] R. Picard, C. Graczyk, S. Mann, J. Wachman, L. Picard, L. Campbell, Vision Texture Database, Media Laboratory, MIT, Cambridge, Massachusetts, 1995.
[46] L.C. Ribas, D.N. Gonçalves, J.P.M. Oruê, W.N. Gonçalves, Fractal dimension of maximum response filters applied to texture analysis, Pattern Recognit. Lett. 65 (2015) 116–123.
[47] L.C. Ribas, A. Manzanera, O.M. Bruno, A fractal-based approach to network characterization applied to texture analysis, in: International Conference on Computer Analysis of Images and Patterns, Springer, 2019, pp. 129–140.
[48] C.D. Ruberto, L. Putzu, G. Rodriguez, Fast and accurate computation of orthogonal moments for texture analysis, Pattern Recognit. 83 (2018) 498–510.
[49] J.J.M. Sá Junior, A.R. Backes, ELM Based signature for texture classification, Pattern Recognit. 51 (2016) 395–401.
[50] L.F. Scabini, R.H. Condori, W.N. Gonalves, O.M. Bruno, Multilayer complex network descriptors for color-texture characterization, Inf. Sci. 491 (2019) 30–47.
[51] W.F. Schmidt, M.A. Kraaijveld, R.P.W. Duin, Feedforward neural networks with random weights, in: Proceedings., 11th IAPR International Conference on Pattern Recognition. Vol.II. Conference B: Pattern Recognition Methodology and Systems, 1992, pp. 1–4.
[52] A.N. Tikhonov, On the solution of ill-posed problems and the method of regularization, Dokl. Akad. Nauk USSR 151 (3) (1963) 501–504.
[53] J.S. Weszka, C.R. Dyer, A. Rosenfeld, A comparative study of texture measures for terrain classification, IEEE Trans. Syst. Man Cybern. (4) (1976) 269–285.
[54] J. Zhang, T. Tan, Brief review of invariant texture analysis methods, Pattern Recognit. 35 (3) (2002) 735–747.
[55] Z. Zhang, S. Liu, X. Mei, B. Xiao, L. Zheng, Learning completed discriminative local features for texture classification, Pattern Recognit. 67 (2017) 263–275.
[56] Z. Zhu, X. You, C.P. Chen, D. Tao, W. Ou, X. Jiang, J. Zou, An adaptive hybrid pattern for noise-robust texture analysis, Pattern Recognit. 48 (8) (2015) 2592–2608.

Lucas C. Ribas received a B.Sc. (2014) in Information System at the Federal University of Mato Grosso do Sul, a M.Sc. (2017) in Computer Science at the Institute of Mathematics and Computer Science at the University of S. Paulo (ICMC/USP) and is now a Ph.D. candidate at the ICMC/USP. His fields of interest include Computer Vision, Neural Networks, Complex Networks and Pattern Recognition.

Jarbas Joaci de Mesquita Sá Junior is a Professor at the Federal University of Ceará, Campus Sobral, in Brazil. He received his B.Sc. (2005) in Computer Science at the State University Vale do Acaraú, M.Sc. (2008) in Computer Science at the University of S. Paulo and Ph.D. (2013) in Teleinformatics Engineering at the Federal University of Ceará. His fields of interest include Computer Vision, Image Analysis and Pattern Recognition.

Leonardo F. S. Scabini received his B.S. in Computer Science from the Federal University of Mato Grosso do Sul (2015) and his MSc de- gree in Computational Physics from the University of São Paulo (2018). Currently a Ph.D. student in Computational Physics at the University of São Paulo. His research interests include computer vision and artificial intelligence, approaching topics such as feature extraction, image classification, complex networks and neural networks.

Odemir M. Bruno is a Professor at the S. Carlos Institute of Physics at the University of S. Paulo in Brazil. He received his B.Sc. in Computer Science, from the Piracicaba Engineering College (Brazil), his M.Sc. in Applied Physics and his Ph.D. in Computational Physics at the University of S. Paulo (Brazil). His fields of interest include Data Science, Artificial Intelligence, Complex Systems, Chaos Theory, Pattern Recognition and Bioinformatics. He is an author of about one hundred and fifty papers in journals and several book chapters. He is a co-author of two books and an inventor of seven patents.